INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **39**:213–246 (DOI: 10.1002/fld.284)

Simulating three-dimensional aeronautical flows on mixed block-structured/semi-structured/unstructured meshes

J. A. Shaw*,† and A. J. Peace

Aircraft Research Association Ltd, Manton Lane, Bedford, MK41 7PF, U.K.

SUMMARY

The design requirements of a computational fluid dynamics (CFD) method for modelling high Reynolds number flows over complete aircraft are reviewed. It is found that the specifications are unlikely to be met by an approach based on the sole use of either structured or unstructured grids. Instead, it is proposed that a hybrid combination of these grids is appropriate. Techniques for developing such meshes are given and the process of establishing the data structure defining the meshes described. Details of a flow algorithm which operates on a hybrid mesh are presented. A description is given of the suitability and generation of hybrid grids for a number of examples, and results from flow simulations shown. Finally, issues still to be addressed in the practical use of these meshes are discussed. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: computational grids; three-dimensional flow; finite volume method; Euler equations of motion; vortical flow; transonic flow

INTRODUCTION

Computational fluid dynamics (CFD) is accepted as a prime area for technology acquisition and application throughout the aerospace industry. This recognition is earned as a result of evidence that substantial reductions in design costs can be attained through enhanced synergy between CFD and wind tunnel tests [1]. However, it comes at a time when cuts are continually being made to research and development budgets. Consistent with this, aeronautical engineers are increasingly expected to have multi-disciplined skills and are placing greater demands on the performance of their available tools. The design of the next generation of CFD technology, which is aimed at simulating viscous flows over complete aircraft, must address these needs. Much is still required from CFD, but after all, much has been promised.

Central to the current debate on providing a capability to simulate high Reynolds number flows is the creation of a suitable mesh. General agreement has been reached on what is needed in a mesh, but crucially, not on how to get it.

*Correspondence to: J. A. Shaw, Aircraft Research Association Ltd, Manton Lane, Bedford, MK41 7PF, U.K.
†E-mail: jshaw@ara.co.uk

Received 1 October 2001
Accepted 19 October 2001

A mesh should conform to boundaries, contain points which are distributed effectively and be defined in a manner amenable to efficient computations. Furthermore, the connectivity of points should form elements that satisfy certain geometric criteria and do not overlap.

Methods for forming such a mesh are generally classified as either structured or unstructured. Promoters of structured schemes highlight the efficiency and accuracy that is attained through the employment of regularly arranged hexahedral volumes [2]. Supporters of unstructured schemes emphasize the geometric flexibility and suitability for adaptation inherent to the use of irregularly connected tetrahedral volumes [3].

For some years now, both approaches have appeared to have limitations when considering them as a basis for viscous flow simulations over entire aircraft. On the one hand, as more structure is added to a grid, so the constraints imposed on the grid increase. This ultimately impacts on both the ease of use of, and range of problems that can be addressed with, structured grids. On the other hand, as more structure is removed from a grid, so the amount of information that is readily accessible decreases. This ultimately impacts on both the generation of highly compressed elements, and the accuracy of discretization schemes, for unstructured grids.

In an attempt to overcome these difficulties, there is clear evidence of an increasing cross-fertilization of ideas and techniques between the two camps. The limit of this trend is to replace the sole use of one mesh type by the use of combined meshes composed of both structured and unstructured grids—hybrid grids. This combination of grid types not only allows the benefits of structured and unstructured grids to be attained simultaneously, but also allows high quality, efficient grids to be achieved throughout the domain, due to the appropriate use of each element type.

This paper begins by motivating further the use of hybrid grids. The techniques employed to generate each region of grid are discussed in turn, with attention given to the issue of abutting the different mesh types together. The development of the data structure which describes the meshes is covered and a flow algorithm that operates on hybrid grids is detailed. Results from a number of applications are presented, emphasizing the usefulness of the approach. The paper concludes with a discussion on factors that are seen to be central to the future acceptance and practical use of hybrid grids.

MOTIVATION FOR USING HYBRID GRIDS

Design needs for a CFD capability

A number of factors must be considered in the extension of CFD to provide useful aerodynamic data from viscous flow simulations over aircraft configurations. These can be summarized as usability, efficiency, robustness and accuracy.

(i) Usability. A CFD method is not seen to be acceptable in an industrial environment if it absorbs significant manpower, or is difficult either to learn or to pick up again after a period of time. This statement is particularly relevant to the mesh generation phase if a ‘one-off’ flow calculation is all that is required. When the dominant cost in the use of CFD is in obtaining the grid (i.e. not the end product), the use of the tool is more difficult to justify. There is a strong onus on the developer to improve usability by

Table I. Ability of approaches to meet requirements.

	Unstructured	Structured
Usability	Yes	No
Efficiency	No	Yes
Robustness	Yes	No
Accuracy	No	Yes

providing error messages and other useful output if problems arise during execution. However, in practice, this functionality often only reaches maturity after the software has been used widely.

(ii) Efficiency. The use of CFD is increasing. One of the main reasons for this is the emergence of capabilities based on the repetitive execution of flow solvers; design optimization [4], performance analysis prediction and store release simulation [5], are just three examples of this. For these applications of CFD, the CPU and memory requirements of the flow solver will always dominate the issues of cost and turn-around.

(iii) Robustness. A CFD system must be capable of being applied to a wide range of problems, involving highly three-dimensional surfaces and/or complex flow fields containing vortices, shock-waves, etc. This capability must be achieved with software which both has the required applicability and functions reliably.

(iv) Accuracy. Ultimately, a CFD tool must deliver an answer with a possible error bound that is both definable and acceptable to the aeronautical engineer. This requires that the method is well validated and evaluated for the problem of interest. It must also behave in a consistent manner to provide engineers with confidence that changes to output data are solely a result of changes to input data. In addition, the modelling errors must not be so large as to introduce misleading solutions.

For any one particular application, the emphasis placed on each of these factors will be different. However, what is apparent is that an aerodynamics department would prefer to have a single CFD environment, possibly composed of multiple codes, that can be applied to a wide range of problems. The potential benefits from this in terms of validation, evaluation, training, resourcing and support are immediately apparent. A total CFD capability must therefore be able to meet all requirements.

Matching gridding approaches to needs

In Table I, the ability of approaches based on either solely structured or unstructured grids to meet each of the requirements is summarized.

From this table, it is evident that neither of the approaches in isolation is capable of meeting the requirements. However, an approach based on the use of both types of mesh, exploiting the qualities that each can deliver, can realize the objectives.

Focusing further, the aerospace industry has specific requirements, such as an accurate high Reynolds number capability for detailed design or performance analysis of an aircraft, which are not shared by other industries. These requirements place a heavy emphasis on solution accuracy (due to the influence this has on the parameters on which a design is judged, i.e.

drag) and efficiency (due to the size of problems that need to be addressed). It is believed that in the design of a CFD tool for this purpose, these two objectives should take precedence over the others. As such, this drives a CFD strategy based on the use of structured meshes wherever possible, with unstructured meshes embedded locally to meet the remaining requirements.

This is the motivation and basis for our use of hybrid grids.

DEVELOPING A STRATEGY FOR HYBRID GRIDS

Initial findings on using hybrid grids

Early work. The idea of hybrid grid generation was first advocated in two quite distinct pieces of work [6, 7]. Nakahashi and Obayashi [6] observed that the approach 'combined both the computational efficiency of the finite-difference method and the geometric flexibility of the finite-element method'. Weatherill [7] observed that 'the real advantages of the structured approach are the disadvantages of the unstructured approach, and vice-versa. The combination of the approaches is an attempt to capitalise on the merits of both approaches'.

The need for the Pyramid. The work of Weatherill [7], in two dimensions, motivated studies into the use of hybrid meshes in three dimensions [8–10]. As part of these studies, the decision was made to develop a centred finite-volume cell-vertex-based flow solver. The natural extension to 3D of the use of the quadrilateral and triangle are the hexahedron and tetrahedron. Since these elements do not share a common type of face, the issue of how to treat their interface is raised immediately. To avoid the need for the flow solver to handle hanging nodes and edges in the mesh, the decision was taken to introduce an additional element, the pyramid, whose triangular faces could abut a tetrahedron and whose quadrilateral face would abut a hexahedron.

Initial flow results. Inviscid flow computations [8] were performed for the simple test case of a bump in a quasi three-dimensional channel. The inflow conditions were chosen to ensure that a shock would develop across the channel, above the bump. Two types of mesh were formed using the same point set; solely block-structured (BS) to act as a reference solution and hybrid (HY) to test out the approach. The unstructured (UN) region in the latter was embedded where the shock would form.

From this exercise, it was possible to draw a number of conclusions [8]. As the grids were refined, the solution on the BS and HY meshes tended towards the same answer. As the meshes were coarsened the accuracy of predictions on the HY grid deteriorated more quickly than on the BS grid. Comparing the computational requirements of a node in a BS and UN region of mesh, it was found that the UN node would typically require a factor of four times as much storage (this figure arises due to the use of a face-based data structure; the use of edge-based data structures act to reduce it). Furthermore, the time-stepping of the flow solution would only advance at approximately half the rate, due to the larger number of elements in a given volume.

In a following study [9], both a BS grid and HY grid were formed for a wing-foreplane-fuselage, with the latter grid containing tetrahedra embedded around the foreplane. It was found that the isotropic UN surface grid on the foreplane (required to achieve good field

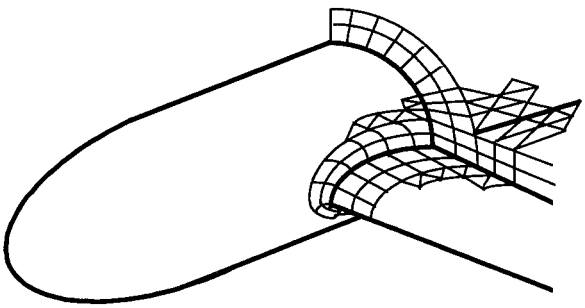

Figure 1. Local block-structured grid for modelling junction regions.

mesh quality) needed to be approximately 10 times as dense as the corresponding BS surface grid for the same predictions of surface pressures to be achieved.

The benefits of the prism. In considering the extension of hybrid grids to viscous flows, it was believed that there would be benefits to be realized, in terms of computational efficiency and accuracy, by introducing an additional element into a hybrid grid, namely the prism. This element could be formed by 'inflating' a surface triangulation, thereby maintaining a regular grid topology normal to the surface—hence, the terminology semi-structured grid. As such the computational efficiency of the prism could be expected to, and in practice does, lie somewhere between that of the hexahedron and tetrahedron.

To explore the potential benefits of using prisms, two studies were undertaken [10]. Firstly, both a completely UN grid and a hybrid semi-structured/unstructured grid were formed for an inviscid modelling of the flow over a wing. In both cases, the same surface grid was used, which for the latter acted as the initial boundary from which to advance layers of prismatic grid. The results achieved on the hybrid mesh were significantly better than the unstructured grid. A further investigation used the same point set in defining a BS grid and a purely prismatic grid to calculate the viscous flow over a semi-infinite wing, formed by stacking RAE5225 aerofoil sections. The agreement between results on these meshes was excellent, even for quantities such as skin friction.

The results of the two studies emphasized the importance of the grid topology normal to the surface, and supported the findings of earlier works in two dimensions [11, 12].

The need for local structured grids. Having accepted the need for prismatic elements, the question arises as to how to grid the junction region of two intersecting surfaces, both of which have prisms grown from them. The regular topology of the grid normal to each surface, motivates the use of local structured grids in these junction regions, whose quadrilateral faces naturally abut onto the prism, see Figure 1.

Efficiency of hybrid mesh elements. In Table II, a comparison is given of the approximate relative computational efficiency per node of the main elements of a hybrid mesh. The quoted memory factor is based on the total storage required for a typical flow run and not just the memory required to store the grid.

Table II. Comparison of the approximate relative computational efficiency.

	H	P	T
Time step efficiency, T	1.0	0.75	0.50
Work per time step, W	1.0	1.50	3.00
Efficiency factor $= T/W$	1.0	0.50	0.16
Memory factor	1.0	1.50	2.00

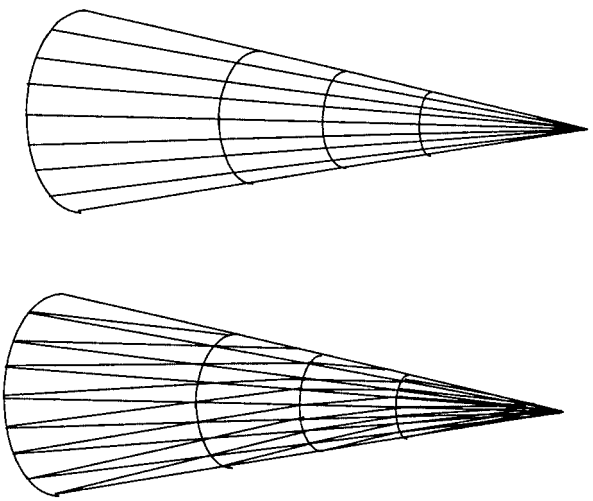

Figure 2. Geometries requiring anisotropic grids.

It is clear from this data that tetrahedra should only be used when strictly necessary. This is particularly true in shear layer regions, where the required point density of the mesh normal to a surface will be independent of the grid type used. Put another way, the potential advantage of unstructured grids for coarsening meshes away from surfaces cannot be realized in the shear layer regions without the likelihood of compromising the flow solution.

Efficient, High Quality Grids. The discussion to date has indicated that unstructured grids can be readily applied to the generation of meshes about arbitrary shapes. Whilst this statement is true, it is somewhat simplistic since it ignores the fact that high grid quality is often an essential pre-requisite to an aerodynamic simulation. Indeed, there are a number of situations where it is difficult to generate high quality, efficient, unstructured surface grids, and significant effort needs to be expended in an attempt to achieve this. Even then, suitable field mesh quality is not always readily attained.

In Figure 2(a), a sketch is given of a situation where a high quality, efficient mesh is more readily achieved with a structured grid than an unstructured grid. The key point is that it is far easier to grid regions that require anisotropic mesh using structured rather than unstructured grids. This is particularly true where a geometric surface degenerates to a point. Thus, for the cone, the high circumferential surface curvature is all that requires resolution by a dense

region of mesh. A further example would be a flap end plate, where the flow gradients will be higher in the cross-flow plane than the axial direction. If an isotropic unstructured grid were generated in these regions, the density of the mesh required to model the flow accurately would impact significantly on the computational resource needed, especially when the data given in Table II is considered.

Summary from studies of hybrid grids

From this discussion, the following conclusions can be drawn:

(i) The use of hybrid grids is a valid strategy for meeting the design requirements of modelling viscous flows over aircraft.
(ii) A hybrid mesh should contain hexahedra, prisms and tetrahedra. Pyramids provide a way of avoiding hanging nodes and edges.
(iii) The order of priority that should be given to these elements should be based on maximizing the use of structure wherever possible, subject to achieving a certain level of element quality efficiently.
(iv) Meshes should be regular in a direction normal to a surface, particularly for Navier–Stokes simulations.
(v) The choice of mesh types for a region should take into account anisotropy of surface curvature and flow gradients.
(vi) The unstructured regions of mesh will have to be more dense than the structured regions to yield comparable accuracy.

HYBRID GRID GENERATION IN SAUNA

The attraction of hybrid meshes is clear. However, the task of developing a readily useable system based around the concept is significant. In recent years, much effort has been expended in developing a hybrid mesh generation capability for the CFD system SAUNA (structured and unstructured numerical analysis). The system is capable of forming either solely block-structured, semi-structured or unstructured grids. In addition, it is capable of forming a hybrid combination of any of these mesh types. Hence, the same system can be used to form meshes efficiently for problems as diverse as the steady, viscous flow around a civil aircraft, or the unsteady inviscid flow over a store released from a carriage bay.

Numerous papers [13–18] have been written detailing the generation of the meshes, so only a summary is given here. The mesh generation procedures are described in their order of use, starting with the most constrained problem, and working through to the least constrained. The important area of interfacing the different elements is then covered. Finally, a description is given of a technique that allows hybrid grids to be regenerated automatically in response to a geometry perturbation driven by a design optimization tool.

Structured grid generation

The topology of a structured grid is known to have a significant impact on the performance of a flow solver, with the best results achieved when a match is found with the geometric characteristics of the surface being modelled. For domains defined by multiple surfaces, this grid property can only be achieved, if at all, through the use of block-structured grids. In this

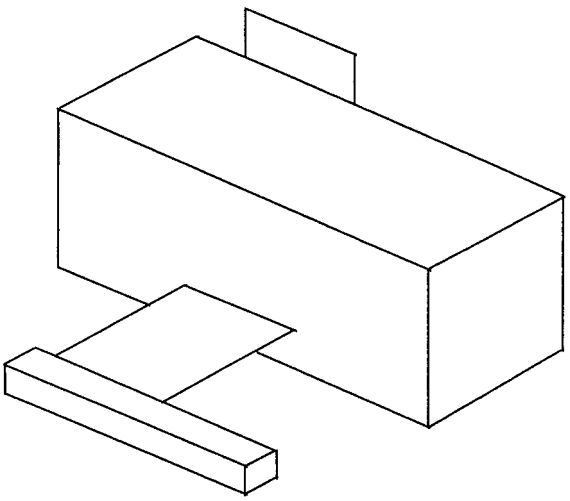

Figure 3. Schematic representation of combat aircraft.

approach [13], the domain is subdivided into non-overlapping regions, each of which has its own locally structured grid and is topologically equivalent to a cube. Since a domain may need to be split into many hundreds of blocks, the task of forming a mesh has some clearly identifiable potential bottlenecks—the division of the domain into blocks and the generation of a grid of suitable quality within each block.

To alleviate the requirement to partition the domain manually, a semi-automatic procedure has been developed in which user interaction is limited and yet sufficient to determine the final grid topology [13, 14]. The real domain is replaced by a schematic representation in which the surfaces that bound the domain are depicted as either of two topological entities, namely planes or boxes. This is illustrated in Figure 3 which shows a schematic diagram for half of a combat aircraft (which itself is shown in full in Figure 4) with the intake faired over. Having reduced the domain to one which merely has many planar rectangular boundaries, a Cartesian grid topology is readily created. Algorithms are then invoked on request to create local 'C' and 'O' grid topologies around the topological entities.

Rather than viewing the development of a capability to generate a block-structured grid as creating a process which will be applied to form each block in turn, the approach adopted is to form the entire grid from a single process. To this end, the grids are generated as the solution of a set of elliptic partial differential equations, with points which lie on internal block boundaries evolving as part of the solution, rather than having to be predetermined. The location of points which lie on the domain boundaries comes from a preceding surface mesh generation phase.

The generation of surface grids is potentially dominated by the time taken to establish point distributions on boundaries. With this in mind, algorithms have been developed [14] to generate default boundary point distributions, which can be readily adjusted to meet particular requirements—desired size of trailing-edge cell, for example. Figure 4 shows an example of a surface grid generated on a combat aircraft.

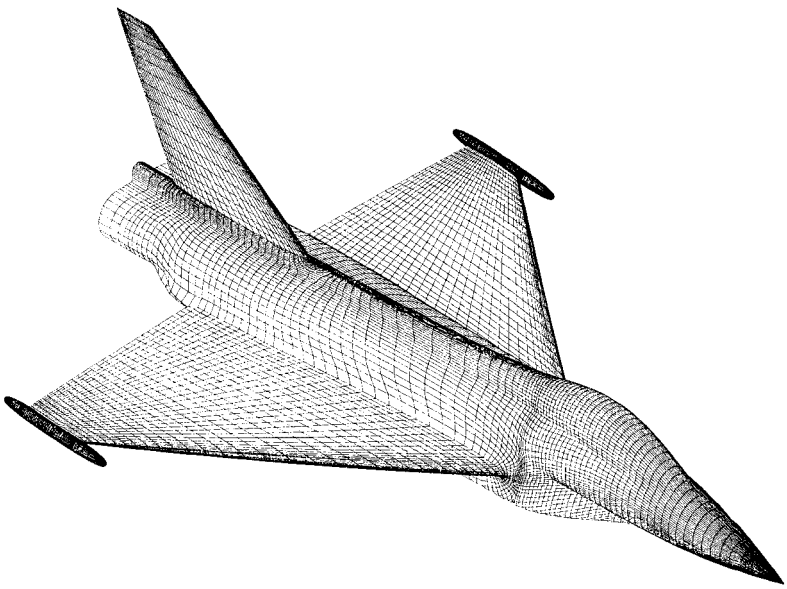

Figure 4. Surface grid for combat aircraft.

Semi-structured grid generation

In Figure 5, an example of a prismatic mesh is given. The technique [15] employed for generating prismatic elements is a marching method, similar to the hyperbolic pde approach, and as such starts from a defined surface and propagates outwards to an outer-boundary, the exact shape or location of which cannot be predetermined. The prismatic grid is built up one layer at a time. At each stage, the positions of points in the next layer are determined as a function of the current outer grid surface, which will initially be the input unstructured surface grid. The generation of a prismatic layer can be separated into two distinct processes. The evaluation of normal vectors and the determination of marching distances along these vectors.

The first stage of the prismatic grid generation process is the determination of marching direction vectors at all points on the unstructured surface. This is achieved by evaluating the normals (to the plane each triangle lies in) to all surface triangles and sending contributions to the forming nodes weighted by the angle subtended at the node. All nodal vectors are then normalized to unit magnitude. This yields an approximately normal marching vector for every point on the current grid surface. However, if these vectors are used in this form, the normal grid lines will converge from concave surface regions leading to grid crossover. This undesirable feature can be overcome by an iterative smoothing of the vectors using a Laplacian filter.

The marching distance along a vector is determined as the desired spacing for the layer multiplied by a distance variation function. The goal of this function is to compensate for regions of high concave and convex curvature, increasing marching distances in the former case and reducing them in the latter. The overall effect is that the shape of the grid layers tends towards a sphere as distance from the geometric surface increases.

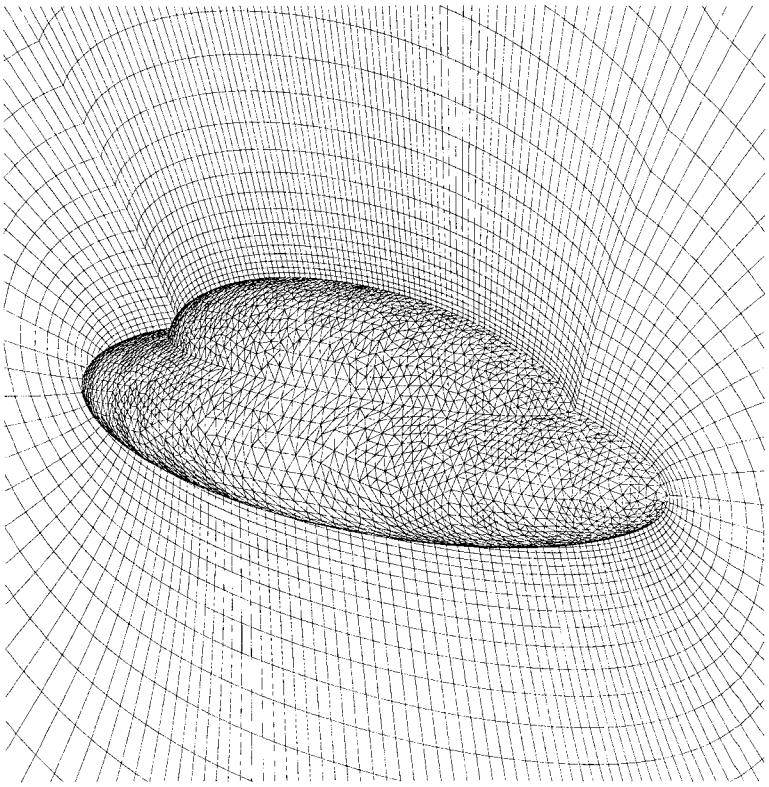

Figure 5. Prismatic grid for double ellipsoid.

Unstructured grid generation

The unstructured mesh generation is performed in two stages, surface grids followed by volume grids. For the former, the generation of grids that are independent of the geometry definition has been a particular focus of effort [16], for the latter, the problem of boundary integrity requires careful attention [17].

Separate meshes are formed for each surface of the configuration and for the boundary of the domain. For each, boundary point distributions are defined in a graphics-based working environment, with the boundary lines delimited into segments to facilitate precise control over distributions. To be consistent with the creation of a high quality field mesh, it is required that the surface meshes consist of triangles which are approximately equilateral in physical space. To this end, a pseudo-Delaunay surface triangulation procedure has been developed [16], which is coupled to an algorithm to determine the location of points. Control of grid density in regions of high surface curvature is assured through the solution of an optimization problem based on determining a desired edge length distribution. Figure 6 shows a typical surface grid for a missile layout.

Techniques for forming unstructured field meshes generally fall into one of two categories, either based on the Advancing-Front [3] or Delaunay algorithms [19]. In the first approach,

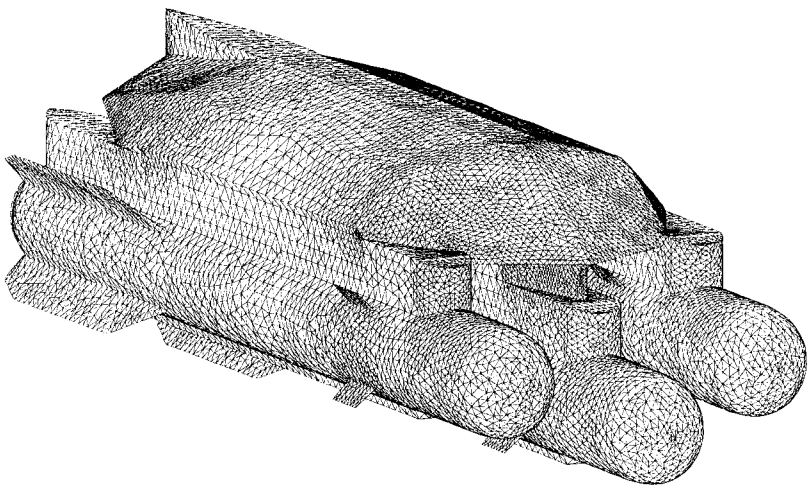

Figure 6. Unstructured surface grid.

a front is defined, which represents the entire boundary. For a selected triangle on the front, a field point is added in as near optimal a position as possible, a tetrahedron formed and the front updated. The mesh evolves by advancing from boundaries into the interior, yielding high quality elements near surfaces. In the second approach, a point set is triangulated, and points sequentially added interior to the domain until mesh quality tests are satisfied. It is more rapid, but yields its poorest mesh quality at boundaries, in part due to methods used to ensure that the triangulation is boundary conforming. This latter issue is of particular importance in the generation of hybrid meshes in which the tetrahedral elements are the last to be formed. Here, standard techniques of edge-swapping and point addition cannot be considered since the unique facial abutment of tetrahedra to pyramids and prisms would be corrupted.

With these factors in mind, a dual approach has been developed [17] for forming the field mesh. The boundary grids and their connectivity are input and a Delaunay triangulation created. No attempt to preserve boundary integrity is made. The mesh is then advanced one element at a time away from the boundary, and the Delaunay triangulation updated. This process continues until a time at which the Delaunay mesh naturally conforms to the continually updated boundaries of the advanced grid. The remainder of the mesh is then created using a standard Delaunay approach. High-quality boundary-conforming field meshes are thereby achieved in an efficient manner.

Interfacing different grid types

The interfacing of the different elements of a hybrid grid represents a significant component in the development of a hybrid grid generation system, which must be performed in an automatic manner. Different issues are involved in the treatment of each type of element-to-element abutment. These are discussed in the order in which they are met in forming a hybrid mesh.

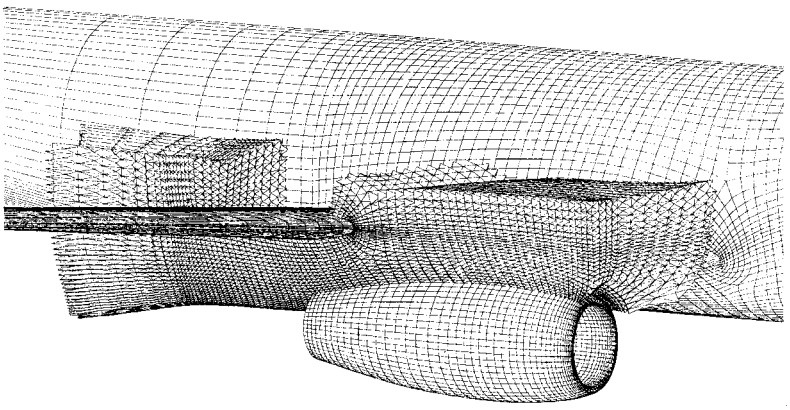

Figure 7. Structured/unstructured interface: triangular faces of pyramids.

Structured/semi-structured. At the interface of block-structured and prismatic grid regions, the quadrilateral faces of the elements must abut. This means that all the points on the interface are fixed points to which the prismatic grid generator must conform as the layers are formed. To make the transition from block-structured to prismatic grid as smooth as possible, the vectors resulting from the fixed boundary points are used in the smoothing process for the normals in the prismatic region [15]. This has the effect of preventing any sharp changes of direction near the interface. To obtain a representative marching distance for the prismatic grid, a Laplacian equation is solved for each layer, with the block-structured grid providing the necessary boundary data.

Structured/unstructured. Figure 7 shows such an interface for an unstructured region embedded around the pylon of a civil aircraft. The interfacing of these regions of grid has already been covered in motivating both the use of an additional element, the pyramid, and the approach taken to achieve boundary integrity in the unstructured field grid. The latter removes the need to create a buffer interface of tetrahedra, which is subsequently modified, as was required in earlier work [18].

Semi-structured/unstructured. There are three principal factors which govern the ideal extent of the prismatic region, the first two of which place a lower limit and the third an upper limit on the extent of the prismatic region:

(i) The grid should extend to a distance where viscous effects become negligible. With this in mind, it is necessary to define 'wake-plane' geometries that extend downstream of surfaces such as wings, and grow the mesh away from these surfaces in the same manner as other surfaces.
(ii) The cell aspect ratio (height/side length) should be as close to unity as possible to promote a smooth transition to the tetrahedral region (further techniques are used to adjust the marching distance function to achieve this).
(iii) The quality of the triangulation of the outer layer should be as good as possible to achieve a good quality abutting tetrahedral mesh.

In practice, however, it can prove difficult to advance meshes the required distance from concave boundaries. This is a problem frequently encountered in hybrid meshes composed only of prisms and tetrahedra and is indicative, in part, of the use of inappropriate elements in such regions. Again, the route taken to boundary integrity [17] negates the need for the special buffer elements employed in earlier work [15].

Remeshing hybrid grids

To conclude the discussion on mesh generation techniques, a method [20] to regenerate hybrid meshes automatically in response to design modifications is described. This method allows the CFD system to be executed efficiently within a constrained optimization design analysis system, CODAS [4], in which an initial geometry is perturbed many hundreds of times. For each new geometry, a new mesh is formed and the flowfield initialized from the previous solution.

The technique has two basic steps, which are each applied to the surface and field grids in turn. Firstly, a pre-conditioning step is applied which perturbs the initial reference mesh points as a function of their proximity to both the nearest moving and fixed boundaries. This has the effect that the mesh close to the moved/re-designed surfaces performs a rigid body motion akin to that of the surfaces, thereby maintaining element quality in this region. The maximum perturbation to the mesh occurs approximately halfway between the moved and fixed boundaries. The subsequent step of smoothing the mesh, heavily under-relaxed near boundaries, acts to remove poor mesh quality in these regions.

Figure 8 gives an example of the use of the technique to remesh a block-structured grid on a combat aircraft, following the deflection of the foreplane by 15 degrees. Importantly, the overall quality of the mesh is retained, an important feature in a design optimization strategy, where mesh dependency effects need to be minimized to avoid misleading conclusions being reached.

DATA STRUCTURES

The data structure that describes a hybrid grid to a flow algorithm is central to the success of the approach and its creation represents a major component in the development of a hybrid grid capability. Prime factors affecting the definition of a data structure are the discretization scheme adopted by the solver (these are given later) and the architecture of the platform used.

Node numbers, pointers and matrices

The nodes of the hybrid mesh are uniquely numbered, with all nodes at which a given boundary condition is applied stored contiguously in a predetermined hierarchy. These nodes are followed by the nodes that either lie on block interfaces or bound regions of differing elements. Finally, nodes that either lie inside each block, or solely within the unstructured or semi-structured field grids, are also stored contiguously in a ID array.

Throughout the formation of the ID array, pointers are established to identify the start and end of each group of data. A single integer is stored for each block to define the starting point for the storage of the interior nodes of the blocks. A pointer system, founded on the faces of each block, is used to access nodes that lie on block faces; these nodes may be either part of more than one block, or be part of other elements. These data are used to form so-called

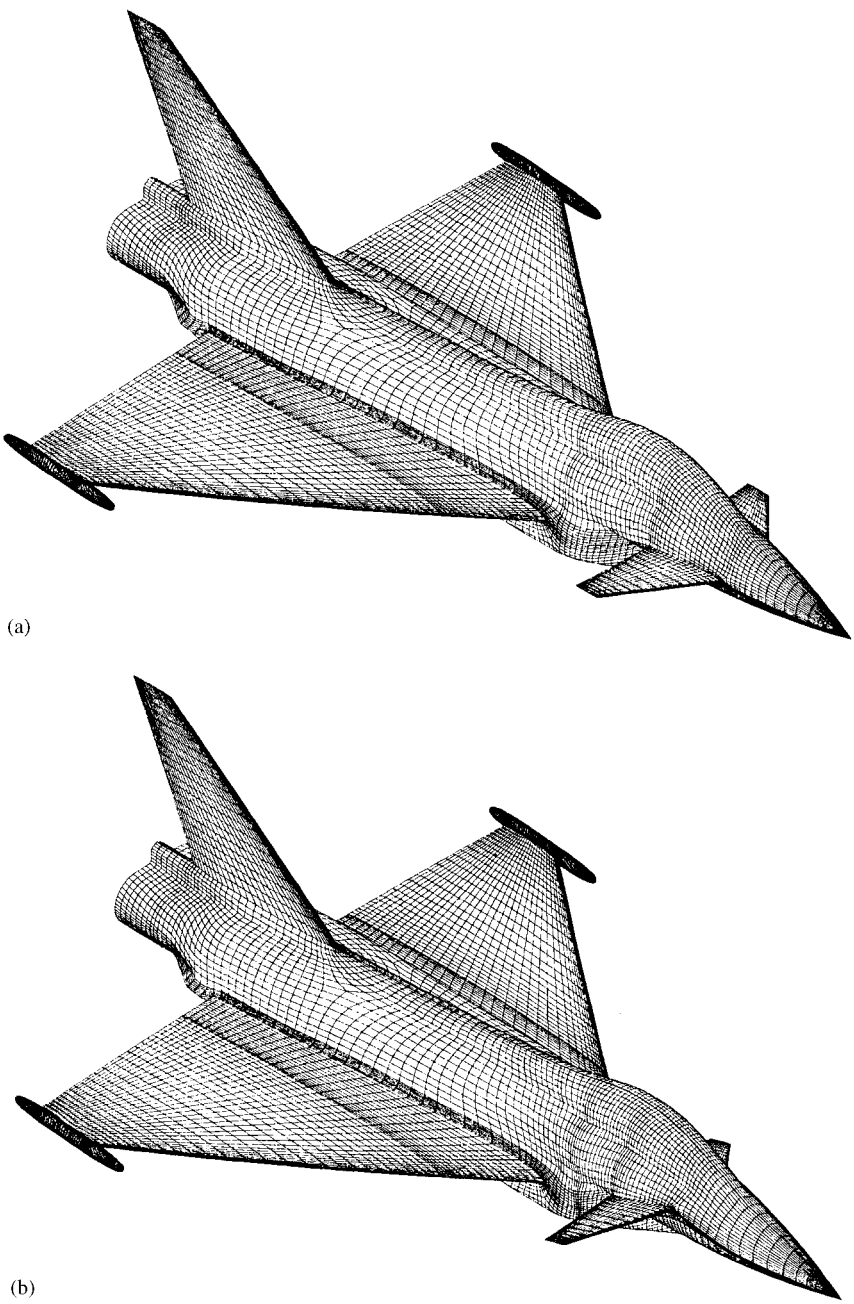

Figure 8. (a) Initial grid; (b) remeshed grid.

'working blocks' which allow the regular I,J,K notation to be employed when processing each block, rather than explicitly storing the connectivity of hexahedra. This ensures that the efficient memory storage and access patterns associated with structured grids is maintained.

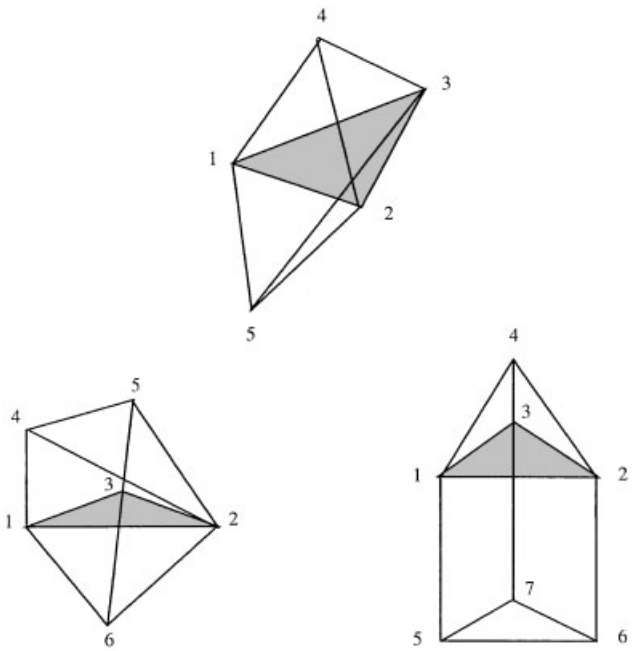

Figure 9. Connectivity matrices.

Connectivity matrices, see Figure 9, are used to describe the join of faces of tetrahedra to the triangular faces of either other tetrahedra, or prisms, or pyramids. Similarly, for the prisms, the unstructured surface grids are stored in edge-based connectivity matrices, with surface node-based pointers used to define the nodes lying along the lines of off-surface structure in the grid. All edges in the unstructured grid, and on the boundaries of both blocks and regions of semi-structured grid, are stored also.

Domain decomposition for parallel computations

The use of the SPMD programming model to facilitate parallel processing, whether for an MPP machine or a cluster of workstations, is accepted as a means of reducing the elapsed time of CFD simulations [21]. To be effective, it requires the mesh to be partitioned and mapped to processors in a way that accounts for both the communication overheads and the performance and memory associated with the available processors (typically, this is very different for MPPs and for different workstation networks).

The decomposition is performed by optimizing an initial partition derived by recursive geometric bisection (RGB) via simulated annealing [22]. To this end, a graph consisting of nodes (representing blocks and unstructured grid points) and edges (representing block faces and unstructured edges) is used to describe the grid, with weights for each node and edge used to indicate the computational cost of each entity. A simple user-defined array describes the performance characteristics of the target architecture.

Random perturbations [22] are applied to each current best solution and a cost, based on execution time, calculated. The perturbations are applied primarily to the decomposition, either

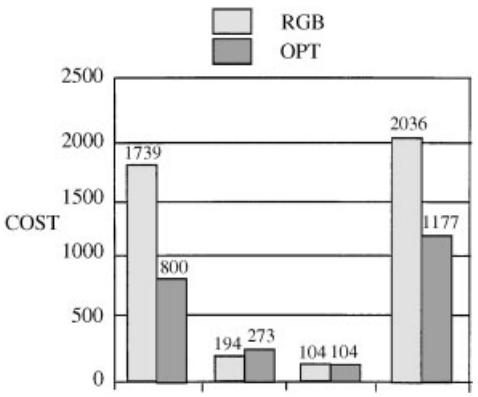

Figure 10. Comparison of run times for alternative decompositions.

by moving a graph node to a different processor or by randomly selecting an interface edge (an edge with its end nodes on different processors) and moving one of its end nodes to the same processor as the other.

The new solution is accepted or rejected based on a dynamic tolerance to changes in cost, this continuing until a stationary solution is attained. The optimizer defines a node-to-domain and a domain-to-processor mapping for the grid and target machine. In Figure 10, the benefits of adopting this approach are summarized by comparing the execution times for the flow solver (COST) due to key elements of the standard RGB decomposition and the optimized decomposition.

Domain assembly and grouping for parallel and vector processors

Once the blocks that are to reside on a given processor have been determined, an algorithm is employed which groups them together into as few larger blocks as possible, the aim being to reduce the overall surface area over which communication is required. Reductions in memory requirements and processing time for the flow solver are also gained. The algorithm effectively renumbers the nodes in the block-structured region of the mesh, requiring the block-based pointers and connectivity matrices to be updated.

The same algorithm is also employed for vector processors. In this case, the requirement is to make the vectors (over which operations such as the flux balancing are performed), as long as possible, which indicates that the blocks should be as large as possible. The only restriction on the combination of blocks is that their union must be topologically cuboidal. Further benefits can be realized by grouping the connectivity matrices defining the unstructured regions of grid in a manner which allows loops over these matrices to vectorize (colour coding).

Improvements for RISC processors

RISC processors yield their optimal performance when data can be accessed from cache rather than main memory. In an unstructured region of mesh, neighbouring nodes are usually numbered in an effectively random manner, which means that when a flux evaluation is

performed, there is a low expectation that all of the data required will be in cache. The probability can be enhanced significantly by renumbering the grid to reflect the spatial locality of the points. Consistent with this, the face- and edge-based connectivities also need to be renumbered. Execution times for a flow solution based on a totally unstructured mesh have been reduced by a factor of four using this approach [23].

Multiple grid levels

Flow solver efficiency is greatly enhanced through the use of multiple grid levels in the solution process, either through the use of full multi-grid or a simple grid sequencing strategy. Put in its simplest form, grids coarser than the datum fine grid are created by reducing grid density in all directions. This is a simple process in the structured grid regions, where coarsening implies removing every other point in each direction within each block. A block is allowed to have only a single cell in any direction at the coarsest grid level. In the unstructured grid regions, the strategy used is to create coarser, unrelated meshes, wherein points are not common between grid levels. This route has been chosen in preference to an agglomerated approach, to avoid any data structure extensions related to complex element shapes.

Best practice is always to create three levels of grid—coarse, medium, and fine—as this appears to optimize the flow solver solution procedure. A complete data structure is created for each grid. In addition, files are created which contain influence coefficients (weights) for transferring data between grid levels in the unstructured grid regions. This process can operate efficiently in structured grid regions within the flow solver without any pre-processing.

FLOW SOLVER

The flow solver used on the hybrid grids described above has matured over a period of years, through additional functionality, the inclusion of efficiency and robustness-enhancing measures and confidence gained by detailed validation studies. Some of the early decisions that are made in the genesis of any solver, however, are difficult and costly to change (particularly with regard to the spatial discretization), so it is instructive to reflect briefly on the decisions made at this stage for the flow solver in SAUNA.

The first decision was to develop a finite-volume vertex-based solver rather than a cell-centred one. It was always anticipated that regions of less than ideal grid quality would be present in a general hybrid grid (the regions where different elements interface were an early concern). Fundamental studies available at the time on truncation error analysis [24, 25] had shown that vertex-based schemes were inherently more accurate, particularly on unstructured grids and particularly on grids with rapid changes in element size.

The second fundamental decision was to develop generalized discretization procedures which could be applied to all regions of grid. This route would guarantee continuity of discretization across grid region boundaries, whilst foregoing, for example, optimum dissipation operators in structured grid regions. This does not preclude making use of any inherent structure of grid regions in efficiency-promoting measures, and earlier discussion on data structure issues has highlighted this aspect.

Spatial discretization

To go into more detail, the flow solver addresses the Reynolds-averaged Navier–Stokes (RANS) equations. The spatial discretization procedure is a flux balancing approach, with fluxes being accumulated around a control volume, which is defined as the union of all of the elements which meet a given vertex. This accumulation is a straightforward summation of contributions, giving an essentially centred (rather than upwind) approach. The primary target flow regime of the flow solver is for free-stream Mach numbers up to and including supersonic, but not hypersonic, which is consistent with such an approach.

The above procedure applies to the inviscid terms in the equations of motion. The viscous term contributions are accumulated around a control volume formed from the centroids of the elements which comprise the inviscid control volume. This is a natural approach given that the first derivatives required for the stress tensor evaluation are computed on an element basis, using Green's theorem, and are located at element centroids. The existence of separate control volumes for inviscid and viscous terms, whilst perhaps lacking in elegance, is not currently thought to be disadvantageous in any practical sense, nor does it affect the conservation properties of the scheme.

Artificial dissipation

The discretization of the RANS equations is augmented by artificial dissipation terms to provide a shock capturing capability and to eliminate undamped modes from the centred approach used here. These terms are of Jameson type being composed of a blend of second and fourth difference stencils. The adoption of a general discretization procedure implies the use of a 33-point stencil in a block-structured region. This derives from a two-stage construction, firstly the accumulation of edge differences at a vertex to compute a second difference term, and secondly the accumulation of differences of this quantity along edges to derive a fourth difference term. The second difference term is multiplied by a sensor based on second difference of pressure. Each difference in the final accumulation process is multiplied by an edge-computed scaling factor, based on grid aspect ratio. This gives an anisotropic scheme in the highly compressed grid regions needed to model shear layers. This aspect is critical to the accuracy of predictions in such layers.

Additional 'smart dissipation' techniques are employed to limit the magnitude of these non-physical terms further and to prevent them from compromising the modelling of real viscous effects. These apply on and are local to no-slip surfaces where default dissipation stencils become one-sided and are therefore of lower order. An example of the impact of these additional techniques is given in Figure 11, where the skin-friction coefficient is plotted for a representative aerofoil from solutions obtained using two alternative dissipation models. Model A is the default inviscid flow model and Model B includes the above surface modifications. The increase in overall skin friction levels with the latter model is marked, giving a more accurate prediction for skin-friction drag.

As every CFD developer knows, the precise details of the artificial dissipation or smoothing techniques used in a numerical scheme are one key to its success. The fine balance between accuracy, efficiency and robustness is controlled by the smoothing. A 'minimum' smoothing level scheme will be accurate, but may converge slowly and may diverge for cases more extreme than those used in the tuning process. A 'heavy-handed' smoothing scheme will converge quickly and will be highly robust, but may suffer a great deal in accuracy terms.

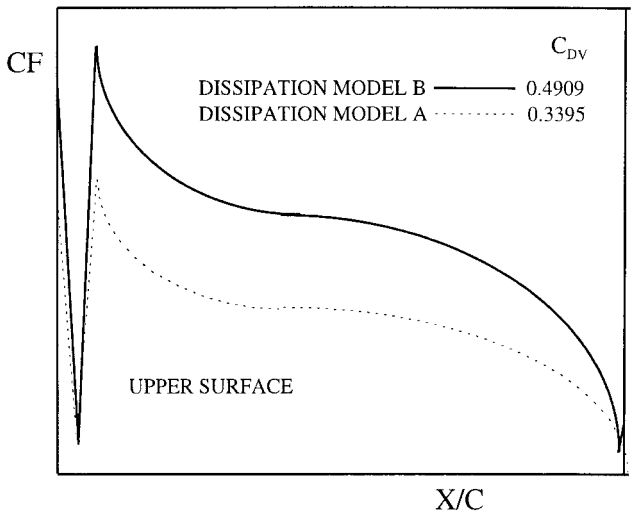

Figure 11. Effect of dissipation on skin friction.

The balance chosen here, as has already been strategically defined, is to favour low smoothing levels to ensure that accuracy requirements are met. A high level of robustness has still been achieved, however, through careful construction of dissipation operators coupled with grid quality demands.

Time integration

Explicit time-marching techniques are used to advance an initial flowfield to a steady state solution of the governing equations. These include Runge–Kutta time-stepping with residual smoothing and a full multi-grid strategy. Convergence criteria based on one or more force coefficients on chosen parts of the configuration becoming stationary are normally used. These techniques give an efficient scheme, as required. An incompressible flow option exists in the flow solver and in this case time-marching is based on an artificial compressibility approach. Details of the RANS algorithm are given elsewhere [26, 27].

Turbulence modelling

For viscous flow investigations a range of turbulence models are available. Although early studies made use of algebraic or single ordinary differential equation models such as Baldwin–Lomax and Johnson–King (these being the only ones implemented at the time), there is a recent strong trend towards partial differential equation models. The generality of implementation of these latter models (no profiles required), and their potential for modelling more complex physics, makes them a natural choice for a hybrid grid strategy. Two-equation models are now frequently used, with k-ω being the preferred option due primarily to its total non-reliance on normal-wall distances. Current state of the art in the code is full differential Reynolds stress modelling based on a multiscale model due to Wilcox [28]. The more advanced models are only available on structured grids at present. A number of difficult implementation

issues have had to be addressed with these advanced models, to maintain consistency with the design requirements of accuracy, robustness and efficiency. A well-chosen amalgam of techniques available in the literature and their further enhancement has been necessary in this respect [29].

Adaptivity

Mesh adaptation to either viscous or inviscid flow phenomena is performed using the LPE method of Catherall [30]. This involves the numerical solution of equations for node positions which are formed as a linear combination of an inverted Laplace equation, an inverted Poisson equation and an Equidistribution equation. The Laplace term promotes smoothness and orthogonality, the Poisson term enables the retention of favourable features of the initial mesh and the Equidistribution term controls the redistribution of nodes according to a measure of solution activity.

Post-processing

A discussion on flow solvers is not complete without making comment on the post-processing facilities which are available. Post-processing is the user-interface at the end of a CFD system and is therefore of key importance. Here again, aerospace makes some very specific demands in terms of the extraction of quantities of interest in the design environment. The usual data extraction techniques are available for hybrid grids, but a key area is that of drag prediction and particularly drag decomposition into its wave, vortex and viscous components.

Surface integration has traditionally been a principal means of drag extraction, although this approach does not permit decomposition of the drag. Near-surface methods are also applied to the SAUNA flow solver output. Shock-front, lifting line and boundary-layer momentum thickness methods allow extraction of wave, vortex and viscous drag, respectively, but their scope is limited due to both theoretical and practical constraints.

Recent research has been focused on far-field techniques, wherein a cutting plane is generated downstream of the configuration of interest and flowfield variables on this plane are used to extract drag quantities [31]. This methodology, in principle, allows the required drag decomposition and is applicable to geometries of arbitrary complexity.

EXAMPLES

In this section, the creation of hybrid meshes for use in performance analysis, store release and design studies is discussed. In all cases, the approach taken is to maximize the use of structured mesh, wherever this is consistent with attaining a certain level of mesh quality within a practical time-scale. Regions of the structured mesh are then removed and unstructured grids containing extra geometric surfaces embedded. The examples serve to demonstrate a number of scenarios in which it is believed the use of hybrid meshes out-performs any other approach. However, before discussing the examples in detail, further comments on the use of block-structured and unstructured grids are made, to support the use of hybrid grids.

Block-structured grids can be used to mesh a wide range of problems. The time taken to form a mesh for a configuration might be longer than an unstructured grid. However, this is

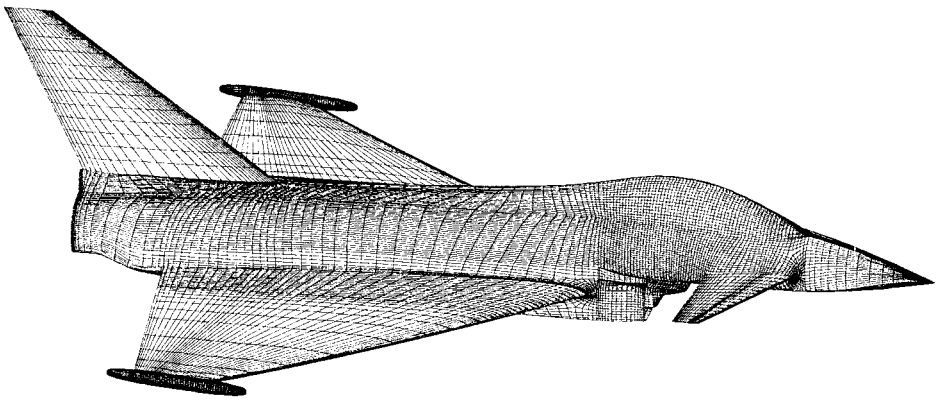

Figure 12. Block-structured grid on EF2000.

acceptable if the effort is seen to contribute, either in part or in whole, to the overall cost effectiveness of a task.

Such a scenario would be where either multiple release conditions for a store were being investigated, or where the release of alternative weapon assemblies is of interest. Parallels can be drawn here with a wind tunnel model of an aircraft, whose creation is justified, in part, because it will be used on more than one tunnel entry programme.

Furthermore, confidence in the structured grid can be established separately, and this will remain the same for all time. Thus, grid dependency issues between solutions can be minimized. Again, this has parallels with putting a wind tunnel model back into the same, rather than a different, tunnel.

The creation of a high quality, efficient unstructured field mesh for an aircraft is not as straightforward a task as promoters of unstructured grid technology might lead one to think. Indeed, the task is often an iterative one, involving intermediate flow runs before confidence in a solution is achieved. This situation arises due to the strong requirement to optimize the use of the number of available grid points, which is limited either by the memory, or speed, of the accessible hardware. Mesh adaptivity reduces this to some extent, but only providing the initial mesh is sufficiently dense to allow flow gradients to be resolved in the initial flow solution.

There can be little doubt that the amount of user input that will be required to generate efficient unstructured field meshes for viscous flows will be significantly greater than for inviscid flows. This arises due to the need to add directional information to ensure that suitable mesh compression is attained in shear layer regions.

Performance analysis

In Figures 12, 13 and 14, block-structured meshes are shown for powered military, transport and ASTOVL aircraft. It is believed that these types of configuration represent the very limit of geometric complexity that the block-structured approach can be applied to, both in terms of the time required to form the mesh and the acceptable quality of mesh that can be attained. For as the configuration gets more complex, so the probability increases that the block topology

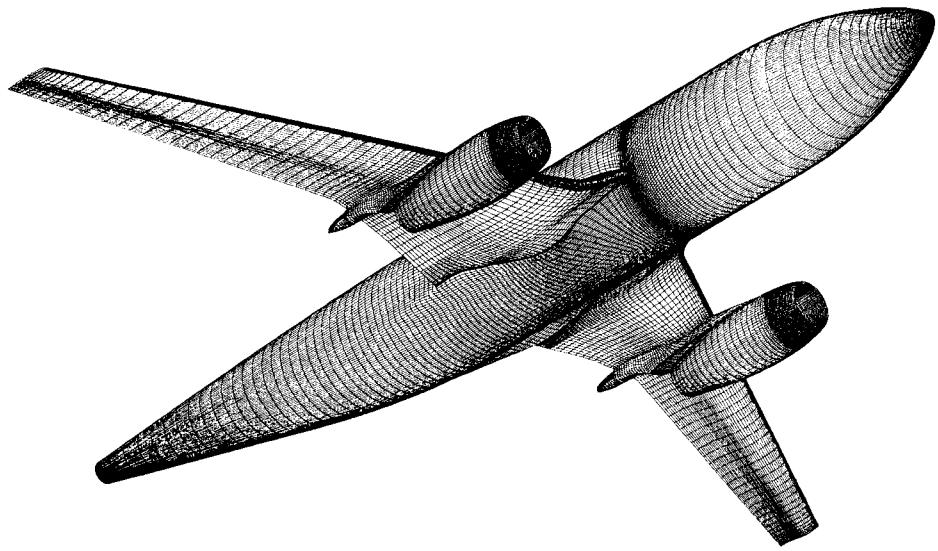

Figure 13. Block-structured grid on civil aircraft.

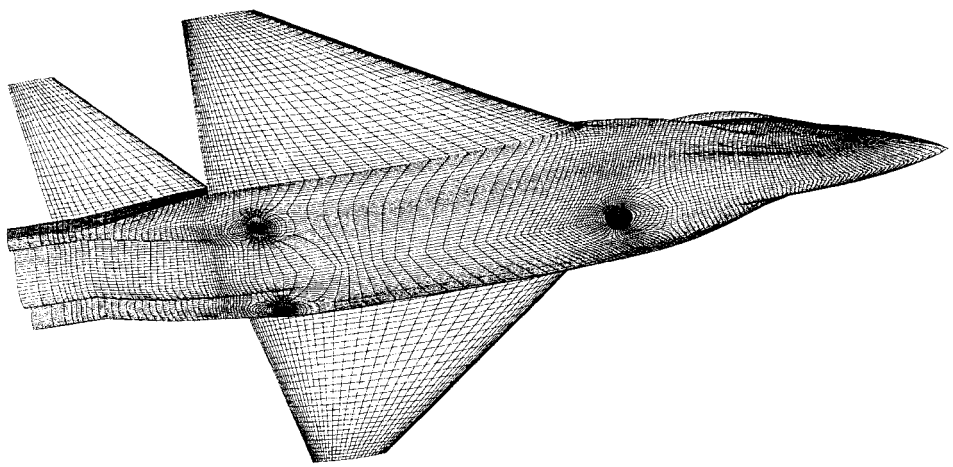

Figure 14. Block-structured grid on ASTOVL aircraft.

required in one region of the mesh will conflict with that in another. This conflict leads to difficulties in controlling mesh quality, which ultimately can affect the flow solution.

Examples of this are the modelling of the intake and pylon on the configurations given in Figures 12 and 13, respectively. In both cases, grids have had to be created in which complete block faces degenerate down to lines on the surface of the aircraft. This is equivalent to hexahedra collapsing to form prismatic elements with high aspect ratio triangular bases. The effect is not only that the grid quality is compromised, but also that the time-stepping of the flow solution is significantly impaired in this region.

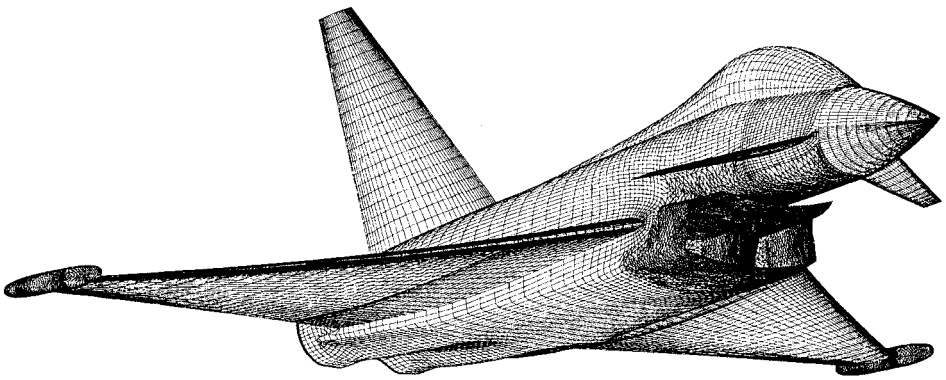

Figure 15. Hybrid grid on EF2000.

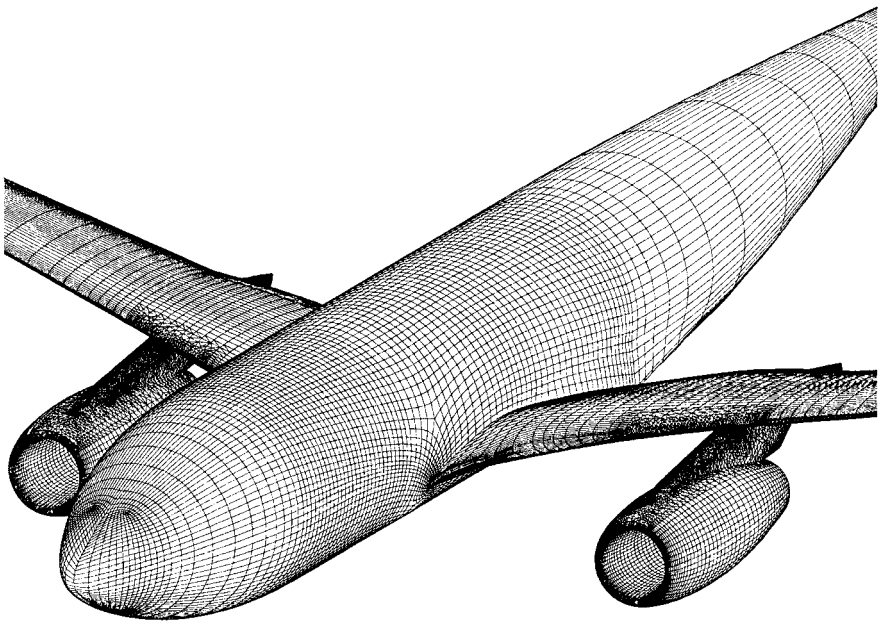

Figure 16. Hybrid grid on civil aircraft.

The fact that the grid quality deteriorates with configuration complexity impacts in two ways in the practical creation of block-structured grids. Firstly, it is more difficult to conceive either a grid topology, or a schematic representation of the aircraft to drive a topology generation algorithm. Secondly, special care has to be taken to control a mesh, this often necessitating extra geometric surfaces (through which flow can pass), to be specified in the field to restrain the freedom of the mesh.

Figures 15 and 16 show hybrid grids for the two configurations already shown in Figures 12 and 13. In the first case, a block-structured mesh was formed around the fuselage

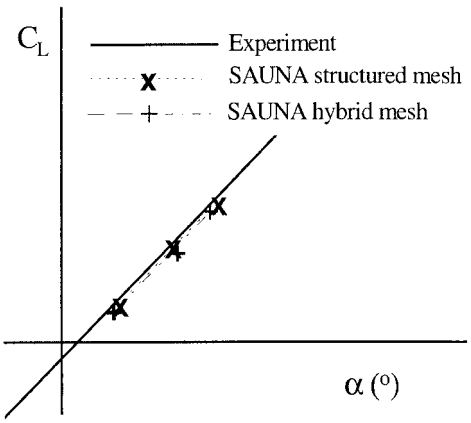

Figure 17. Comparison of theoretical and experimental data.

with the intake faired over. The intake geometry was subsequently introduced with locally embedded unstructured grid. In the second case, the block-structured mesh was formed around the wing-fuselage and through-flow nacelle. The pylon, which overhangs both the nacelle and wing trailing-edges was then added. In this case, the unstructured grid region is particularly well suited to the modelling of the highly three-dimensional pylon, and its complex junction intersections with both the nacelle and wing.

In both cases, the hybrid mesh required significantly less time and expertise to form the mesh than the block-structured grid. Furthermore, the general quality of the block-structured grid regions was higher, since the meshes were only being used in situations where they were readily controllable. Figures 17 and 18 show results from simulations on both mesh types for the military aircraft configuration. Results on the civil configuration are given by Hackett *et al.* [32]. Together, these serve to validate further the use of hybrid grids, in that the results compare favourably with those on solely block-structured grids.

Store release

Ensuring the safe clearance of a store from an aircraft is an area in which CFD is increasingly making an impact. Currently, the computational requirements for modelling the release of stores limit the flow simulations to being pseudo-unsteady and inviscid. Typically, a steady-state flow solution is obtained and the forces and moments acting on the store evaluated. These are then used in a six degrees of freedom (six DOF) model [33], which uses a four-stage Runge–Kutta scheme for the time integration, to predict a new position for the store.

During its trajectory, the store is free to pitch, roll, and yaw with respect to the aircraft. Solely block-structured grids are unable to accommodate the generalized location of one body with respect to another; even if they were, the resource required to create suitable topologies and grids is hardly conducive to the iterative nature of the complete simulation.

The flexibility of the unstructured grid approach overcomes these limitations of structured grids and is highly suited to a local refinement/coarsening of the mesh as the simulation

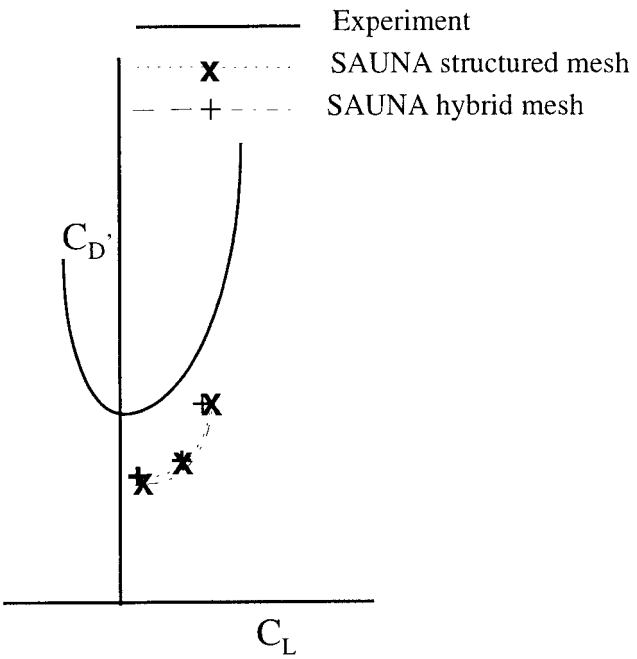

Figure 18. Comparison of theoretical and experimental data.

evolves. However, the use of unstructured grids throughout the whole flowfield rapidly imposes limits on the density of grid that can be utilized, in terms of both total memory limitation and an acceptable CPU time. Ultimately, this impacts on the accuracy of the simulation, and hence the value gained from the exercise.

A cost-effective programme of work on store release from an aircraft can be established based on the use of hybrid grids, in which a block-structured grid is used to mesh the majority of the domain and unstructured grid is used in the region covered by the trajectory of the store.

Figure 19 shows the initial hybrid grid generated for the release of 1500-litre fuel tanks from a Tornado. The block-structured grid took a week to generate, contained 2072 blocks and 690 000 cells. A region of this grid was then removed, covering the expected trajectory of the store, and an interface region of pyramids automatically generated; this would effectively act as the farfield boundary for the unstructured grid generation. Unstructured surface grids were then formed on the fuel tanks, shoulder pylon and part of the fuselage underside. The remainder of the volume mesh was then filled with tetrahedra.

The release of one of the tanks commenced with a flow solution obtained with the tank at end of stroke position. A transpiration boundary condition was applied on the surface of the store in an attempt to model the first order time dependent effects of the influence of the motion of the store with respect to the aircraft. The new position of the store was predicted from the loads acting on it, placing a requirement to regenerate the grid and reconverge the flow solution; this process continued until the store was judged to have cleared the aircraft.

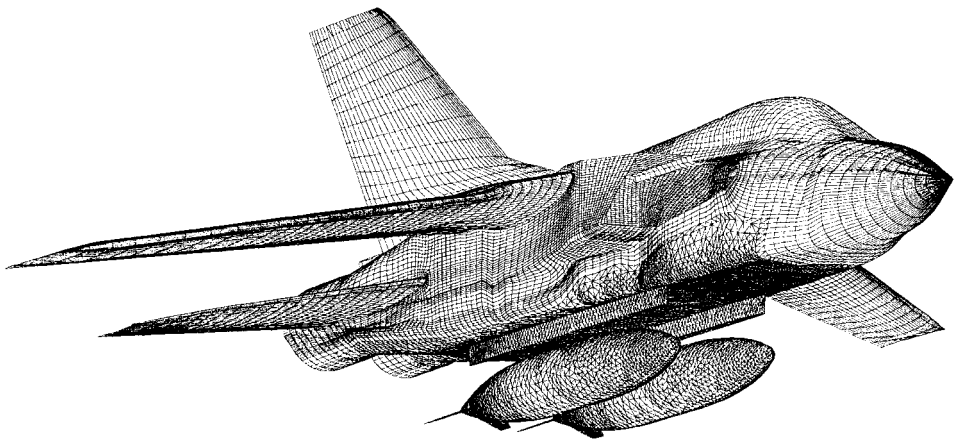

Figure 19. Hybrid grid on Tornado with 1500 litre fuel tank.

Clearly, the structured region of grid did not need to be modified, this in itself representing a significant computational saving over other approaches in which the whole domain is remeshed. The unstructured region was redefined using a three-step process. Firstly, a rigid-body pre-conditioning was applied which ensured that the mesh quality local to the store was maintained. Secondly, the mesh was smoothed to improve overall quality. Finally, local regions of mesh were enhanced by changing the mesh connectivity, either through edge-swapping, point addition or point removal techniques. Only when satisfactory mesh quality could not be achieved through this approach, which happens about once every five complete Runge–Kutta time steps, was the whole unstructured domain regenerated.

The flowfield is initialized for each simulation based on the solution at the previous time-step. The nodes in the structured region maintain their values, whilst interpolation processes are used to transfer the flow solution from the old to the new regions of unstructured grid. The overall benefits of these processes is to reduce the total run times of the simulations by a factor of about 20.

Figure 20 shows good agreement between theoretical predictions and flight trial results for the prediction of vertical displacement and pitch angle for the release of a store from an aircraft.

The reductions in CPU requirement that have been described make it possible to consider simulations based on the Reynolds-averaged Navier–Stokes equations, with the turbulence modelling needing to be limited in the first instance to cost-effective models, such as Baldwin–Lomax. In this case, local block-structured grids would be generated in the junction region of the tank with its fins. Semi-structured grids would be grown from the remainder of the tank and fin surfaces, conforming to the local structured grids, see Figure 21. These grid regions would remain fixed with the store throughout its trajectory. Unstructured grids would be used to fill the remainder of the domain. In this way, the grid topology away from the store surface is always regular, a fact that can be exploited in the flow modelling [10]. The efficiency of the grid around the store serves to bring forward the time when such calculations can routinely be undertaken.

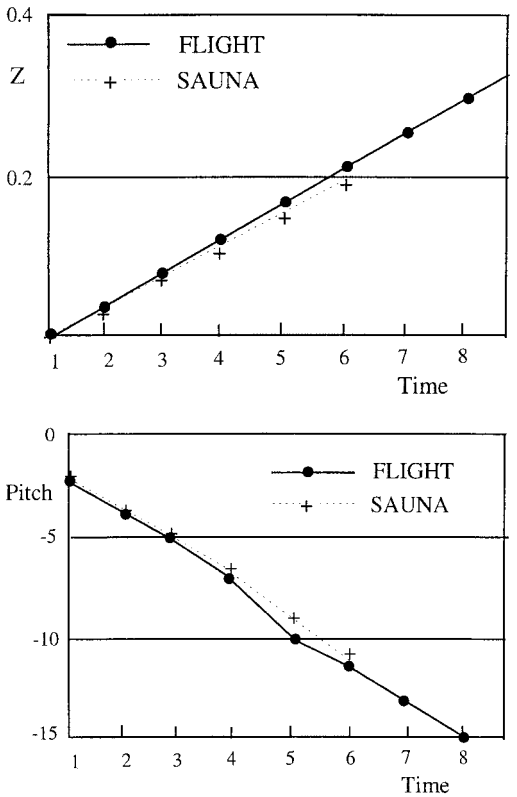

Figure 20. Comparison of flight data and theoretical data.

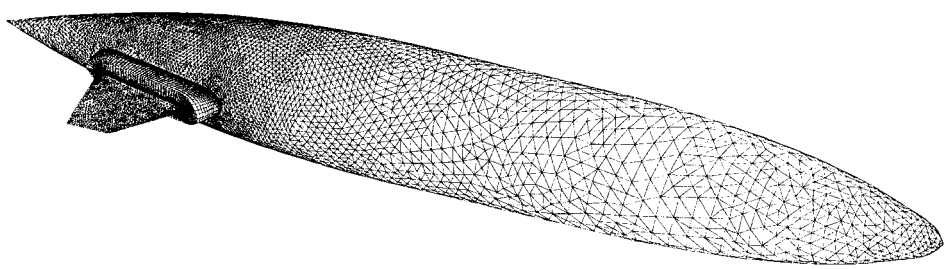

Figure 21. Use of local structured grid for viscous modelling.

Finally, the idea that the same reference block-structured grid can be used for more than one study is illustrated in Figures 22 and 23, where the same aircraft is shown carrying two alternative weapon assemblies. In the case illustrated in Figure 22, a smaller region of the block-structured grid was removed since carriage loads were all that was of interest. Here, the unstructured region is used to allow the geometric complexity of the assembly to be resolved, see Figure 6.

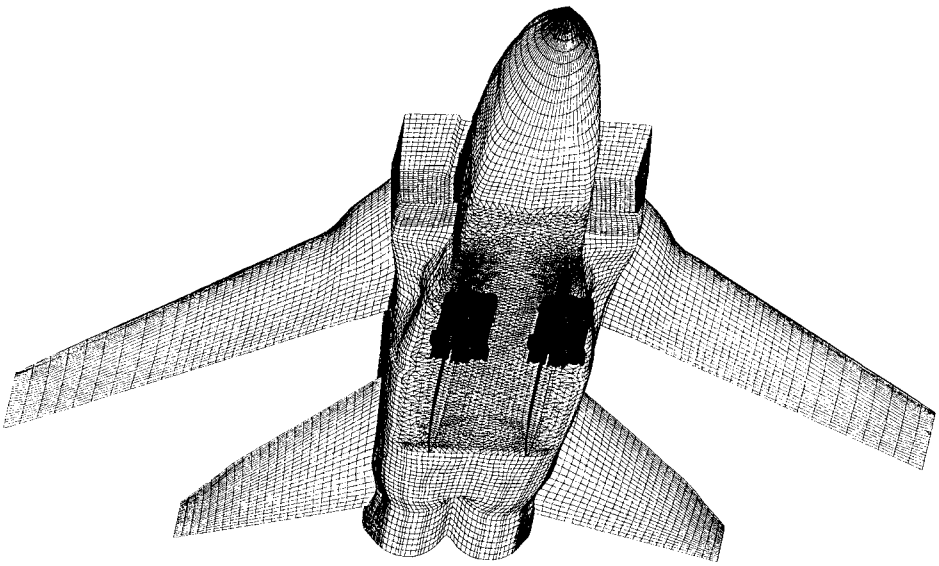

Figure 22. Hybrid grid for Tornado with missile configuration.

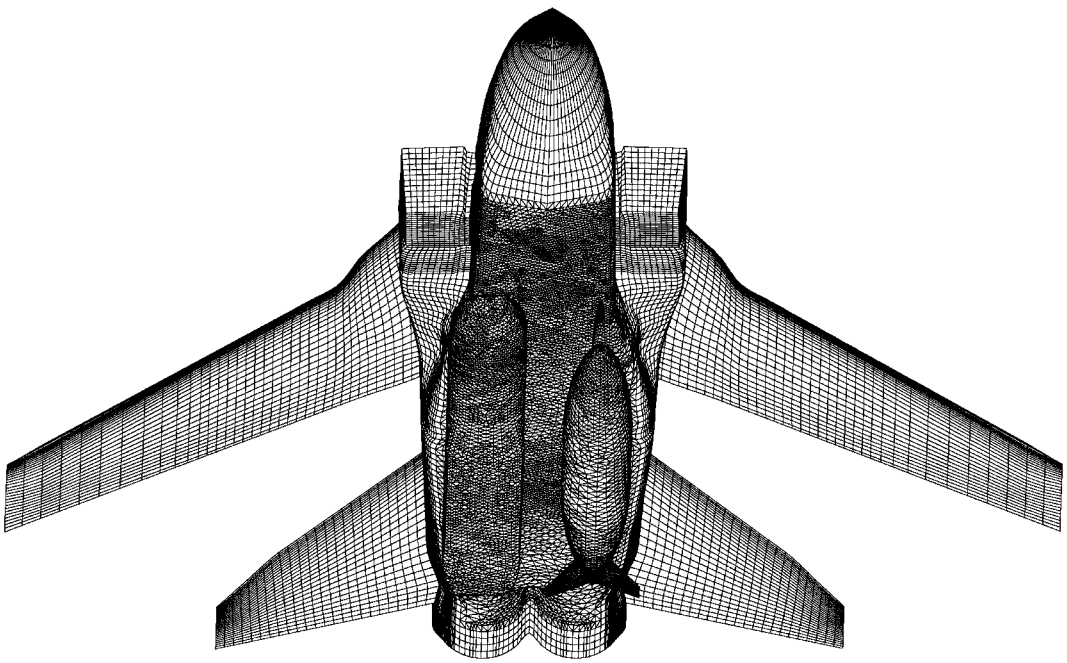

Figure 23. Hybrid grid for Tornado carrying alternative stores.

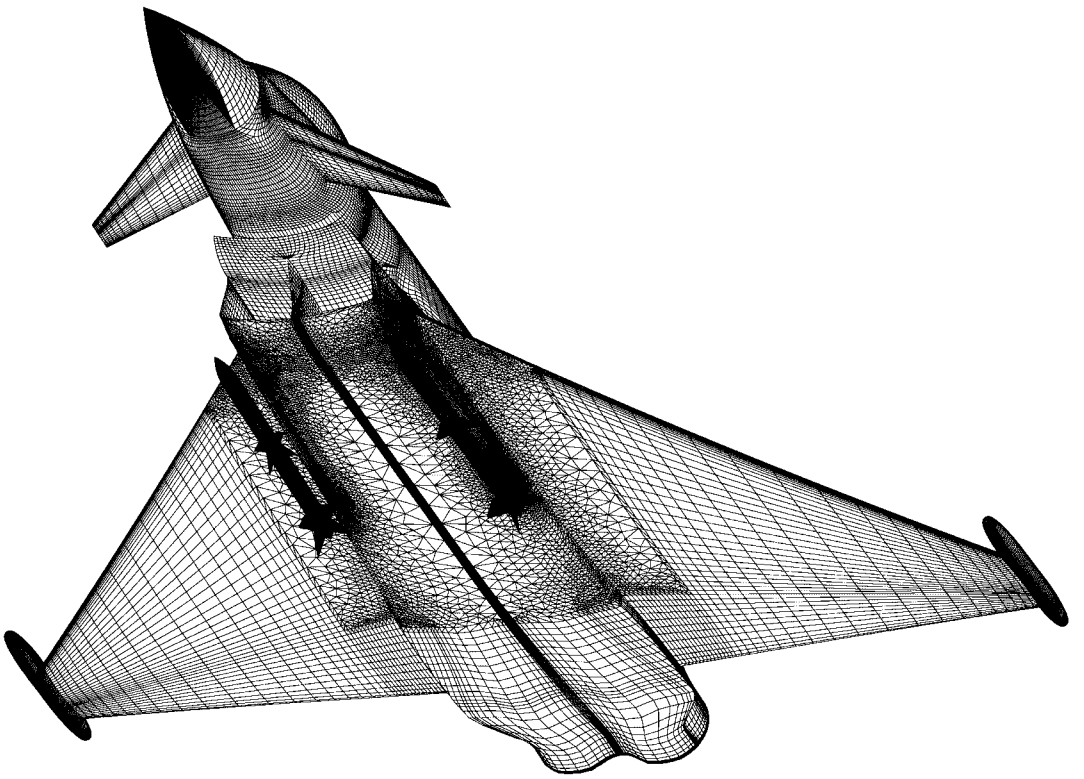

Figure 24. Hybrid grid for ejection of missile.

The idea is demonstrated also in Figures 24 and 25 for the ejection and rail release of weapons.

Design studies

Increasingly, aerodynamicists are interested in the integrated effect of component surfaces rather than consideration of a surface in isolation. Examples include blended wing-fuselage and underwing-pylon-nacelle design studies. One issue which affects the latter type of study is the nature of the engine cowl. To this end, flowfield simulations using both long and short cowls have been undertaken for a civil aircraft, with both the by-pass and core jets modelled, see Figure 26.

Use of the block-structured approach for the complete configuration was considered first. Whilst it was believed that this would be possible, concerns about the density of mesh that would be required were soon raised. As part of the creation of the background Cartesian topology, each surface of the nacelle would require to be represented by four planes—a top, bottom and two sides. Polar mesh topology would then be needed to be embedded around the interior and exterior of both the core and by-pass cowl and around the exterior of the central plug. This would lead to the creation of many thou-

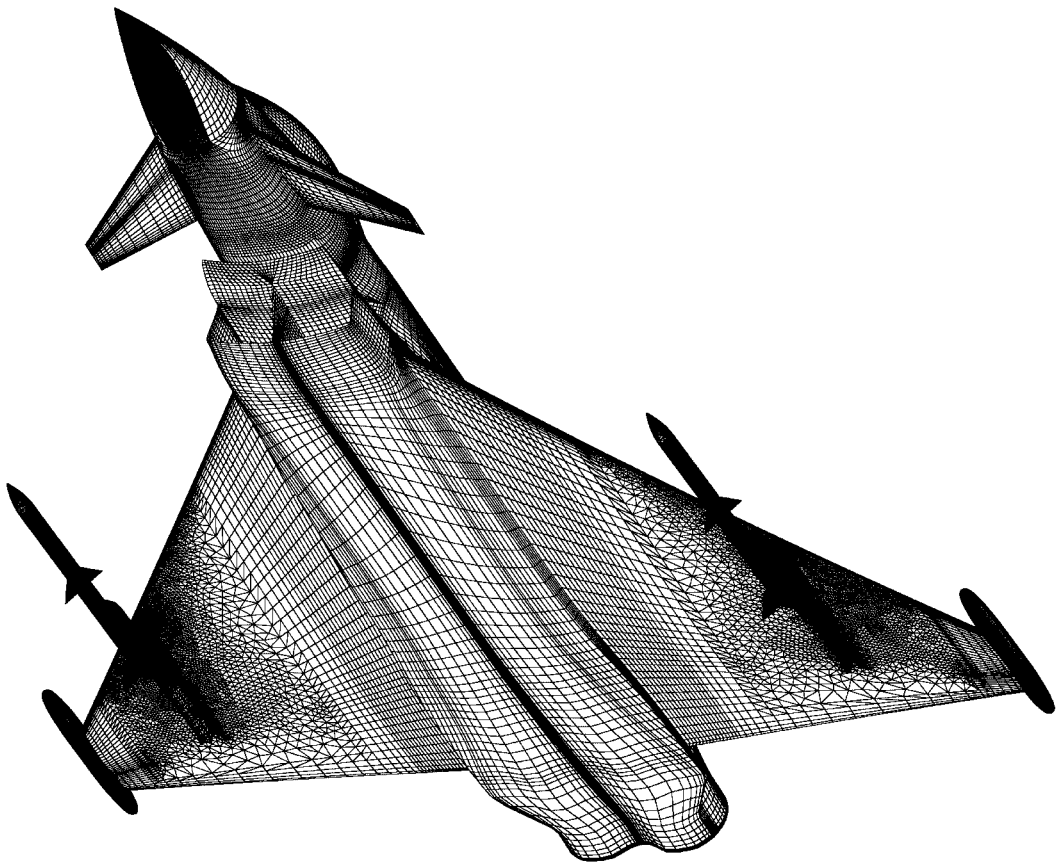

Figure 25. Hybrid grid for rail release of missile.

sands of blocks, each of which must have at least four cells in each co-ordinate direction. The propagation of this grid density through the domain would be highly inefficient.

It was therefore decided that a hybrid grid approach would be more appropriate, with the same block structured grid used for both variants and the unstructured region employed around the aft half of the nacelle.

A block-structured mesh was generated around the fuselage, wing, pylons and a subset of each nacelle; the latter comprised the intake lip, throat and engine face and the forward half of the exterior cowl, which was extrapolated downstream, thereby effectively defining a stream-tube. A region of the block-structured mesh adjacent to the streamtube was removed, and the interface pyramid elements created. Unstructured grids were formed on the remaining surfaces of the nacelles, and on part of the exterior surface of the by-pass cowl and the pylon.

The surface grid views of the differing builds in Figure 26 show that the desired result in terms of gridding has been achieved, in that the grids over the vast majority of the configu-

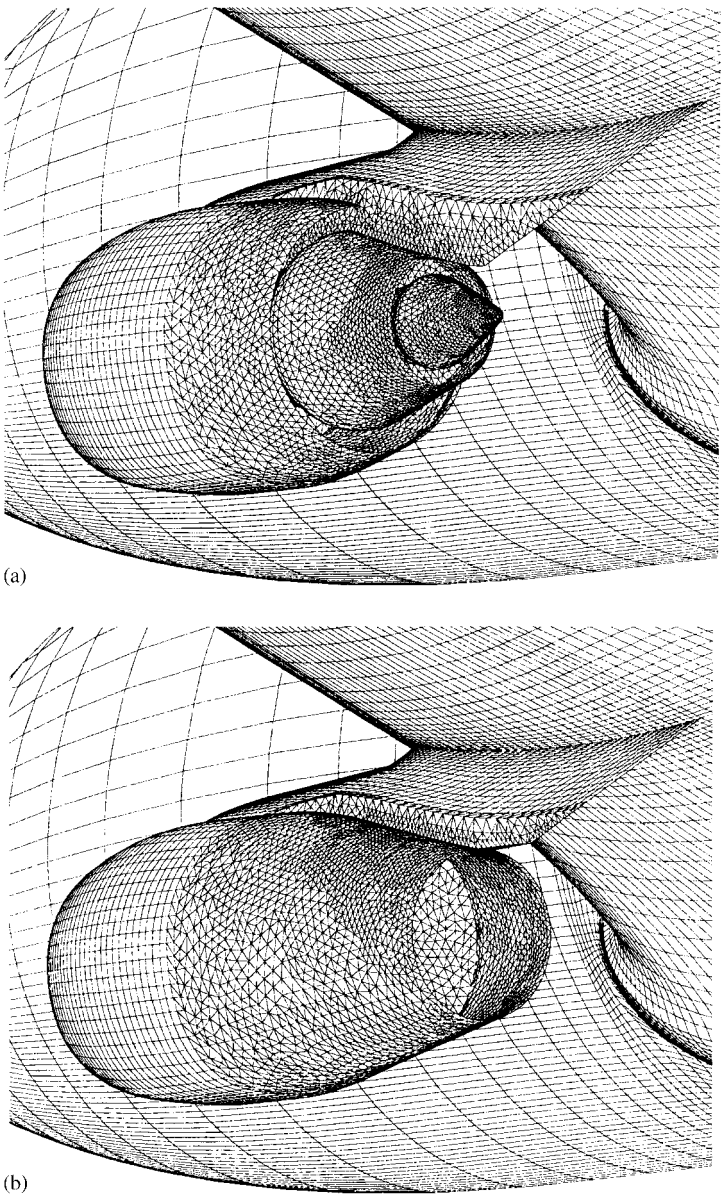

Figure 26. (a) Short and (b) long hybrid grid on cowl configuration.

rations are essentially identical, with the only deviation being due to the cowl variants. Thus, benefits have been achieved in terms of reduced mesh dependency effects, which is critical if, possible, small load increments between the two builds are to be modelled correctly and also in terms of resource required to generate the grids.

CONCLUDING REMARKS

The desire to simulate viscous flows over aircraft places severe technical demands on CFD, particularly in terms of accuracy and efficiency. The largely economic requirements of usability and robustness serve to exacerbate the situation. In the design of a CFD strategy to meet these needs, it is advocated that the technical demands should take precedence, whilst still ensuring the economic requirements are realized.

Traditionally, there is a strong correlation between meeting the requirements of accuracy and efficiency and using strategies based on structured meshes. Similarly, unstructured meshes are associated with usability and robustness. It is concluded that a CFD strategy based on the use of both mesh types, with priority given to structured meshes wherever possible, meets in full the design requirements.

Consideration of the demands of three-dimensional flow modelling promotes the idea that general hybrid meshes should not only be formed of hexahedra and tetrahedra, but also include prisms. These semi-structured regions of grid allow the topology of the mesh to be regular normal to the surface, even when it is irregular on the surface. This feature is believed to be crucial in the accurate prediction of viscous flows. Such general hybrid meshes are best generated in a hierarchy based on the constraint the mesh topology imposes, i.e. structured, semi-structured, unstructured. In generating these individual mesh types, special attention needs to be given to interfacing the elements together.

To maintain the benefits of structure in the grid, special attention needs to be paid to defining the basic data structure that describes the hybrid grid. There are significant benefits to be had by manipulating this data structure to suit the particular architecture of the platform used for the flow computations.

A generalized centred discretization procedure can be developed for solving the Reynolds-averaged Navier–Stokes equations on hybrid meshes. However, this requires that careful attention is paid to artificial dissipation models to ensure that numerical effects do not corrupt the physics of the flow.

The application of hybrid grids to complete aircraft configurations, including moving body problems, shows the very real technical and economic benefits the approach has over others. These are supported by comparisons of data from hybrid grid simulations with other CFD results, experimental data and flight test data.

The hybrid mesh generation system that is described has been developed in a research environment. Its initial requirement has been to prove conclusively that the concept of hybrid grids offers a sound basis for RANS simulations over aircraft. It is believed that this has now been achieved and that the challenge now is to turn the software into an easily used, robust capability.

In terms of ease of use, it is believed that whilst it will never be easier to generate a hybrid grid than it is an unstructured grid, the time required to achieve a high quality, efficient grid will be favourable. To ensure this, it is essential that the environment in which the user interacts with the mesh is as independent of the grid type as possible. A strategy for achieving this has been identified and work in this area is currently in progress.

Robustness is undoubtedly an area of concern in that hybrid grids, by their very nature, require significantly more software to be developed than unstructured grids. The probability of an error occurring, and time taken to trace that error, is inevitably greater. This is offset, however, by the probability that the element quality in a hybrid grid is always likely to be

higher, which will increase the likelihood of the flow code successfully attaining a converged solution.

The hybrid grid philosophy requires a long-term commitment to its development. When many groups have already invested heavily in either solely structured or unstructured approaches, the decision to move to hybrid grids is not taken easily. However, it is believed that the benefits hybrid grids offer the engineer, in both the total elapsed time and cost to achieve the desired end result, justifies this required investment.

ACKNOWLEDGEMENTS

This work has been undertaken with the support of the Procurement Executive, United Kingdom Ministry of Defence.

The work described in this paper represents the effort of a team of people from within the Research Department at ARA. The authors are grateful to their colleagues who have both contributed significantly to the ideas described and provided the figures for the examples discussed. References have been used wherever possible to attribute the source of the work described, but this has not been possible in all cases.

This paper is largely based on a paper given in 1998 at the ICAS Conference in Melbourne, Australia.

The authors wish to dedicate this paper to Dr Clive Albone, who has for many years acted as technical monitor to the work described. Throughout this time, his support and guidance has been greatly appreciated.

REFERENCES

1. Rubbert PE. CFD and changing world of airplane design. AIAA Wright Brothers Lecture, ICAS-94-0.1, Anaheim, CA, U.S.A. 18–23 September, 1994.
2. Weatherill NP, Forsey CR. Grid generation and flow calculations for aircraft geometries. *Journal of Aircraft* 1985; **22**:855–860.
3. Morgan K, Peraire J, Peiro J. Unstructured mesh methods for compressible flows. *AGARD Report 787. Special Course on Unstructured Grid Methods for Advection Dominated Flows*, 5.1–5.39, 1992.
4. Lovell DA, Doherty JJ. Aerodynamic design of aerofoils and wings using a constrained optimisation method. *Proceedings of the 19th ICAS Congress*, Paper-94-2.1.2, 1994.
5. Meakin RL. Computations of the unsteady flow about wing/pylon/finned-store configuration. AIAA-92-4568-CP, 1992.
6. Nakahashi K, Obayashi S. FDM–FEM zonal method for viscous flow computations over multiple bodies. *AIAA Paper* 87-0604, 1987.
7. Weatherill NP. On the combination of structured–unstructured meshes. In: S. Sengupta, J. Hauser, P.R. Eiseman and J.F. Thompson (eds), *Numerical Grid Generation in CFD '88*, pp. 729–739. Pineridge Press: Swansea, U.K., 1988.
8. Shaw JA, Peace AJ. Weatherill NP. A three-dimensional hybrid Structured–unstructured method: Motivation, basic approach and initial results. In: L. Fezoui, J.C.R Hunt and J. Periaux (eds), *Computational Aeronautical Fluid Dynamics*, pp. 157–201. Clarendon Press: Oxford, U.K., 1994.
9. Shaw JA, Peace AJ, Georgala JM, Childs PN. Validation and evaluation of the advanced aeronautical CFD system SAUNA—A method developer's view. *Recent Developments and Applications in Aeronautical CFD*, Paper 3, Royal Aeronautical Society, London, U.K., 1993.
10. Peace AJ, Chappell JA, Shaw JA. Turbulent flow calculations for complex aircraft geometries using prismatic grid regions in the SAUNA CFD code. *Proceedings of the 20th ICAS Congress*, Paper-96-1.4.2, 1996.
11. Weatherill NP, Johnston LJ, Peace AJ, Shaw JA. A method for the solution of the Reynolds averaged Navier–Stokes equations on triangular grids. *Proceedings of the Seventh GAMM Conference on Numerical Methods in Fluid Dynamics*, Louvain-La-Neuve, Belgium, Berlin: Springer-Verlag, 1987.
12. Sykes LA. Development of a two-dimensional Navier–Stokes algorithm for hybrid (structured/unstructured) grids. ARA Report 81, 1990.
13. Shaw JA, Weatherill NP. Automatic topology generation for multi-block grids. *Applied Maths and Computation* 1992; **52**:355–388.

14. Shaw JA, Lovell C, Chappell JA. Topology and surface grid generation for block-structured grids. In: M.A. Cross *et al.* (eds), *Numerical Grid Generation in Computational Field Simulation*. Mississippi State University: Mississippi, 1998.
15. Chappell JA, Shaw JA, Leatham M. The generation of hybrid grids incorporating prismatic regions for viscous flow calculations. In: B.K. Soni, J.F. Thompson, P.R Eiseman and J. Hauser (eds), *Numerical Grid Generation in Computational Field Simulation*, pp. 37–546. Mississippi State University: Mississippi, 1996.
16. Childs PN, Shaw JA. Generation and analysis of hybrid structured/unstructured grids. In: M.J. Baines and K.W. Morton (eds), *Numerical Methods for Fluid Dynamics IV*, pp. 499–507. Clarendon Press: Oxford, U.K., 1992.
17. Leatham M. The Efficient Generation of 3D Delaunay Grids Without the Need for Boundary Modification, loc. cit. in Ref. [13], 1998.
18. Shaw JA. Hybrid grids. In: J.F. Thompson, B.K. Soni and N.P. Weatherill (eds), *The Handbook of Grid Generation*. CRC Press: Boca Raton, Florida, 1998.
19. Weatherill NP. A method for generating irregular computational grids in multiply connected planar domains. *International Journal for Numerical Methods in Fluids* 1988; **8**:181–197.
20. Leatham M, Chappell JA. On the rapid generation of hybrid meshes due to design driven geometry perturbation, loc. cit. in Ref. [13], 1998.
21. Forsey CR, Ierotheou CS, Block U, Leatham M. Parallelisation and performance evaluation of the aeronautical flow code ESAUNA. *Proceedings of HPCN '96*. Berlin: Springer-Verlag, 1996.
22. Leatham M. Developments to the SAUNA system to facilitate hybrid grid use and remeshing within a parallel design environment.
23. Stokes S, Chappell JA, Leatham M. Efficient numerical store trajectory prediction for complex aircraft/store configurations. AIAA 99-3712, 30th AUAA Fluid Dynamics Conference, 28 June–1 July, 1999, Norfolk, VA.
24. Giles MB. Accuracy of node based solutions on irregular meshes. 11th International Conference on Numerical Methods in Fluid Dynamics, Williamsburg, Virginia, June, 1988.
25. Roe PL. Error equation. ICASE Report No. 87-6, January, 1987.
26. Peace AJ, Shaw JA. The modelling of aerodynamic flows by solution of the Euler equations on mixed polyhedral grids. *International Journal of Numerical Methods and Engineering* 1992: **35**:2003–2029.
27. Peace AJ, May NE. A multi-grid Navier–Stokes method for block-structured grids. ARA CRM237/8/4,1996.
28. Wilcox DC. Multi-scale model for turbulent flows. *American Institute of Aeronautics and Astronautics Journal* 1998; **26**:1299–1310.
29. May NE. Efficient implementation of two equation and differential Reynolds stress turbulence models into a cell-vertex, explicit time-marching Navier–Stokes flow code. ARA CR M345/1, 1998.
30. Catherall D. Adaptivity via mesh movement with three dimensional block-structured grids, loc. cit. in Ref. [14], 1996: 57–66.
31. Hunt DL. Development and application of farfield drag extraction techniques for complex viscous flows. *Aeronautical Journal* 2001; **105**:161–169.
32. Hackett KC, Rees PH, Chu JK. Aerodynamic design optimisation applied to civil transports with underwing mounted engines. *Proceedings of the 21st ICAS Congress*, Paper-98-2.3.1, 1998.
33. Robinson G. The incorporation of active control methods into the TGRID store trajectory prediction code. DRA/AS/HWA/TR95040, 1995.

