University of Glasgow
DEPARTMENT OF
AEROSPACE
ENGINEERING

Towards Automatic Multiblock Topology Generation

B. J. Gribben, K. J. Badcock and B. E. Richards

Towards Automatic Multiblock Topology Generation

B. J. Gribben, K. J. Badcock and B. E. Richards

University of Glasgow

Department of Aerospace Engineering

Report Number 9826

August 1998

Abstract

The need for automation of the multiblock grid generation process is discussed. A new approach to automatically process a multiblock topology in order to prepare it for the grid generation process is described. The method is based on a cost function which attempts to model the objectives of the skilled grid generation software user who at present performs the task of block positioning and shaping in an interactive manner. A number of test cases are examined. It is also suggested that an existing unstructured mesh generation method could be adopted as an initial topology generation tool. Further work towards creating a fully automatic grid generation tool and extension into three dimensions are discussed briefly.

Contents

1 The need for automation

1.1 Introduction

Multiblock or zonal structured grids remain a popular choice in CFD. This approach involves an unstructured arrangement of blocks with structured grids which conform with the problem geometry. The alternatives of unstructured, Cartesian, hybrid structured-unstructured and overset (Chimera) grids each have their own advantages and disadvantages. The choice of which one to use is difficult, an essential element of which is a compromise between the relative complexity of grid generation and flow solution. Multiblock grids afford the advantage of easier calculation management and lower operation counts and memory requirements due to their inherent structure, but grid generation for complex configurations is problematic and time-consuming. The subject of which method to choose is not discussed further here, for an introduction to the issue see [1],[2]. Here we are interested in the multiblock grid generation procedure, and note that to address its particular problems is relevant and useful since simulation using multiblock grids is popular. For some recent examples of its application see [3],[4],[5],[6].

1.2 Elements of the analysis process

Figure 1 shows a schematic diagram of the elements of a CFD analysis using multiblock grids. With modern CAD and graphical plotting software, the geometry definition and results analysis stages present few problems. Numerous satisfactory commercial packages exist for these tasks, with present work concentrating on improving speed and extending capability although the basic tools are well established. The flow solution stage is the subject of much ongoing research, but with modern computing power even large scale simulations can be achieved with reasonable turnaround times. The primary obstacle to obtaining accurate flow solutions is the lack of a practical, accurate and general turbulence model.

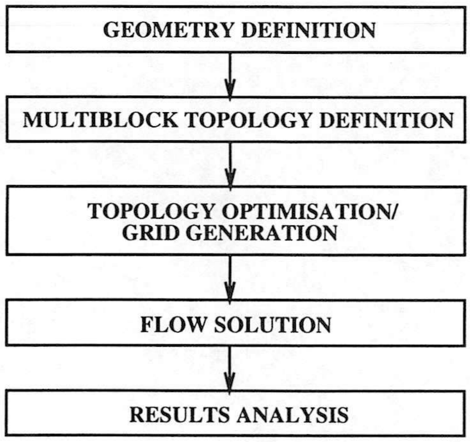

Figure 1: *A CFD analysis process using the multiblock method*

The bottleneck in the process occurs at the second and third stages. Even for fairly simple geometries in two dimensions, the task of designing a suitable arrangement for the grid blocks can be a demanding one. Each part of the problem geometry requires a body conforming local arrangement of the blocks, for example a 'C'-shaped arrangement around an aerofoil, but these local patterns are often difficult to match as a coherent whole. The task of defining an appropriate block pattern is known as 'topology generation'. In three dimensions the task can be daunting and requires considerable skill. Fine adjustments of the curves and block faces making up the topology and actually generating the grid to satisfaction can take man-months of effort for complicated configurations like full aircraft. The need to facilitate the topology and grid generation process by providing interactive graphical environments specifically designed for the task was recognised nearly ten years ago [7],[8]. A few years later Thompson and Weatherill [2] were able to list several commercial packages providing this capability and subsequent development has continued apace. Although these tools undoubtedly accelerate the process, the amount of time and effort required for grid generation still impedes routine analyses for multiple geometries, especially for complex configurations. Progress towards the alternative goal of fully or mostly automatic grid generation for arbitrary geometries [2],[8] has not been as impressive. In Thompson's recent review paper [9] the need for automation is particularly stressed. Real progress has been made by several authors but all of the diverse approaches suggested to date require a degree of skilled user input. The main problem is the difficulty in encapsulating the 'art' of topology generation in a programmable method. The approach of Dannenhoffer[10],[11], which is an integral part of the National Grid Project[12], is probably the most advanced method to date in terms of automating as much of the grid generation process as possible. An abstract "topology plane" is employed initially to interactively design the topology, and block faces are automatically set up by the code. A stochastic process is then employed to reduce the number of blocks. The user then proceeds to edit the topology and construct the grid using a state-of-the-art GUI. Stewart[13],[14] employs a search algorithm with a directional probe to build a two-dimensional block decomposition. This promising approach has proven difficult to apply generally, and it is unclear how well it could extend to three dimensions. The SAUNA[15] system employs a library of known topologies; to generate a new grid with a known topology is therefore straightforward, but for a new topology considerable effort is required to add to the library. The ICEM-CFD system[16] can automatically generate local topologies around recognisable components, after which the user must create the remainder of the topology. Unstructured quadrilateral and hexahedral mesh generation techniques have also been employed to create block topologies[17],[18]. Note that the methods used are not fully automatic and appear to suffer from generality problems. However, this type of approach appears very promising since a number of general, fully automatic methods have been established in the parallel field of structural mechanics. In Section 1.3 it will be suggested that a particular method for initial topology generation possessing the characteristics required already exists and has been well developed. There is therefore the potential to fully automate one of the troublesome elements of Figure 1. Section 2 is the main part of the present study. Having defined an initial multiblock topology, the actual shape and location of the blocks must be defined. A new, straightforward method is proposed for automatically adjusting both the relative placement of blocks and the shape of the curves making up their edges. In this way the subsequent generation of the block interior grids can take place with little or no recourse to further manual block placement or edge shape modification. This process is here called 'topology processing'. Together with established algebraic grid generation and elliptic smoothing techniques this provides the potential for automation of the third stage in Figure 1. After a topology of good quality has been obtained, the task of generating the grid proper in the interior of the blocks becomes straightforward using conventional algebraic grid generation tools. Any remaining grid smoothness problems across block boundaries can be treated using elliptic smoothing.

1.3 Automatic topology generation

A multiblock grid consists of an unstructured arrangement of structured grid blocks. Traditionally the definition of this block arrangement is conceived by the expert user who views the domain in question and imagines the best way to fill it with blocks. This is a skilled task, especially in three dimensions. An attempt to replicate the expert's thought processes in code to produce an automatic tool would necessarily involve shape recognition and trial and error as well as an appreciation of the target flow solver's requirements for the grid. Rather than starting from scratch in an attempt to create such a tool, a simpler alternative is possible. Since the topology consists essentially of unstructured quadrilateral blocks in 2D or unstructured hexahedral blocks in 3D, it is possible that one or more automatic mesh generation procedures developed for structural analysis could be suitable for generating initial multiblock topologies. In this way the expert task of generating multiblock topologies for each individual case can potentially be reduced to the expert task of choosing an existing automatic mesh generation method which produces multiblock topology-like results. Several automatic unstructured quadrilateral and hexahedral mesh generation methods exist, see for example [19], the resulting meshes each having there own attributes. An approach which generates results consistently striking in their similarity to good multiblock topologies is the Medial Axis approach of Armstrong et al., see [20]-[24]. The method is based on a skeletonization technique (where for example a 2D shape is encoded in an esssentially 1D manner) which is well known for its high quality of shape description. Intriguingly, the method was initially proposed as a model for human shape perception, which perhaps explains its ability to generate domain decompositions which fit geometries well, the main requirement of a multiblock topology. This speculation aside, in the Medial Axis approach there is an established automatic domain decomposition technique which results in good quality unstructured quadrilateral or hexahedral meshes which appear to meet the requirements of multiblock decompositions. Of course an initial topology formed in this way would consist of blocks with straight-sided faces. The initial topology may also have other unwanted features such as poor orthogonality at block corners and poorly shaped blocks which do not conform well with other blocks and the problem geometry. The re-shaping of the initial topology for our purpose is the subject of the next section.

2 Automatic topology processing

2.1 Rationale

Once the initial topology has been constructed, it is necessary to form the detailed shape of the curves making up the edges of the blocks and to decide on the placement of important points such as where a number of block corners meet. This process is referred to here as topology processing. There is no generally applicable definition of an optimal multiblock grid or topology. Different grids and different topologies can be used to obtain good results, see for example [25] where various grids and topologies were employed to good effect on the same two-element aerofoil problem. In the absence of a definite objective in optimising the topology, to achieve our goal of obtaining an automatic procedure we instead attempt to model the actions of an experienced grid generation engineer. Topology processing is achieved with modern grid generation packages using an interactive Graphical User Interface (GUI). The GUI enables simultaneous design and assessment of the topology but is very labour intensive. The skill involved is to shape the topology in a manner which will allow the generation of a grid with good characteristics such as orthogonality and smoothness. These qualities are in themselves difficult to define as well as to achieve, which is one reason why grid generation is often referred to as an art as well as a science. An engineer experienced in multiblock grid generation soon recognises certain simple elements to this process however; in this section it is argued that these elements can be formulated in a cost function which can be used to quantify the quality of the topology. To simulate the interactive operations of an engineer the cost function can then be minimised to achieve a topology of good quality. The cost function will be constructed using geometric considerations only. In some cases another factor in grid generation, including the topology design, is the expected behaviour of the flow itself; notably grid lines can be deliberately aligned with streamlines and shock waves. Topology design based purely on geometry will in many cases be sufficient, and at the least will provide an advanced starting point for further modification based on the actual flow.

In Section 1.3 it was discussed how an unstructured mesh generation method can provide an initial topology definition. An ideal initial topology generator would produce topologies which would require no processing, this stage could be by-passed and grid generation could proceed directly. Even if the topology generator produced straight-sided blocks, elliptic smoothing could be sufficient to provide a smooth grid especially if a large number of small blocks were used. However, although it is difficult to quantify how much poor quality in a block topology elliptic smoothing can cure, there does not appear to be at present an automatic, unstructured quadrilateral/hexahedral mesh generation method which can deliver the ideal level of topology quality. Even the most promising method available for this application, the Medial Axis approach discussed in Section 1.3, would require significant additional refinement of block edge shape and singular point location, which is too much to demand of elliptic smoothing in the general case.

2.2 Curve definitions

In the present study each curve or block edge is defined simply as consisting of straight line segments joining p equally spaced points with index j, see Figure 2. To simplify programming, all curves have p points irrespective of the actual curve length or shape. The initial location of the points is obtained by interpolation from the initial topology definition. A straight line segment approach cannot provide a high quality of shape description without using an excessive number of data points. However, since these curves are used here only to define internal block boundaries, onto which a spline can be fitted for algebraic grid generation and across which elliptic smoothing may be employed, such a definition is adequate. Note that although the problem geometry is also represented by straight line segments during the topology processing, the problem geometry is fixed and the original definition can be recalled on proceeding to the grid generation stage.

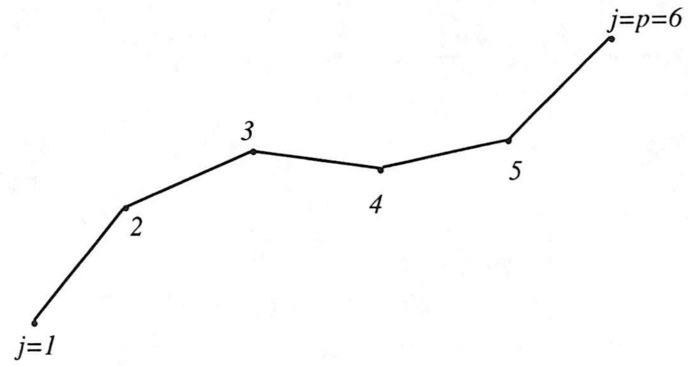

Figure 2: *Curve simply defined as straight line segments*

2.3 Cost function

Figure 3 shows a multiblock grid for the NLR 7301 wing/flap configuration which has been used in a CFD study where excellent agreement with experiment was obtained[26]. Figure 4 shows the topology defined by the block edges, and Figure 5 shows a detail of this around the flap. Reference will be made to these figures to help illustrate the objectives of the cost function construction.

In Figure 5, there are two points in the vicinity of the flap leading edge where five blocks corners meet. Here the designer must consider how best to set the block

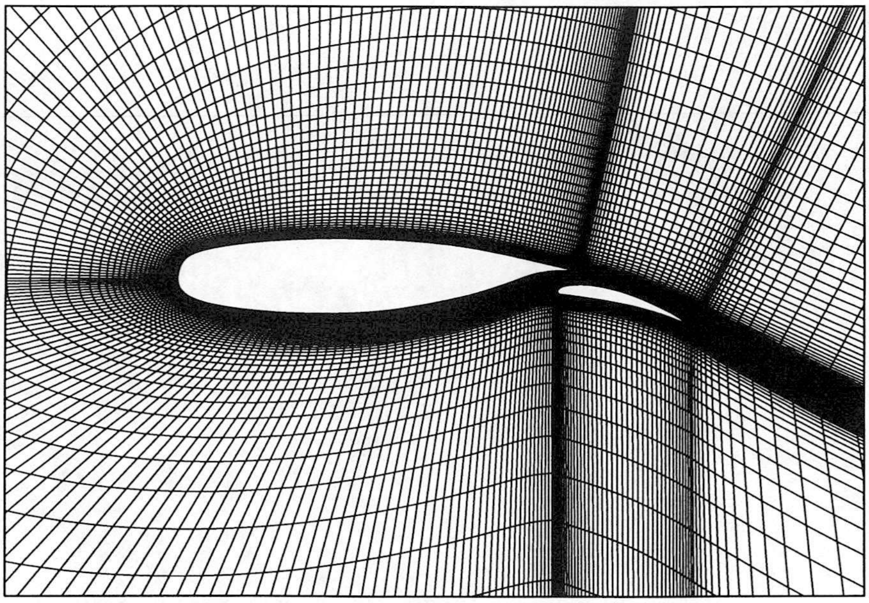

Figure 3: *Multiblock grid for NLR 7301 wing/flap configuration*

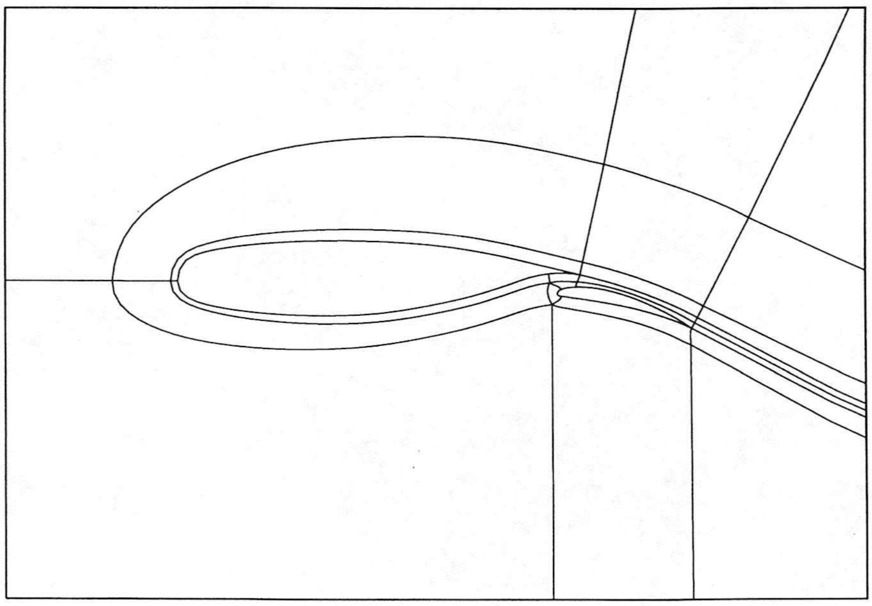

Figure 4: *Block topology for NLR 7301 wing/flap configuration*

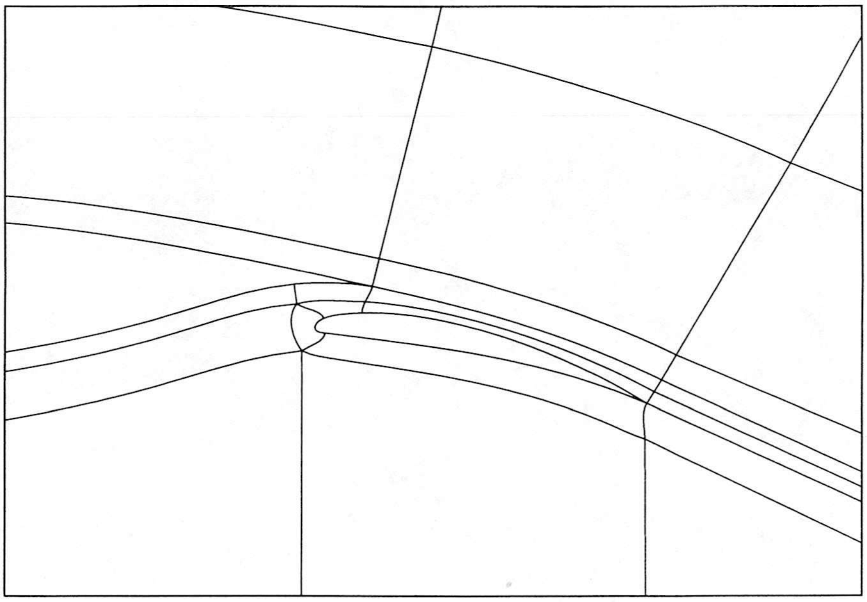

Figure 5: *Detail of block topology around flap*

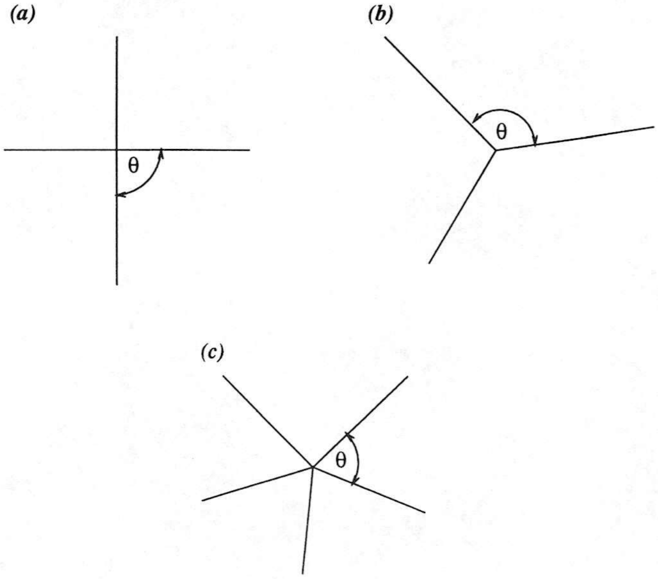

Figure 6: *Several block corners meeting at one point*

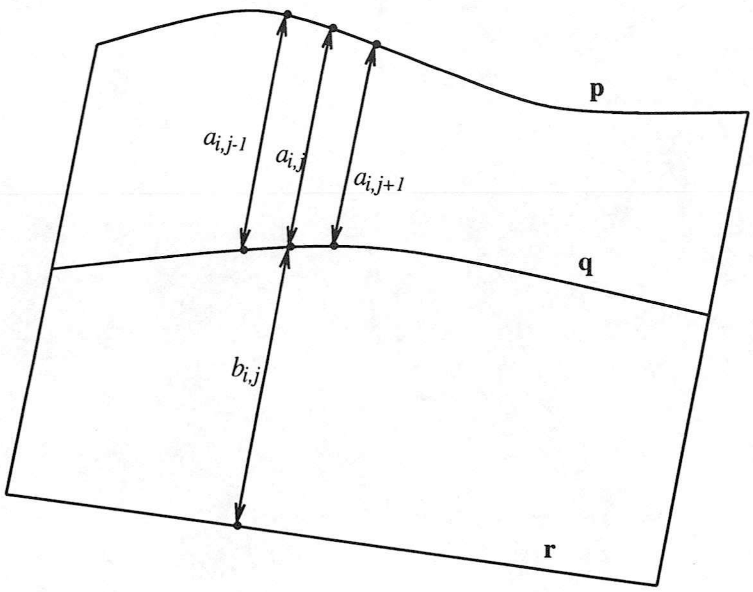

Figure 7: *Measurement of shape-following cost element*

corner angles. Structured grid flow solvers give most accurate results when the grid cells are orthogonal since this minimises the truncation error associated with the discretisation. When four blocks meet at a point, as shown in Figure 6(a), it is therefore desirable to ensure that the angle θ in the corner of each block is as close as possible to a right-angle. Similarly when three, five or more blocks meet at a point, as shown in Figures 6(b) and 6(c), it is desirable to have the same value for θ in each of the block corners so that no one block corner has cells with a large deviation from orthogonality. In our cost function we therefore penalise deviation of the vertex angles θ_i for each block corner or vertex i. A simple way of achieving this is to write the cost C_v associated with block vertices

$$C_v = \sum_{i=1}^{v} \left(\theta_i - \frac{\pi}{2}\right)^{1.1} \tag{1}$$

where v is the total number of vertices. In this way where four blocks meet at a point the cost is zero if all of the block corners form right angles, and the cost increases

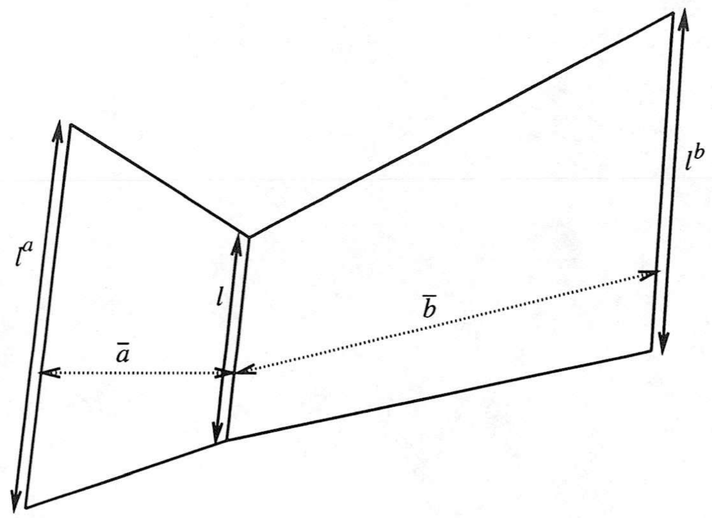

Figure 8: *Measurement of block expansion cost element*

sharply on deviation from this. Since the exponent is greater than one, when a number of blocks other than four meet at a point the minimum cost is incurred when all the block corner angles are equal. A value of 2 was used initially for the exponent, but the cost incurred when other than four blocks would meet at a point rendered other costs insignificant.

Figure 3 shows a grid with good smoothnesss properties. On examining figure 4, it is evident how the shape of the interior block edges follows the shape of the aerofoil surfaces to contribute to this smoothness. Grid smoothness is possible only if gradual changes in the curvature of adjacent grid lines are permitted. Consider Figure 7 where two blocks are shown which share a common edge **q**. To encourage grid smoothness between the block edges **p** and **r** the shape of **q** will ideally represent a transition from the shape of **p** to **r**. The closer **q** is to **p**, the more closely the shape of **q** should follow that of **p**, and the more the influence of **r** should diminish.

A cost element C_s to penalise poor 'shape-following' has been constructed as

$$C_s = \sum_{i=1}^{c}\sum_{j=2}^{p-1}\left\{\frac{\overline{b}_i}{\overline{a}_i+\overline{b}_i}\left(\frac{A_{i,j}}{\overline{a}_i}\right)^2 + \frac{\overline{a}_i}{\overline{a}_i+\overline{b}_i}\left(\frac{B_{i,j}}{\overline{b}_i}\right)^2\right\} \tag{2}$$

$$A_{i,j} = a_{i,j+1} - 2a_{i,j} + a_{i,j-1}$$

$$B_{i,j} = b_{i,j+1} - 2b_{i,j} + b_{i,j-1}$$

where c is the total number of curves. On each curve i there are p equally spaced points. The distance from the point j on the curve i to the corresponding point on an opposing block face is labelled $a_{i,j}$ as indicated in Figure 7. The quantity $A_{i,j}$ is therefore a measure of how well the local curvature of $\mathbf{p}$ is following that of $\mathbf{q}$. This is summed over the length of the curve. Each curve in the interior of the domain (i.e. each curve that does not define a fixed geometry) has two opposing counterparts; $a_{i,j}$ and $b_{i,j}$ are the distance measures to each. To ensure greater influence of curves in close proximity, the influence on curve i of each opposing curve is scaled by their average separations $\overline{a}_i$ and $\overline{b}_i$ from i, defined as

$$\overline{a}_i = \frac{1}{p}\sum_{j=1}^{p} a_{i,j} \tag{3}$$

$$\overline{b}_i = \frac{1}{p}\sum_{j=1}^{p} b_{i,j} \tag{4}$$

The construction of a cost function element to model shape-following is not straightforward. The engineer with experience of multiblock grid generation can readily recognise when blocks are well shaped, but how to define what this means in terms of gradients, curvatures etc. is not obvious. The cost element (2) tries to match local curvature. The definition of the gradients at the curve ends is then important to the success of the method. A previous attempt at constructing a cost element was based on local gradient rather than curvature. Referring to Figure 7, this worked very well for cases where the curves **p** and **r** have approximately the same orientation as **q**, but becomes a poor measure of shape-following otherwise.

In Figure 4 the blocks are fairly regularly shaped in that none of the blocks expand in size very sharply. If a block expands too sharply, then cell orthogonality and grid smoothness can be adversely affected in the block interior. Figure 8 shows two blocks sharing a common edge of length l. The opposing edges have lengths l^a and l^b. A cost element C_e to penalise sharp block expansions has been constructed as

$$C_e = \sum_{i=1}^{c} \left[\frac{(l_i - l_i^a)^2}{l_i \bar{a}_i} + \frac{(l_i - l_i^b)^2}{l_i \bar{b}_i} \right] \tag{5}$$

The total cost C_t associated with the quality of the topology can then be written as

$$C_t = k_v C_v + k_s C_s + k_e C_e \tag{6}$$

where k_v, k_s, k_e are positive constants which define the relative importance of the cost elements. Appropriate values for these constants were found by experimentation using simple model cases and verification on more complex cases, see Section 2.4.

2.4 Cost function minimisation

Equation (6) defines a cost function which measures the quality of a multiblock topology. This cost function is minimised in order to obtain a topology of good quality. The resulting topology is referred to as the 'processed' topology. To do this, a straightforward iterative improvement technique is employed. The number of points p defining each curve is chosen as the minimum number which give a suitable definition of the problem geometries, typically between 8 and 40. A point on one of the c curves is chosen at random. Two random numbers between -1 and 1 are multiplied by the pre-defined maximum displacement distance d_{max}, and the selected point is displaced in the x and y directions by each result respectively, remembering that a point may belong to more than one curve. Points on curves defining the domain boundaries, i.e. on "exterior" edges, are not permitted to move. If the total cost of the modified topology has been reduced then the move is accepted. Otherwise the move is rejected and the former position of the displaced point is recalled. A large number Nc of trial moves are attempted, N being some large integer.

In the cost elements (2) and (5) the quantities $\overline{a}_i$ and $\overline{b}_i$ are used as scaling factors. In implementing the cost function minimisation procedure, care must be taken to ensure that the block shapes are not being inadvertantly altered to maximise these quantities in order to minimise the cost (they are on the denominator). To achieve this they are evaluated infrequently, every 100 successful trial moves.

It is well known that simple iterative improvement does not provide a mechanism for avoiding local minima. Careful selection of the trial moves can help avoid this problem. Trial moves of curve sections as well as single points were employed. Although this helps to avoid local minima to some extent, this simple approach to cost function minimisation could be improved upon, as will be discussed in Section 3.1. It is considered sufficient however for the task of demonstrating the general method. As will be demonstrated below for a number of test cases, iterative improvement has succeeded in finding a good enough local minimum where the block topology properties have clearly been improved in terms of preparation for the grid generation stage.

2.5 Calibration test cases

A simple test case was constructed, consisting of two blocks sharing a common edge, in order to find appropriate relative magnitudes of k_v and k_s in equation (6). For these tests k_e was set to zero. Figure 9 shows some representative results for a

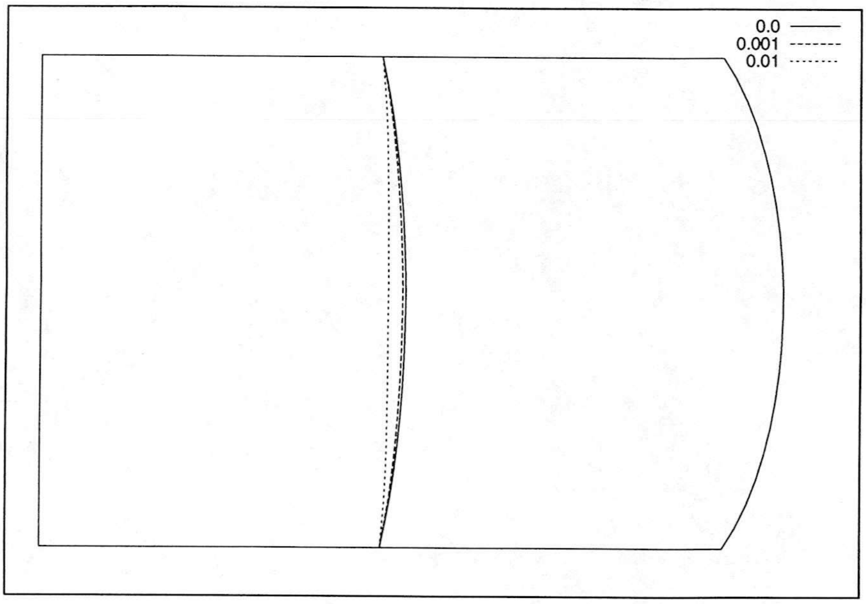

Figure 9: *Test case to find value for* k_v

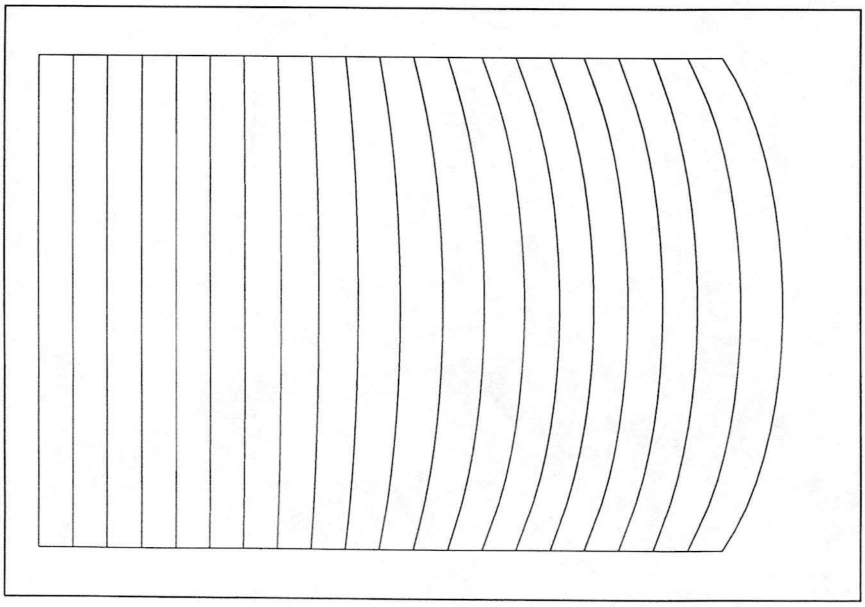

Figure 10: *Test case to check shape-following*

number of cases where $k_s = 1.0$ and the magnitude of k_v was varied. The curve definition $p = 10$ was used. With $k_v = 0.0$ the shape-following cost is the only non-zero part of the cost function. As expected the shape of the resulting curve lies somewhere between the straight line of the left-hand opposing curve and the greater curvature of the right-hand opposing curve. As k_v is increased the tendency for the ends of the curve to form right angles at the block corners increases, eventually to the detriment of the overall shape. A good compromise is found at values around $k_v = 0.001$ (with $k_s = 1.0$), at which condition the effect of the cost associated with corner angles becomes noticeable. For this case the final cost becomes converged to three significant figures for $N = 15000$. A set of similar tests was carried out to ensure that the proximity of opposing block edges has the desired influence on the shape-following cost. Keeping $k_v = 0.001$ and $k_s = 1.0$, the location of the common edge was varied; the results are shown in Figure 10. Note that the original result with a central common edge is shown with the other results superimposed. There is a smooth transition in curve shape as required.

An eight block grid for a single element aerofoil was used to determine a suitable value for the block expansion cost coefficient k_e. Figure 11 shows the initial topology, taken from a grid known to be of good quality which has been used successfully in a CFD study[27]. The figure also shows a processed topology obtained by setting $k_s = 1.0$, $k_v = 0.001$ and $k_e = 0.0$. For this case the curve definition $p = 20$ was used and the final cost becomes converged to three significant figures for $N = 20000$. The result obtained using $k_e = 0.0$ is satisfactory in this case since the initial topology used does not contain blocks with an unacceptable block expansion rate. However, following the same approach as for k_v, gradually increasing the value of k_e should indicate a value where the block expansion cost element begins to have an effect but is not yet dominating the other cost elements. Figure 12 shows the effect of varying

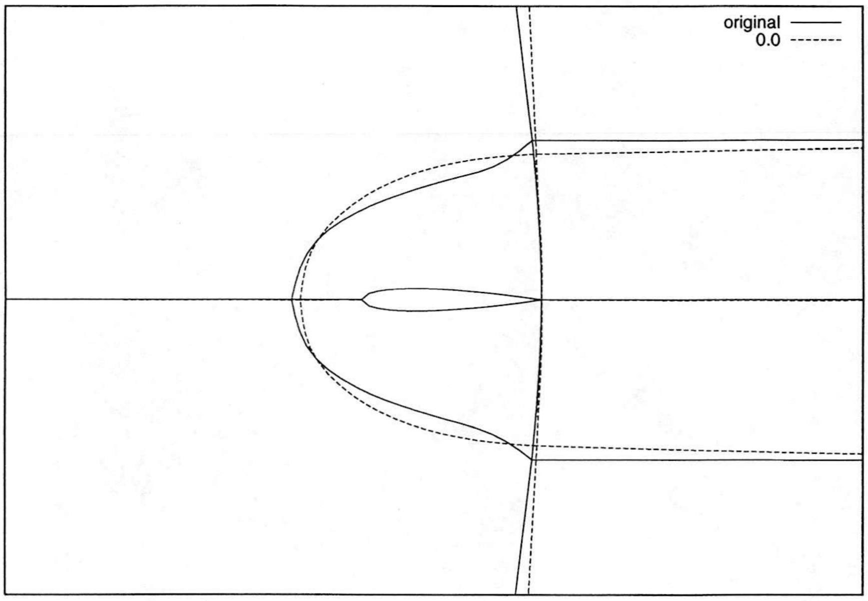

Figure 11: *Block expansion test case, initial topology and processed topology with* $k_e = 0.0$

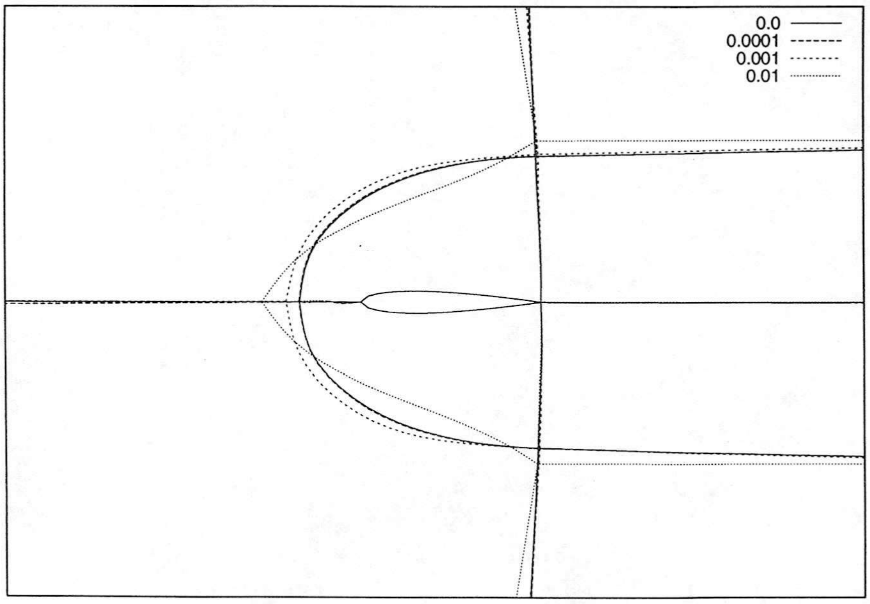

Figure 12: *Processed topologies with various values of* k_e

the value of k_e. The block expansion cost element begins to take effect for values of k_e around 0.001; in the figure for this value the block edge emanating from the aerofoil leading edge has been stretched slightly to match the length of the block edges emanating from the trailing edge. For lower values of k_e there is no effect, and for higher values the block expansion cost begins to swamp the other cost elements, as shown in the figure where the processed topology for $k_e = 0.01$ has poor shape-following and block corner angle charactersistics.

These two examples have indicated appropriate values for the coefficients in equation (6) and demonstrated that the method works well for simple cases. Encouraged by this, the method will now be applied to other existing multiblock topologies from real problems, in order to examine how the method performs on topologies which are known to be already of good quality and to check that no deleterious effects are experienced, before moving on to more realistic test cases. The same coefficients will be used throughout as were used for the example test cases ($k_s = 1.0$, $k_v = 0.001$, $k_e = 0.001$) in the hope that their values will be case independent.

2.6 Existing topologies

A simple and common multiblock topology is a three block grid around a single element aerofoil. The same grid as used above for the calibration case was also used in three-block form. Figures 13 and 14 show the original and processed block outlines for this case. The curve definition $p = 40$ was used and the final cost becomes converged to three significant figures for $N = 10000$. The topology processing method has improved the block corner angles at the trailing edge and maintained a satisfactory shape for the interior block edges.

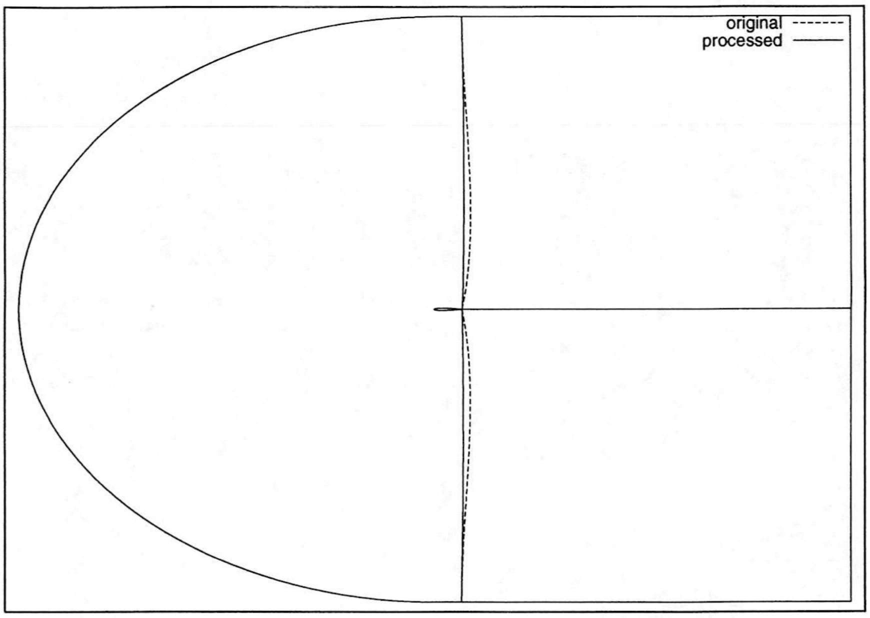

Figure 13: *Three block single element aerofoil grid, entire domain*

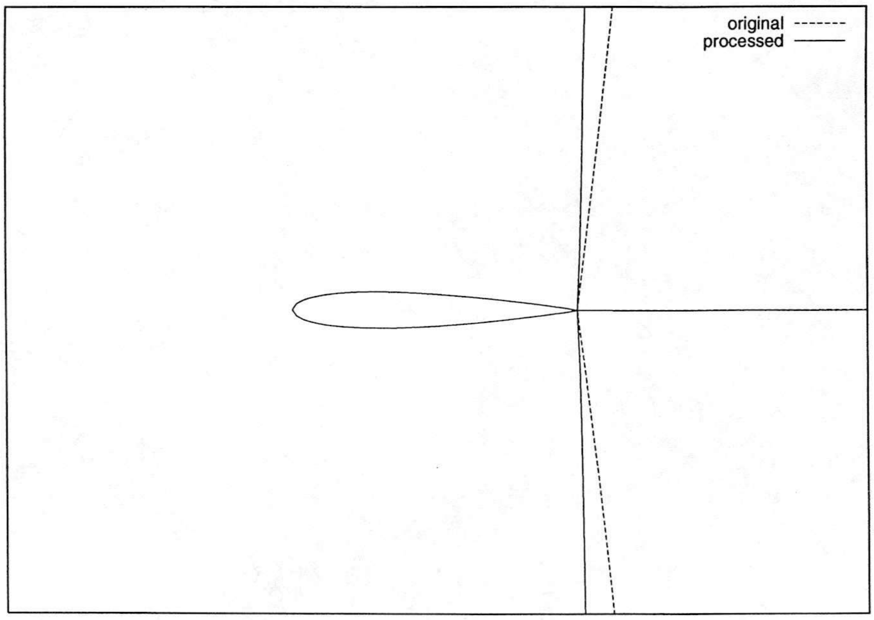

Figure 14: *Three block single element aerofoil grid, detail*

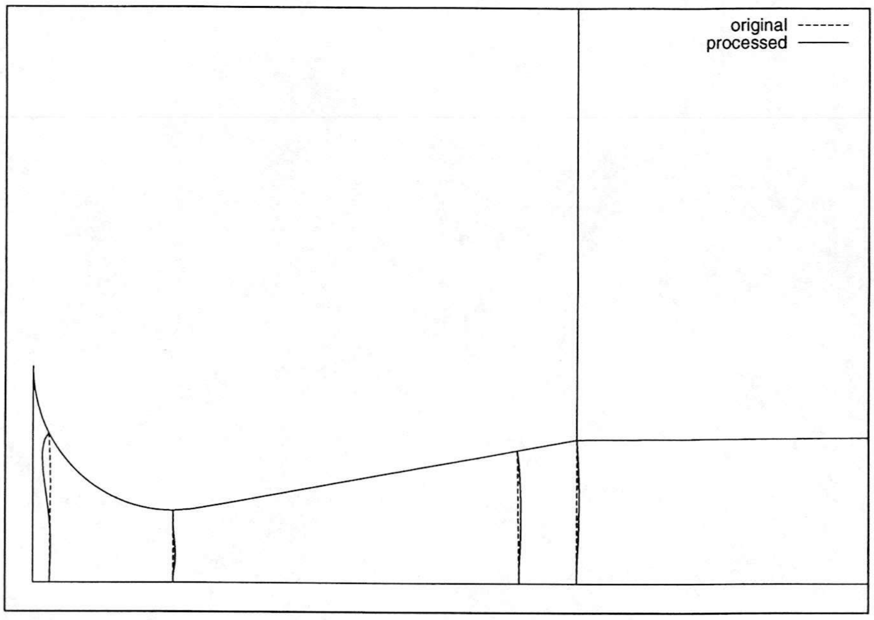

Figure 15: *Nozzle/plume grid*

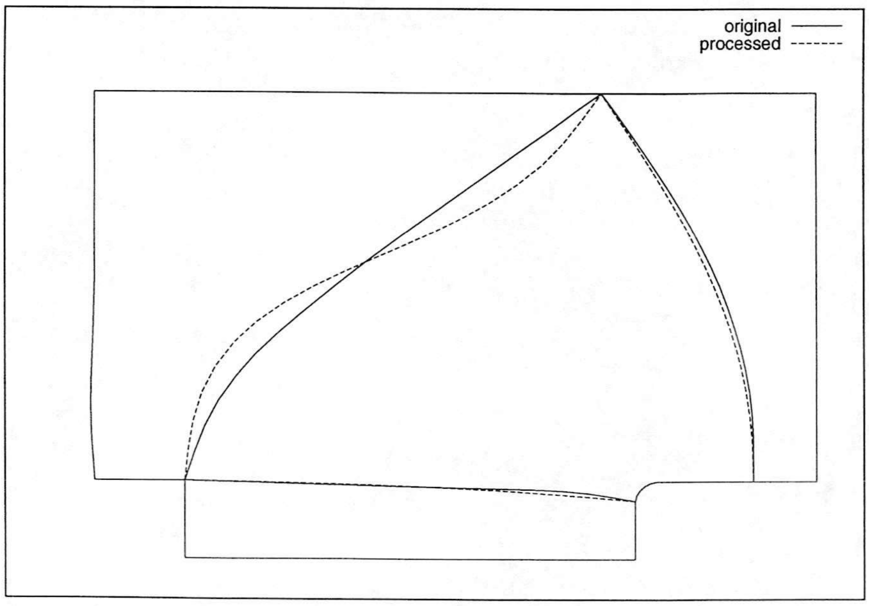

Figure 16: *Cavity flow topology*

Figures 15 shows the original and processed topology for a grid used in a nozzle/plume study[29]. For this case the curve definition $p = 10$ was used and the final cost becomes converged to three significant figures for $N = 15000$. Again the topology processing method has improved the block corner angles, quite significantly changing the shape of one curve, but a satisfactory trade-off between orthogonality and curve smoothness/shape-following has been achieved.

Figures 16 shows the original and processed topology for a grid used in a cavity flow study. The cavity has a right-angled leading edge and a radiused trailing edge, the novel topology created for this configuration is a good example of how some imagination can be required to create a topology suitable for even simple configurations. For this case the curve definition $p = 10$ was used and the final cost becomes converged to three significant figures for $N = 20000$. The topology processing method has again significantly altered the shape of one of the curves in order to improve block corner angles.

Figure 17 shows the multiblock topology for a multi-element aerofoil grid from British Aerospace which has been used in a CFD study of a high-lift configuration where good agreement with experiment was achieved[28]. The large number of blocks required for even moderately complex configurations (81 in total for this grid) is evident from the figure. The result of the topology processing procedure is shown in Figure 18. For this case the curve definition $p = 30$ was required and the final cost becomes converged to three significant figures for $N = 20000$. There is very little room for improvement from the initial excellent configuration, the only real difference is an improvement of the block corner angles, most notably at the point where five blocks meet below the forward part of the main element.

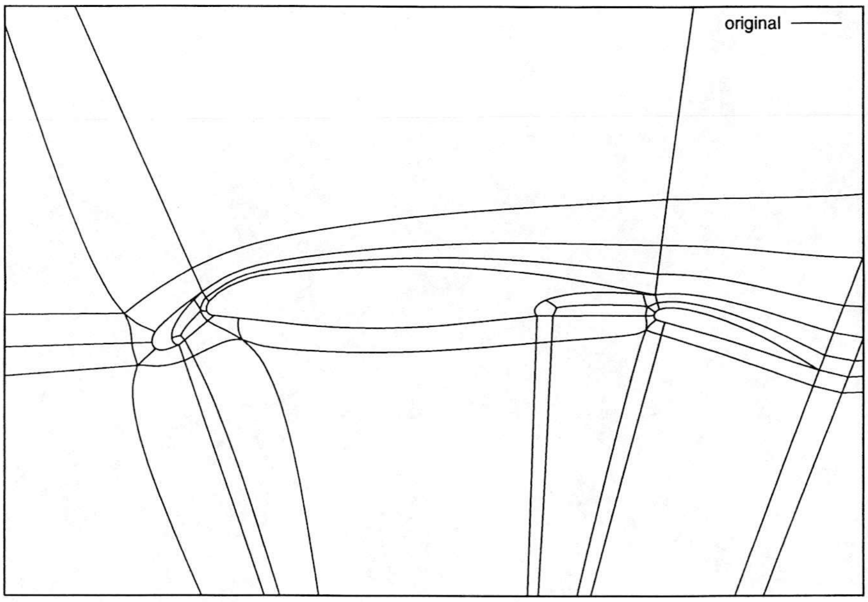

Figure 17: *Original multi-element aerofoil topology*

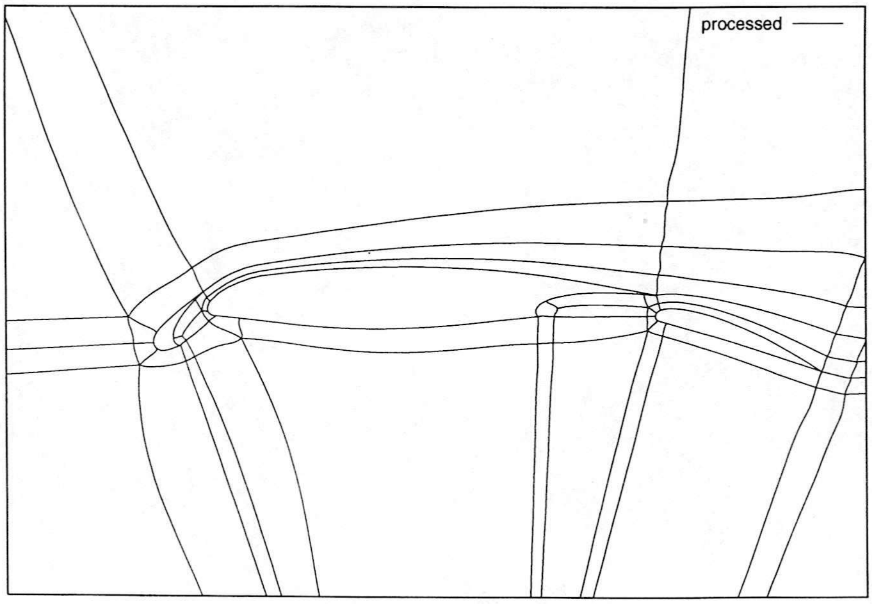

Figure 18: *Processed multi-element aerofoil topology*

2.7 Marine application example

The topology processing method has been applied successfully to simple test cases in Section 2.5 and to real problems where the topology is already of good quality in Section 2.6. The main aim of this work is to produce a topology processing method applicable to the inevitably unrefined initial topologies which can be generated using unstructured quadrilateral grid generation techniques, see section 1.3. A coarse, straight-sided topology has been created manually for a model marine application. This is a demonstration case in order to simulate the result of such an automatic topology generation method, see Figure 19. The corresponding processed topology is shown in Figure 20. For this case the curve definition $p = 10$ was used and the final cost becomes converged to three significant figures for $N = 40000$. The initial configuration has been improved considerably; the blocks have good ortogonality characteristics, do not expand rapidly and conform well with the geometry and each other.

2.8 Two-element aerofoil example

To investigate a further example using initially poor topologies, two coarse straight-sided topologies have been created manually for a two-element aerofoil demonstration case in order to simulate the result of such an automatic topology generation method, see Figures 21 and 22. The Williams B aerofoils[30] are used in both cases. The corresponding processed topologies are shown in Figures 23 and 24. The initial topology A (Figure 21) has the agreeable feature of well located block corners. To modify this topology to obtain a form suitable as a basis for the actual grid generation phase involves changing the shape of the block edges to a smoother, more geometry conforming pattern. This has been achieved by the present topology processing method, see Figure 23. The initial topology B has the additional problem

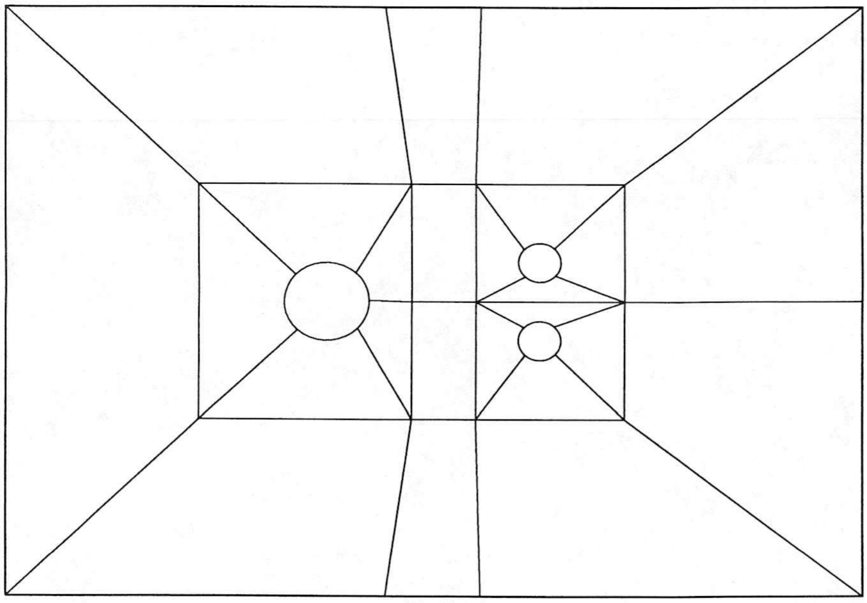

Figure 19: *Initial topology, marine application example*

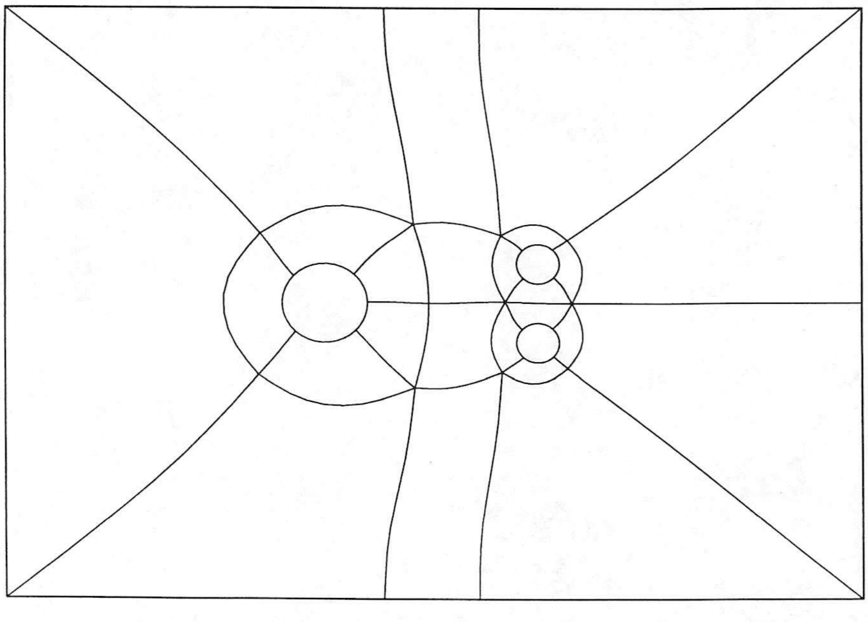

Figure 20: *Processed topology, marine application example*

of an irregularly shaped block at the leading edge of the 'flap'. The topology processing method has also coped with this well, see Figure 24, by drastically reducing the lengths of the long sides of the block at the nose of the flap. Both results from the topology processing method could be used as inputs to the grid generation proper stage. The method has been successful in finding a compromise between smoothing the initial configuration, maintaining reasonable orthogonality and resizing blocks which expand too sharply. It is noted however that the final configurations are different, so the minimisation method has clearly not found a global minimum. This issue will be discussed in Section 3.1.

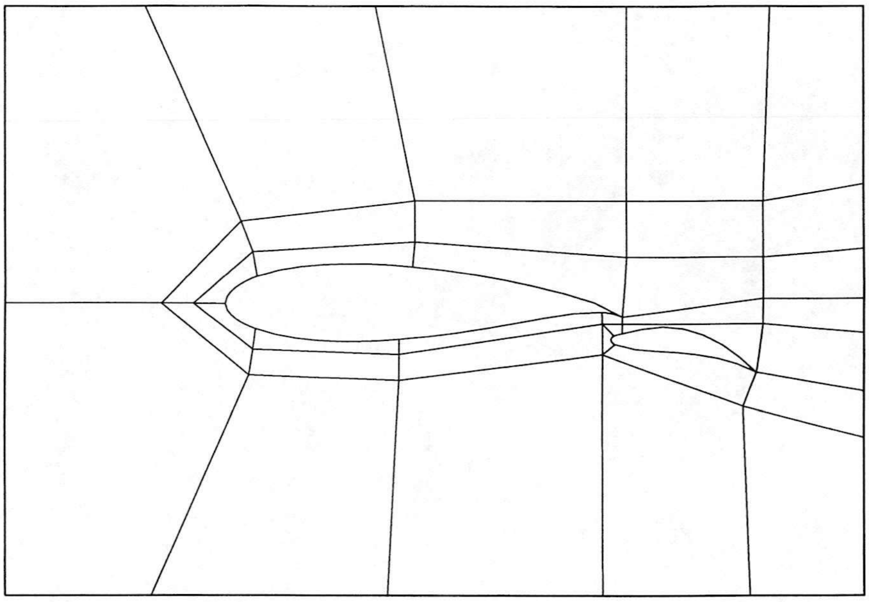

Figure 21: *Initial topology A, two-element aerofoil*

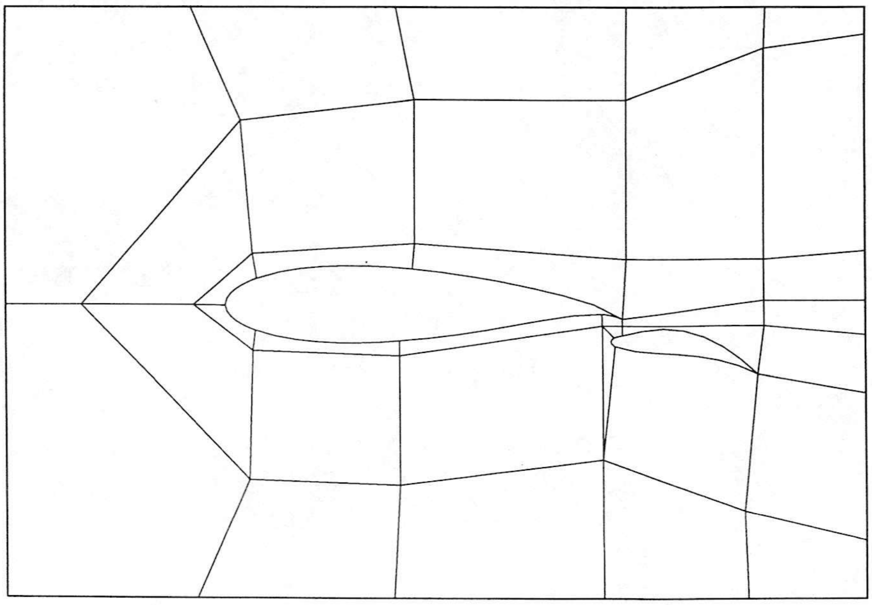

Figure 22: *Initial topology B, two-element aerofoil*

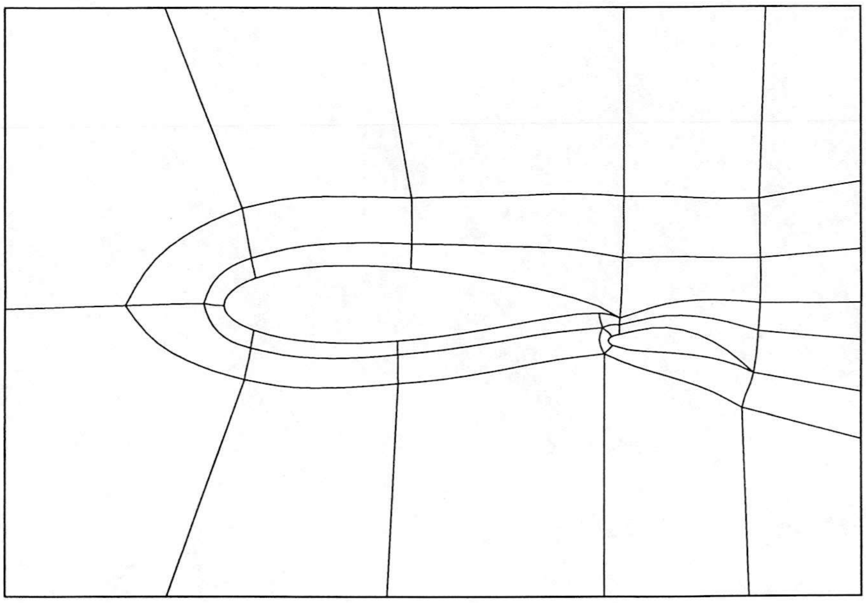

Figure 23: *Processed topology A, two-element aerofoil*

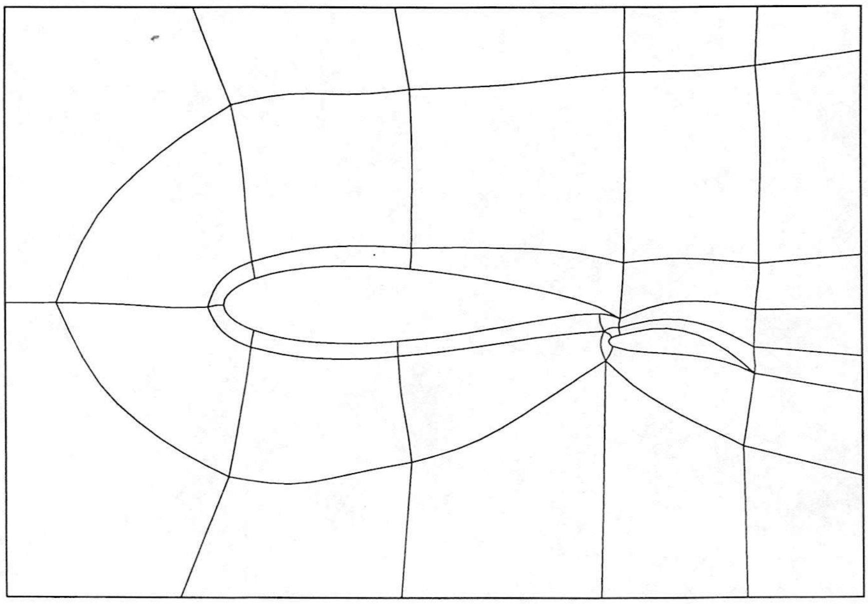

Figure 24: *Processed topology B, two-element aerofoil*

3 Problems encountered and future work

3.1 Global Minimum

The present method does not find a global minimum for the multi-element aerofoil case. This is not surprising given the very simple minimisation procedure employed. The local minima obtained for the cases examined here are satisfactory, but there is doubt whether this will be generally true. A straightforward extension of the iterative improvement technique is simulated annealing[31] which is well known to obtain near-optimal results for a broad range of minimisation problems. A drawback to this method is that it necessitates additional computational effort; the present method already requires a substantial amount of computing time, the two-element aerofoil example requiring approximately half an hour using a desktop PC. Ultimately the most promising direction is likely to be to begin with a higher fidelity curve description (for example using B-splines) to reduce the number of data points and hence operations, and using a more advanced minimisation procedure, perhaps again borrowing from structural mechanics where there are established techniques for shape optimisation in component design.

3.2 Curve definition refinement

As noted directly above, a reduction of the computational time required for the process is desirable. If a small number of points is used in the curve definition, then the number of operations necessary to evaluate the cost function (and hence compuatational time) is reduced. However, often a finer definition of the curves is required to adequately represent the problem geometry. With this in mind, a curve refinement approach was adopted. The initial curve definition could be fairly coarse, and after a number of trial moves the curve definition would be successively refined. This approach did reduce the overall time required to obtain a converged solution in some cases, but was not successful generally. In Figure 11, the curves are defined using 20 points connected by straight line segments. These segments are small enough to represent the strong curvature at the leading edge. When the refinement approach was attempted for this case, a coarser curve definition misrepresents the leading edge curvature, introducing unwanted features into the curves attempting to follow the aerofoil surface. These features must then be removed by applying a large number of trial moves in the later stages when the curve definition becomes adequate.

3.3 Generality

The same cost function and cost function coefficients were used to process all the test cases presented. This provides some evidence that the method may be generally applicable, but realistically a far larger number of test cases from a greater range of problems should be examined before being able to state this confidently. A particular question is how the method will perform using large numbers of small blocks; all of the test cases considered had a relatively small number of large blocks.

3.4 Automatically generated topologies

This report has concentrated on demonstrating the potential of an automatic topology processing method, which has been examined essentially in isolation from the other elements of the multiblock grid generation process. The next step should be to examine whether the method can fulfil its potential by linking with the other elements, see section 1.2. There is little doubt that algebraic grid generation and elliptic smoothing performs well when based on a sound topology. The main question is how well the processing method would perform given automatically generated initial topologies. The next step should therefore be to use an unstructured quadrilateral mesh generation method (such as the Medial Axis approach discussed in section 1.3) to generate inital topologies for the processing stage to verify that this approach can indeed produce multiblock topologies possessing the required characteristics.

3.5 Extension to 3-D

In three dimensions the problem of multiblock grid generation is more demanding and the need for automation is even greater. The present topoloogy processing method generalises to three dimensions, as do the other elements in the suggested automated route. This work has concentrated on multiblock grid generation in two dimensions, but there is a clear path to the generalisation.

4 Conclusion

A new approach for automatic multiblock topology processing has been presented. A cost function which evaluates the quality of a multiblock topology has been created. The elements of the cost function are based on the objectives of the multiblock grid generation software user when interactively constructing the topology. A simple minimisation procedure is employed to obtain a topology of good quality. The potential of the method has been demonstrated using a number of test problems. It has been suggested that full automation of the entire multiblock grid generation procedure is possible using in sequence an existing unstructured grid technique to obtain an initial topology, the present processing method, then conventional algebraic grid generation and elliptic smoothing. Problems encountered during the study and future work have been discussed.

Acknowledgements

Thanks to the other members of the CFD group within the Aerospace Engineering Department for their contributions and fruitful discussions.

This work is supported by a University of Glasgow scholarship and sponsorship from DERA Bedford.

References

[1] J. D. Anderson Jr. *Computational Fluid Dynamics.* McGraw-Hill, U.S.A., 1995.

[2] J.F. Thompson and N.P. Weatherill. Aspects of Numerical Grid Generation: Current Science and Art. In *AIAA Applied Aerodynamics Conference, Monterey, California*, August 9-11, 1993.

[3] F.S. Lien, W.L. Chen and M.A. Leschziner. Multiblock implementation of a non-orthogonal, collocated finite volume algorithm for complex turbulent flows. *International Journal for Numerical Methods in Fluids*, 23(6):567–588, 1996.

[4] D. Bohn, H. Schoenenborn and H. Wilhelmi. Numerical simultion of supersonic, chemically reacting flow using an implicit finite volume method. *Journal of Thermophysics and Heat Transfer*, 11(1):45–51, 1997.

[5] F. Grasso and G. Iaccarino. Influence of crossflow and turbulence on vortex flow around a supersonic missile. *Journal of Spacecraft and Rockets*, 35(1):37–45, 1998.

[6] J.A. Ekaterinaris. Numerical simulation of incompressible two-blade rotor flowfields. *Journal of Propulsion and Power*, 14(3):367–373, 1998.

[7] J. Hauser and A. Vinckier. Recent Developments in Grid Generation. In *AGARD-CP-464, Applications of Mesh Generation to Complex 3-D Configurations*, 1989.

[8] J.F. Thompson. General Structured Grid Generation Systems. In *AGARD-CP-464, Applications of Mesh Generation to Complex 3-D Configurations*, 1989.

[9] J.F. Thompson. A Reflection on Grid Generation in the 90s: Trends, Needs, and Influences. In *5th International Conference on Numerical Grid Generation*, Mississippi State University, 1996.

[10] J.F. Dannenhoffer. Automatic Blocking for Complex Three-Dimensional Configurations. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[11] J.F. Dannenhoffer. A Technique for Optimizing Grid Blocks. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[12] A. Gaither, K. Gaither, B. Jean, J. Remotigue, J. Whitmire, B.K. Soni, J.F. Thompson, J.F. Dannenhoffer and N.P. Weatherill. The National Grid Project: A System Overview. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[13] M.E.M. Stewart. Domain-Decomposition Algorithm Applied to Multielement Airfoil Grids. *AIAA Journal*, 30(6):1457, 1992.

[14] M.E.M. Stewart. A Multiblock Grid Generation Technique Applied to Jet Engine Configuration. In *NASA-CP-3143, Proceedings of the Software Systems for Surface Modelling, and Grid Generation Workshop*, 1992.

[15] J.A. Shaw and N.P. Weatherill. Automatic Topology Generation for Multiblock Grids. *Applied Mathematics and Computation*, 52:355, 1992.

[16] A. Wulf and V. Akdag. Tuned Grid Generation with ICEM CFD. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[17] T. Schonfeld, P. Weinerfeld and C.B. Jenssen. Algorithms for the Automatic Generation of 2D Structured Multi-Block Grids. In *NASA-CP-3291, Work-*

shop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics, May 1995.

[18] B. Kim and S.D. Eberhardt. Automatic Multi-Block Grid Generation for High-Lift Configuration Wings. In *NASA-CP-3291, Workshop Proceedings: Surface Modelling, Grid Generation and Related Issues in Computational Fluid Dynamics*, May 1995.

[19] Proceedings, 5th International Meshing Roundtable. *Sandia National Laboratories.* 1996.

[20] T.K.H. Tam and C.G. Armstrong. 2D Finite element mesh generation by medial axis subdivision. *Advances in Engineering Software*, 13(5/6):313–324, 1991.

[21] T.K.H. Tam and C.G. Armstrong. Finite element mesh control by integer programming. *International Journal for Numerical Methods in Engineering*, 36:2581–2605, 1993.

[22] C.G. Armstrong, C.L. Toh, D.J. Robinson and T.S. Li. Adaptive Hexahedral Meshing using Solid Primitives. In *Proceedings of the Fourth NAFEMS International Conference on Quality Assurance and Standards in Finite Element and Associated Technologies*, 1993.

[23] M.A. Price and C.G. Armstrong. Hexahedral mesh generation by medial surface subdivision: Part I. Solids wih convex edges. *International Journal for Numerical Methods in Engineering*, 38:3335–3359, 1995.

[24] C.G. Armstrong, R.J. Donaghy and S.J. Bridgett. Derivation of Appropriate Idealisations in Finite Element Modelling. In *Proceedings of the Third International Conference on Computational Structures Technology*, 1996.

[25] W. Haase (editor). *EUROVAL, A European Initiative on validation of CFD Codes.* Notes on Numerical Fluid Mechanics, Vol 42, Braunschweig: Vieweg, Germany, 1993.

[26] A. Parodi. Final Year Project: Two Element Aerofoil. *University of Glasgow, Department of Aerospace Engineering*, 1998.

[27] L. Dubuc, F. Cantariti, M. Woodgate, B. Gribben, K.J. Badcock and B.E. Richards. Solution of the Unsteady Euler Equations Using an Implicit Dual-Time Method. *AIAA Journal*, 36:1417–1424, 1998.

[28] G. Kalitzin, A.R.B. Gould and J.J. Benton. Application of Two-Equation Turbulence Models in Aircraft Design. *AIAA Paper*, 96-0327, 1996.

[29] B.J. Gribben, K.J. Badcock and B.E. Richards. Shock Reflection Hysteresis in an Underexpanded Jet: a CFD Study. *University of Glasgow, Aero Report 9808*, 1998.

[30] B.R. Williams. An Exact Test Case for the Plane Potential Flow About Two Adjacent Lifting Aerofoils. *ARC R&M 3717*, 1973.

[31] R.H.J.M. Otten and L.P.P.P van Ginneken. *The Annealing Algorithm.* Kluwer Academic Publishers, U.S.A., 1989.