INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2000; **32**: 285–311

A grid deformation technique for unsteady flow computations

L. Dubuc, F. Cantariti, M. Woodgate, B. Gribben, K. J. Badcock and B. E. Richards*

Department of Aerospace Engineering, University of Glasgow, Glasgow, G12 8QQ, UK

SUMMARY

A grid deformation technique is presented here based on a transfinite interpolation algorithm applied to the grid displacements. The method, tested using a two-dimensional flow solver that uses an implicit dual-time method for the solution of the unsteady Euler equations on deforming grids, is applicable to problems with time varying geometries arising from aeroelasticity and free surface marine problems. The present work is placed into a multi-block framework and fits into the development of a generally applicable parallel multi-block flow solver. The effect of grid deformation is examined and comparison with rigidly rotated grids is made for a series of pitching aerofoil test cases selected from the AGARD aeroelastic configurations for the NACA0012 aerofoil. The effect of using a geometric conservation law is also examined. Finally, a demonstration test case for the Williams aerofoil with an oscillating flap is presented, showing the capability of the grid deformation technique. Copyright © 2000 John Wiley & Sons, Ltd.

KEY WORDS: Euler equations; unsteady flows; grid deformation; transfinite interpolation; geometric conservation law; multi-block grids

1. INTRODUCTION

An important issue in the development of a computational fluid dynamic (CFD) tool aimed at practical engineering application is its ability to handle complex geometries. The improvement in numerical algorithms, along with increasing computing power, has led to a number of numerical methods with a sufficient level of maturity and reliability to make the solution of the Reynolds-averaged Navier–Stokes equations (within the limitations of turbulence modelling) possible for a wide range of flow problems. However, due to the level of difficulty associated with the grid generation process for complex multi-component configurations, simulations are often restricted to relatively simple geometries. At present, computational tools that combine good flexibility and generality in terms of complex geometries with accurate and efficient flow solvers are still rare.

* Correspondence to: Department of Aerospace Engineering, University of Glasgow, Glasgow, G12 8QQ, UK.

CCC 0271–2091/2000/030285–27$17.50

Received July 1998
Revised January 1999

The grid generation issue is of crucial importance when considering complex geometries. Amongst the most commonly used approaches are the structured grid approach and the unstructured grid approach. Structured grids allow relatively easy implementation and calculation management, but have the disadvantage that grid generation is considerably more difficult for complex geometries, as single block structured grids can only be created on domains that can be mapped onto rectangular parametric space. Alternatively, the unstructured grid approach is more flexible from the point of view of grid generation since no constraints exist on the point connectivity. However, this results in a more complex flow solution and data structure.

A useful compromise between the two approaches is provided by the multi-block technique [1,2]. The multi-block grid consists of an unstructured arrangement of structured grids, where the generation of each structured block is made easier by the partitioning of the computational domain. The multi-block approach allows an extension of the methodologies developed for single-block structured grids. The block decomposition also provides a natural partition of the problem for parallel processing. Examples in the literature of successful applications of multi-block methods for complex geometries are for supersonic air-intakes [3], wing–fuselage configurations [4], shuttle re-entry vehicles [5], general aircraft design problems [6], supersonic internal/external flow interactions [7], multi-element aerofoils [8], twin-jet transport aircraft [9] and complex missile shapes [10]. However, the generation of multi-block structured grids, although based on single-block algorithms, is still a relatively difficult and time-consuming task, especially for the three-dimensional case.

Grid movement becomes an important issue when solving unsteady problems where the mesh has to conform to the instantaneous shape of a deforming body. In most cases (e.g. single aerofoils), rigid body motions can easily be treated by moving the mesh rigidly in response to the motion of the body. However, this approach is no longer applicable if the body deforms as in an aeroelastic problem, if the outer boundaries of the mesh are fixed multi-block boundaries or if the relative motion of a multi-component configuration is to be taken into account (e.g. an oscillating flap). To tackle such problems, efficient grid regeneration or movement techniques are required.

A number of methods exist for deforming the grid, e.g. methods based on a spring analogy [11], but they generally involve either a complete regeneration of the mesh around a deformed geometry or require the solution of a large system of equations for the displacements. This approach has efficiency drawbacks in the context of an unsteady flow simulation, when the grid deformation process must be cheap, ideally taking only a small fraction of the overall CPU time required by the flow solver.

Algebraic techniques are generally quicker than methods based on partial differential equations, but at the expense of grid quality. A technique based on transfinite interpolation (TFI) was used in [12] to regenerate at each time step the mesh around a single aerofoil undergoing an oscillating pitching motion. The same strategy was also employed in the case of an oscillating trailing edge flap [13], and in [14] was extended to the three-dimensional case for unsteady wing problems.

In the work presented here, a moving mesh technique allowing more general deformations around more complex geometries is described. The method uses the TFI algorithm based on a multi-dimensional interpolation of the grid point displacements and is incorporated within a

multi-block environment. The advantage of using interpolated displacements rather than regenerating the grid completely is that it allows the overall quality of the original grid to be retained. The method can be seen as a perturbation method rather than a complete regeneration method. Also, the method is independent of the technique employed to create the initial grid.

In Section 2, a brief description of the flow solution algorithm is given. In Section 3, the formulation of the geometric conservation law used to compute the cell areas on deforming meshes is recalled. Section 4 highlights the need for a deforming grid technique when using multi-block meshes and Section 5 gives a detailed description of the TFI method of perturbation used here to deform the mesh. Finally, Sections 6 and 7 show some results for two types of problems: an NACA0012 aerofoil in pitching oscillation and a multi-element aerofoil with an oscillating flap.

2. FLOW SOLUTION

The two-dimensional Euler equations in Cartesian co-ordinates (x, y) can be written in non-dimensional conservative form as

$$\frac{\partial \boldsymbol{W}}{\partial t}+\frac{\partial \boldsymbol{F}}{\partial x}+\frac{\partial \boldsymbol{G}}{\partial y}=0, \tag{1}$$

where $\boldsymbol{W}$ denotes the vector of conservative variables,

$$\boldsymbol{W}=\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho E\end{pmatrix},$$

and $\boldsymbol{F}$ and $\boldsymbol{G}$ the convective fluxes,

$$\boldsymbol{F}=\begin{pmatrix}\rho U\\ \rho u U+p\\ \rho v U\\ U(\rho E+p)+x_t p\end{pmatrix}, \qquad \boldsymbol{G}=\begin{pmatrix}\rho V\\ \rho u V\\ \rho u V+p\\ V(\rho E+p)+y_t p\end{pmatrix}.$$

In the above equations, ρ, u, v, p and E denote the density, the two Cartesian components of the velocity, the pressure and the specific total energy respectively. The terms U and V are the contravariant velocities, which are defined by

$$U = u - x_t, \qquad V = v - y_t,$$

where x_t and y_t are the grid speeds in the x- and y-directions respectively.

The equations are discretized using a cell-centred finite volume method that transforms the partial differential equations into a set of ordinary differential equations, which can be written as

$$\frac{\partial}{\partial t}(V_{i,j}\boldsymbol{W}_{i,j}) + \boldsymbol{R}_{i,j}(\boldsymbol{W}) = 0, \tag{2}$$

where t is the time, $\boldsymbol{W}_{i,j}$ is the vector of conservative variables, $V_{i,j}$ is the control volume and $\mathbf{R}_{i,j}(\boldsymbol{W})$ is the flux residual for the cell (i, j) that contains all the terms arising from the spatial discretization. The convective fluxes are discretized using the Osher upwind flux difference splitting (FDS) scheme together with a monotone upstream-centred scheme for conservation laws (MUSCL) variable extrapolation to provide second- or third-order accuracy in space. The Van Albada limiter is used to ensure monotonic solutions around shock waves. Finally, the far-field boundary conditions are treated by the characteristic boundary method based on the Riemann invariants.

An unfactored implicit method is used to solve Equation (2). The method combines a dual-time approach to discretize the unsteady equations with an implicit time stepping method for the solution of the steady state problem in pseudo-time [15]. The solution of the large sparse linear system arising from the implicit time discretization is based on a generalized conjugate gradient (GCG) method with a block incomplete lower–upper (BILU) factorization for the preconditioner. This preconditioning strategy allows decoupling of the blocks for the solution of the linear system, resulting in a very efficient parallel implementation of the method. A full description of the method can be found in [16,17].

3. GEOMETRIC CONSERVATION LAW

When computing the flow on a moving grid, the cell areas vary in time and it is therefore important to discretize the time-dependent metrics carefully in order to maintain the conservative properties of the scheme. If the cell areas are calculated analytically in terms of the grid node positions, numerical errors will be introduced in the calculated solution, which increase with time. To avoid such numerical errors, the cell areas must be integrated forward in time by using the same method as used to solve the flow conservation laws [18]. This is achieved by introducing a geometric conservation law (GCL), which can be derived from the continuity conservation law written in integral form by assuming a uniform flow field. This yields

$$\frac{\partial}{\partial t}\int_{\Omega} \mathrm{d}V - \oint_{\partial\Sigma} \boldsymbol{v}\cdot\boldsymbol{n}\,\mathrm{d}\Sigma = 0, \tag{3}$$

where V is the cell area, $\boldsymbol{v}$ is the grid speed, $\boldsymbol{n}$ is the normal area vector and $\partial\Sigma$ is the boundary surface of the control volume Ω. Using the same second-order time discretization as for the flow equations [17], Equation (3) becomes

$$\frac{3V_{i,j}^{n+1} - 4V_{i,j}^{n} + V_{i,j}^{n-1}}{2\Delta t} - \oint_{\partial\Sigma} \boldsymbol{v}\cdot\boldsymbol{n}\, \mathrm{d}\Sigma = 0. \tag{4}$$

This law states that the change in area of each control volume between t^n and t^{n+1} must be equal to the area swept by the cell boundary during $\Delta t = t^{n+1} - t^n$. The volume $V_{i,j}^{n+1}$ at the new time step can then be computed by

$$V_{i,j}^{n+1} = \frac{4V_{i,j}^{n}}{3} - \frac{V_{i,j}^{n-1}}{3} + \frac{2\Delta t}{3} \oint_{\partial\Sigma} \boldsymbol{v}\cdot\boldsymbol{n}\, \mathrm{d}\Sigma, \tag{5}$$

where

$$\oint_{\partial\Sigma} \boldsymbol{v}\cdot\boldsymbol{n}\, \mathrm{d}\Sigma = (\xi_t)_{i+1/2,j} - (\xi_t)_{i-1/2,j} + (\eta_t)_{i,j+1/2} - (\eta_t)_{i,j-1/2},$$

and

$$\xi_t = -(\xi_x x_t + \xi_y y_t), \qquad \eta_t = -(\eta_x x_t + \eta_y y_t).$$

Note that this is an explicit equation for $V_{i,j}^{n+1}$ since the terms ξ_t and η_t are prescribed from the node values. Using the GCL to calculate the volumes numerically rather than analytically yields a self-consistent solution for the effective volume elements. In other words, it ensures that errors arising from the computation of the geometric quantities are consistent with those arising from the integration of the flow equations. The importance of the GCL for flow computations on moving grids has been described in [18–23]. The GCL needs to be evaluated once at every global time step to calculate the new cell areas.

4. DEFORMING GRID ALGORITHM FOR MULTI-BLOCK APPLICATIONS

For unsteady computations, the mesh must be deformed once per (real) time step. Therefore, a fast way of deforming the grid at each time level of the calculation is required. For simple geometries and simple motions, such as single aerofoils in an oscillating pitching motion, the grids can be rigidly rotated with the aerofoil. However, for more complex geometries and motions, such as multi-component aerofoils where the relative motion of the different components must be taken into account, or when the components themselves are deforming, it is necessary to regenerate the grid.

In the multi-block approach, the flow domain is split up into blocks and structured grids are generated in each block with grids in adjacent blocks being matched at common interfaces. With such an approach, the generation of the grid in each block is made easier and the overall quality of the mesh is improved.

By using TFI of displacements, the grid deformation process becomes completely independent of the generation of the initial grid, for which any suitable technique can be used. As well as being fast and simple, an important feature of the deforming grid procedure is that it must maintain the overall quality of the initial grid by introducing minimal distortion of the cells in the regions where the flow is likely to be changing rapidly. It is particularly important to keep the grid as rigid as possible in the near-wall regions and allow cell distortion towards the far-field, or at least in the regions of low gradients where the flow is not changing rapidly.

Also, for multi-block meshes only the mesh blocks embedding the moving surfaces need to be deformed at each time level. In some cases, additional blocks, not directly adjacent to the moving surfaces, may be deformed in order to obtain smoother grids after deformation.

Methods based on the algebraic interpolation of the grid displacements are very attractive as they generally preserve the overall quality of the initial grid as well as being cheap to calculate and easy to formulate. Such methods, also referred to as perturbation methods, have been presented in the literature. In [24,25] the application of a moving grid algorithm using grid displacements was described for single element aerofoils in oscillating pitching motion, where only one boundary was allowed to move. In that case, the interpolation of the displacements across the grid can be performed in one direction only, between the moving solid boundary (aerofoil or wing surface) and the fixed far-field boundary, by using an appropriate interpolation function. The choice of adequate control parameters for different types of grids with different levels of stretching was also addressed therein. However, the direction of interpolation is an additional constraint that is not desirable here when extending the application of the method to multi-block configurations, where the block topology and the orientation of the blocks are not known *a priori*. An extension of the method allowing four moving boundaries and interpolation in each direction is therefore necessary in order to provide sufficient flexibility and generality. Also, due to the complexity in the block connectivity, and in order to avoid a sequential procedure with unnecessary communication between blocks, it is preferable to employ a method that can treat all the blocks independently.

Among the several approaches that were investigated, a method based on the interpolation of the block corner displacements appeared to be the most suitable here. If the displacement of the block corners is known, then the displacement of the block faces and then of the interior points within each block can be interpolated. By using the same interpolation procedure at all block boundary interfaces, the perfect matching of the block boundaries is guaranteed even if the blocks are treated independently.

The technique employed here to deform the grid in each block is a modified version of the TFI method based on the grid displacements (see Figure 1). The TFI grid generation algorithm is a very popular algebraic grid generation technique that effectively interpolates grid points in the computational domain from prescribed points along the block boundaries. The algorithm can equally be applied to the grid point displacements by interpolating the displacements across the grid from the prescribed or calculated displacements along the block boundaries. The cost of the mesh regeneration for the current test cases amounted to less than 0.2% of the total cost for the calculation.

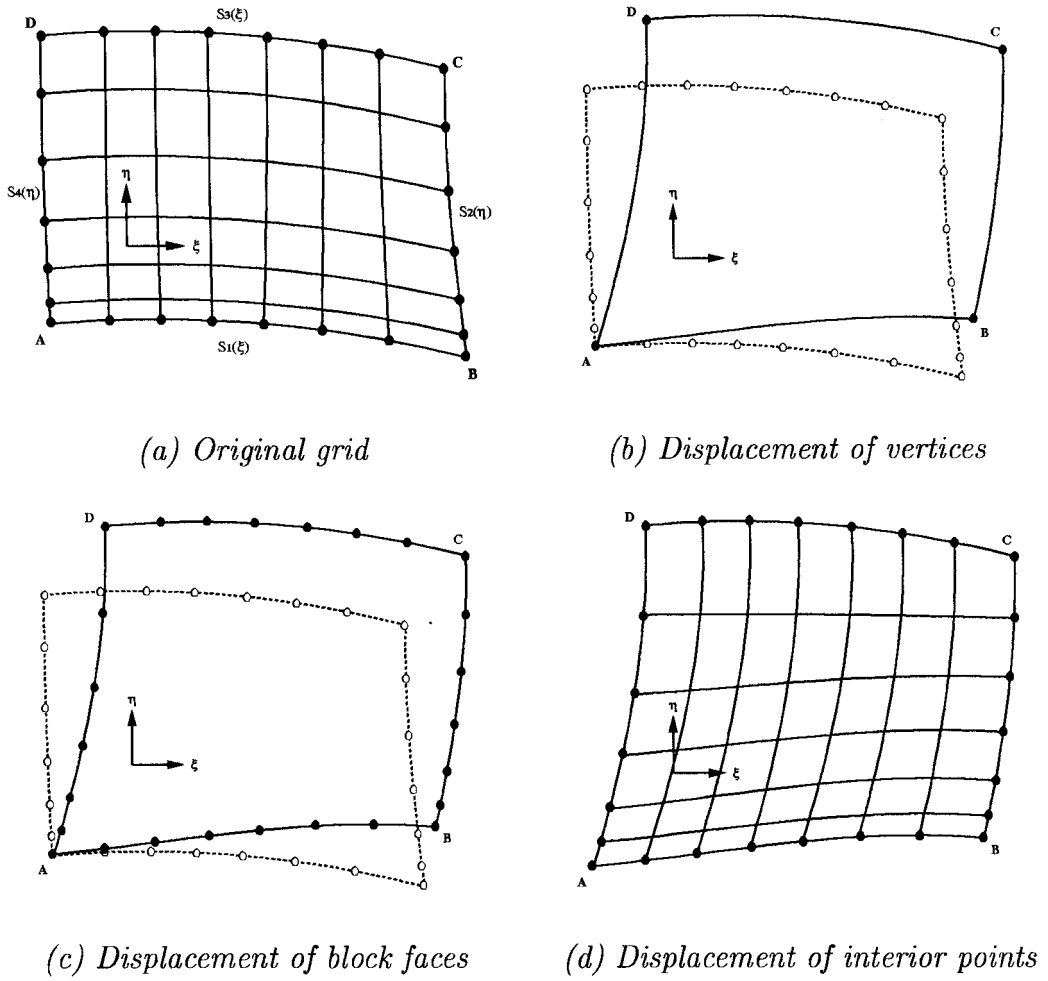


Figure 1. TFI method of perturbations.

5. TFI METHOD OF PERTURBATIONS

5.1. Displacement of the block corners

We first need to determine the displacements of the four block corners (or block vertices) (Figure 1(b)). In order to identify a moving block from a fixed block, a control parameter is introduced into the grid file, which is set to 1 for each moving block and to 0 for all fixed blocks. For each block corner, a search is made over its neighbours, and if at least one of the neighbouring blocks surrounding this corner point (i.e. all blocks having this point as a vertex)

is fixed (i.e. block flagged with the control parameter = 0), then no displacement is allowed for this point. Otherwise, the corner point is moved according to the motion of the solid surface. The displacement of all points lying on a moving surface is assumed known. In the present work, we consider only rigid motions for oscillating pitching aerofoils and oscillating flaps, but the application of the method can be easily extended to more complex configurations and more general deformations [26].

5.2. Displacement of the block faces

The displacements of the four corner points are then used to interpolate the displacement of all the points along the block boundary (Figure 1(c)). We denote the position vector and displacement vector associated with the grid points of the mesh by $\boldsymbol{x}$ and $\mathrm{d}\boldsymbol{x}$ respectively,

$$\boldsymbol{x} = \begin{bmatrix} x(\xi, \eta) \\ y(\xi, \eta) \end{bmatrix}, \qquad \mathrm{d}\boldsymbol{x} = \begin{bmatrix} \mathrm{d}x(\xi, \eta) \\ \mathrm{d}y(\xi, \eta) \end{bmatrix}.$$

Let A and B be the two end-points of a block face with displacements denoted by $\mathrm{d}\boldsymbol{x}_A$ and $\mathrm{d}\boldsymbol{x}_B$ respectively. The displacement $\mathrm{d}\boldsymbol{x}$ of any point P along this boundary can then be obtained by the weighted formula

$$\mathrm{d}\boldsymbol{x} = \left(1 - \frac{a}{c}\right) \mathrm{d}\boldsymbol{x}_A + \left(1 - \frac{b}{c}\right) \mathrm{d}\boldsymbol{x}_B,$$

where $a = \|\vec{AP}\|$, $b = \|\vec{BP}\|$ and $c = \|\vec{AB}\|$. Here, the distances are calculated from the previous grid point co-ordinates. If both end-points are fixed (i.e. zero displacement) then the whole block face remains fixed.

5.3. Displacement of the interior points

Following the original formulation of the TFI algorithm described by Gordon and Hall [27], the general TFI method results in a recursive algorithm, which is here applied to the grid point displacements

$$\mathrm{d}\boldsymbol{x}(\xi, \eta) = \boldsymbol{f}_1(\xi, \eta) + \phi_1^0(\eta)[\mathrm{d}\boldsymbol{x}_{b1}(\xi) - \boldsymbol{f}_1(\xi, 0)] + \phi_2^0(\eta)[\mathrm{d}\boldsymbol{x}_{b3}(\xi) - \boldsymbol{f}_1(\xi, 1)],$$

where

$$\boldsymbol{f}_1(\xi, \eta) = \psi_1^0(\xi)\, \mathrm{d}\boldsymbol{x}_{b4}(\eta) + \psi_2^0(\xi)\, \mathrm{d}\boldsymbol{x}_{b2}(\eta),$$

and $\mathrm{d}\boldsymbol{x}_{b1}$, $\mathrm{d}\boldsymbol{x}_{b2}$, $\mathrm{d}\boldsymbol{x}_{b3}$ and $\mathrm{d}\boldsymbol{x}_{b4}$ are the interpolated displacements along the four block faces. The functions ψ and ϕ are the blending functions in the ξ- and η-directions respectively. These functions are given by the grid point distributions along each block face as

$$\psi_1^0(\xi) = 1 - s_1(\xi),$$

$$\psi_2^0(\xi) = s_3(\xi),$$

$$\phi_1^0(\eta) = 1 - s_4(\eta),$$

$$\phi_2^0(\eta) = s_2(\eta),$$

where $s_1(\xi)$ is the stretching function on the block face $\eta = 0$, $s_2(\eta)$ on the block face $\xi = 1$, $s_3(\xi)$ on the block face $\eta = 1$, $s_4(\eta)$ on the block face $\xi = 0$. The stretching functions are given by the length of the boundary curve in physical space from the first end-point to the point under consideration, relative to the total length of the boundary curve, e.g.

$$s_1(\xi) = \frac{\text{length from } A \text{ to } \boldsymbol{x}(\xi, 0)}{\text{length of curve from } A \text{ to } B}.$$

The co-ordinates of the new grid points (Figure 1(d)) are then simply obtained by

$$\boldsymbol{x}(\xi, \eta) = \boldsymbol{x}_0(\xi, \eta) + \mathrm{d}\boldsymbol{x}(\xi, \eta),$$

where $\mathrm{d}\boldsymbol{x}$ is the interpolated displacement and $\boldsymbol{x}_0$ is the vector position for the initial undisturbed grid.

6. RESULTS FOR NACA0012 AEROFOIL

6.1. Test case

We present the results obtained using a dual-time method [16,17] for a standard pitching aerofoil test case selected from the AGARD database [28]. The test case examined here, referred to as CT2, is for an NACA0012 aerofoil with a free-stream Mach number of 0.6. The periodic motion of the aerofoil is defined by the angle of attack as a function of time as

$$\alpha(t) = \alpha_{\mathrm{m}} + \alpha_0 \sin(\omega t), \quad \text{with } \begin{cases} \alpha_{\mathrm{m}} = 3.16^\circ \\ \alpha_0 = 4.59^\circ \end{cases},$$

where α_{m} is the mean incidence, α_0 is the amplitude of the pitching oscillation and ω is the angular frequency of the motion, which is related to the reduced frequency k by

$$k = \frac{\omega c}{2U_\infty}, \quad \text{with } k = 0.0811,$$

where c is the aerofoil chord and U_∞ is the free-stream velocity. The aerofoil oscillates about its quarter chord and the location of the moment centre used to calculate the pitching moment is $x_m = 0.273$.

The grid used here for the computations is a C-type Euler grid consisting of 128×32 mesh cells with 96 cells on the aerofoil. The far-field boundary is situated approximately 15 chords away from the aerofoil surface. The average spacing for the first layer of points from the aerofoil surface varies between 0.0027, at the leading edge, and 0.010, at the trailing edge.

A grid refinement study was performed in [17] and demonstrated that spatially accurate results are obtained on this grid. A time step refinement study was also carried out that showed that 40 steps per cycle could be used to obtain accurate results temporally.

6.2. Effect of grid deformation

In order to investigate the effect of the grid deformation and grid distortion on the solution accuracy, we first consider a simple problem consisting of an NACA0012 aerofoil where the grid is partitioned into eight blocks: two layers of four blocks wrapped around the aerofoil, four inner blocks and four outer blocks. This partitioning of the grid was chosen to allow different ways of deforming the grid, by allowing either all the blocks to deform or only the four inner blocks (see Figure 2). In both cases, the results were compared with those obtained on a rigidly rotating grid.

When all the eight blocks are allowed to deform, the deforming process results in a grid where the two inner blocks surrounding the aerofoil surface are mainly rotated rigidly (no distortion is introduced around the aerofoil surface), leaving all the outer blocks to absorb the deformation. This has the advantage of preserving the good quality of the grid in the aerofoil region and deforming the grid in the far-field region, where the flow is not changing rapidly.

On the other hand, when only the four inner blocks are allowed to deform (i.e. the four outer blocks are fixed), all the deformation has to be absorbed by a smaller region, therefore resulting in a more distorted grid, especially in the near-wall region. Although the grid's orthogonality near the aerofoil surface is slightly affected, the general good quality of the initial grid is preserved through the regeneration process. The two grids are shown here for a rotation angle of 10° of the aerofoil surface about its quarter chord. The moving blocks are plotted with solid lines and the fixed blocks are plotted with dotted lines.

These results suggest that if surface deformations can be absorbed over a large enough region for aeroelastic or free surface applications, then the quality of the grid in the part of the flow domain which matters most (i.e. close to the surface) will be maintained.

Figure 3 shows a comparison of the results for the AGARD test case CT2 obtained on three different grids: rotating grid, deforming grid with all the blocks moving and deforming grid with only the four inner blocks moving. The results were obtained here with 40 time steps per cycle, and are shown in terms of normal force coefficient and pitching moment coefficient loops. No large difference in the results can be seen between the three grids, apart from a small discrepancy between the two deforming grids and the rotating grid as the incidence rises.

Since the two deforming grids (one being distorted around the aerofoil surface, the other one being mainly rotated rigidly) give similar results, the mismatch in the solutions compared with the rotating grid is not caused by the distortion of the grid around the aerofoil surface. It is believed that the discrepancies in the C_n and C_m prediction is caused by the different position of the wake line (i.e. cut line at the aerofoil trailing edge), which has a different orientation when the grid is rotated. This may affect the flow solution in the wake region and the pressure

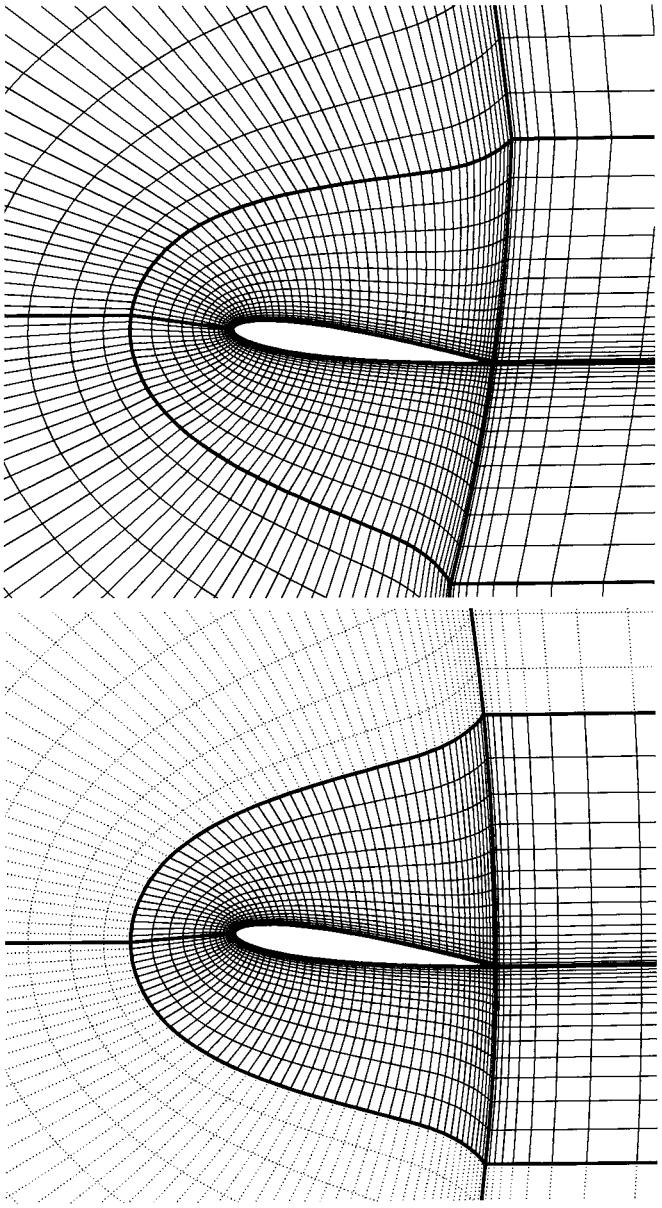


Figure 2. Deformed grid for the NACA0012 aerofoil: eight moving blocks (top) and four moving blocks (bottom). Moving blocks are shown with solid lines and fixed blocks with dotted lines.

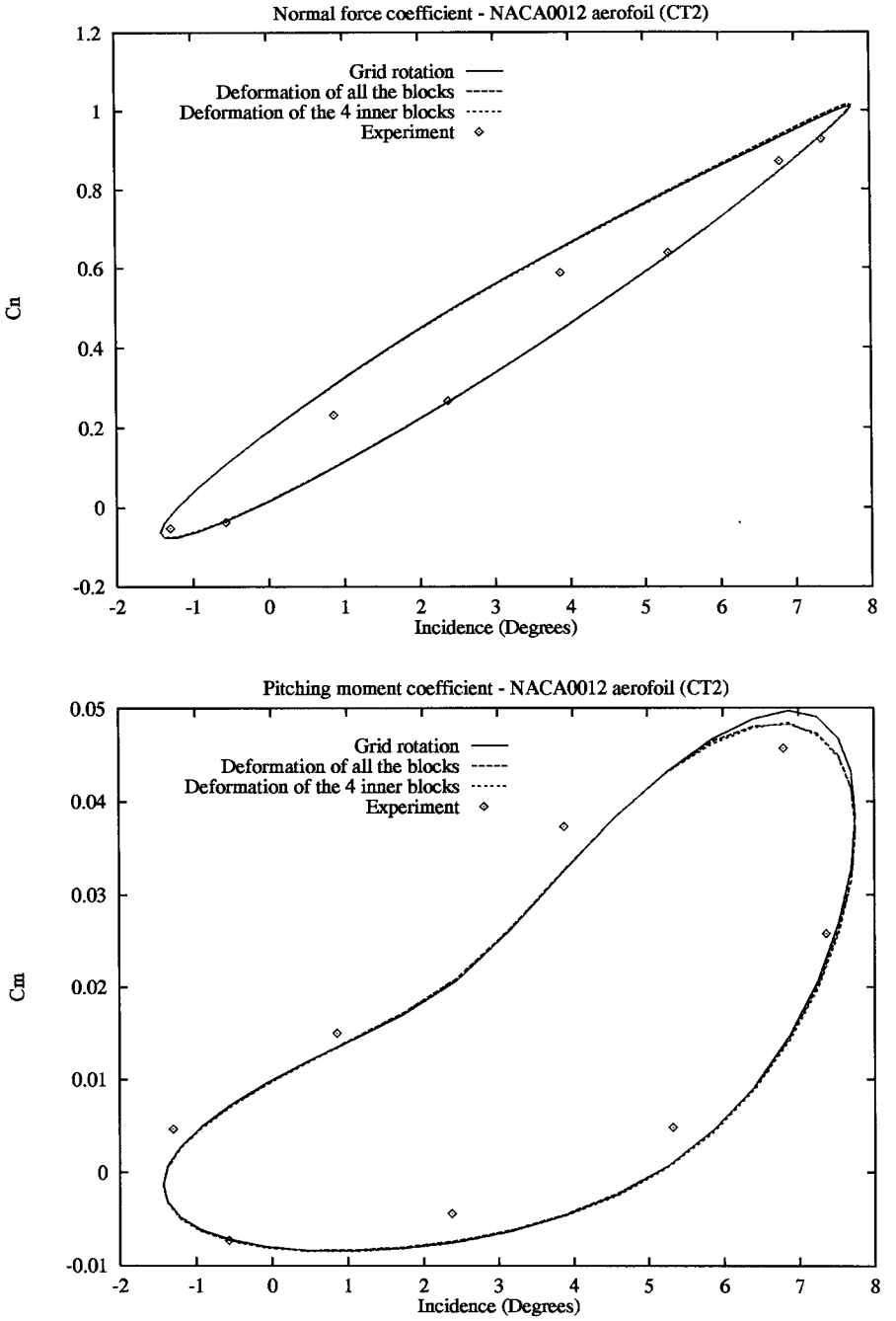


Figure 3. Effect of grid deformation on solution accuracy for case CT2.

at the trailing edge of the aerofoil, therefore causing a small variation of the shock wave location developing on the upper surface of the aerofoil as the incidence rises.

Another parameter that can effect the quality of the grid is the size of the deforming region. This effect is investigated here by performing comparisons of the results using different partitioning of the original eight-block grid, where the number of cells of the inner blocks and outer blocks in the normal direction are varying such as to produce grids were the size of the deforming region is changing (see Figure 4). The number of cells in the streamwise direction is the same for the three grids, only the number of cells in the normal direction between the inner blocks and the outer blocks are varying. *Grid a* consists of 24 cells in the inner block region and eight cells in the outer blocks regions, *Grid b* consists of 16 and 16 cells respectively and *Grid c* consists of eight cells and 24 cells respectively.

When all the eight blocks are moving, the effect is almost insignificant since the inner region is mainly rotated rigidly and the outer region is always big enough to absorb smoothly all the deformations.

On the other hand, the effect can be clearly observed when the outer blocks are fixed and when only the four inner blocks are moving. In that case, the distance between the moving boundary (i.e. the aerofoil surface) and the fixed boundary (i.e. the boundary between the inner blocks and the outer blocks) is varying. When the deforming region is small (Grid c), the grid is highly stretched and distorted.

Figure 5 shows a comparison of the results for the AGARD test case CT2 for varying sizes of the deforming region. Here, only the four inner blocks are moving, the four outer blocks being fixed. No difference in the results is observed between the three grids at low incidence, but as the incidence rises, the deformation of the grids become more important and its effect becomes visible. The solution obtained for Grid c (the grid with the smaller inner region) shows a small departure from the two other solutions, therefore demonstrating that high distortion of the grid can affect the accuracy of the flow solution.

This shows that it is important to choose a deforming region that is big enough to ensure that the grid distortion has no significant effect on the flow solution. This will be further demonstrated in the next section for a multi-element aerofoil problem.

6.3. Effect of GCL

The effect of the GCL on the solution accuracy when using deforming grids is shown in Figure 6 in terms of normal force coefficient and pitching moment coefficient loops. The results were obtained on the eight-block grid with the four inner blocks deforming and the four outer blocks being fixed. Comparison of the C_n and C_m with the results obtained when computing the cell areas algebraically shows no significant difference in the solution accuracy even when using very large time steps. The results are plotted here for ten steps per cycle.

The difference between the GCL and the algebraic computation of the cell areas is small despite the large time step employed, therefore indicating that the grid deformation technique employed here does not produce significant distortion of the mesh cells.

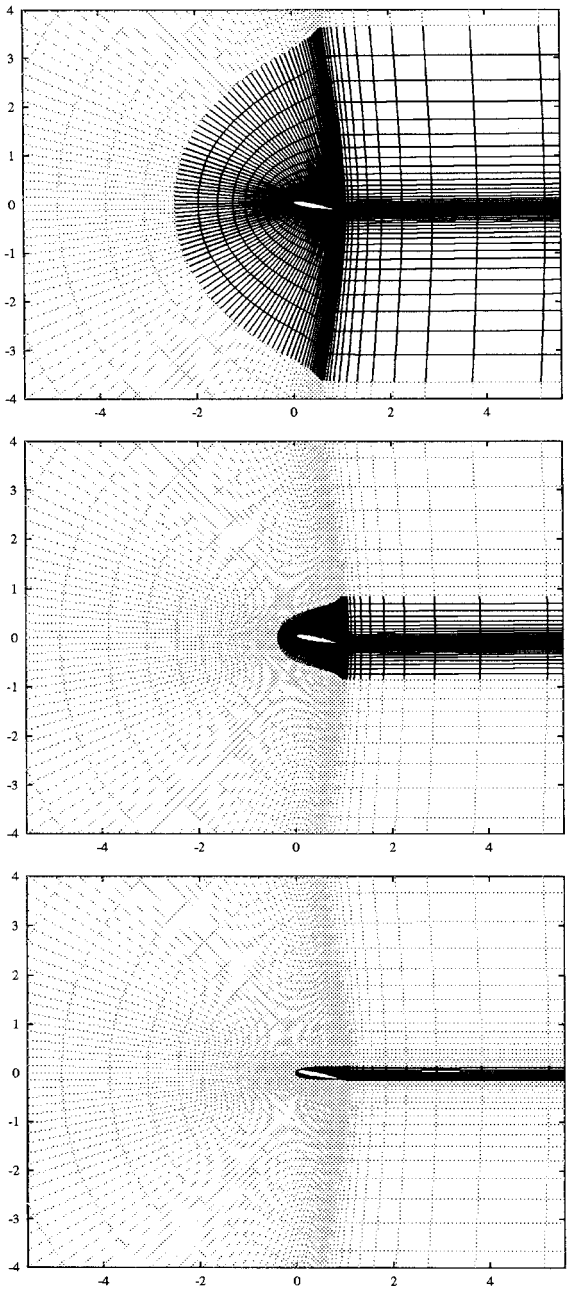


Figure 4. Varying size of the deforming region: Grid a (top), Grid b (middle) and Grid c (bottom). Moving blocks are shown with solid lines and fixed blocks with dotted lines.

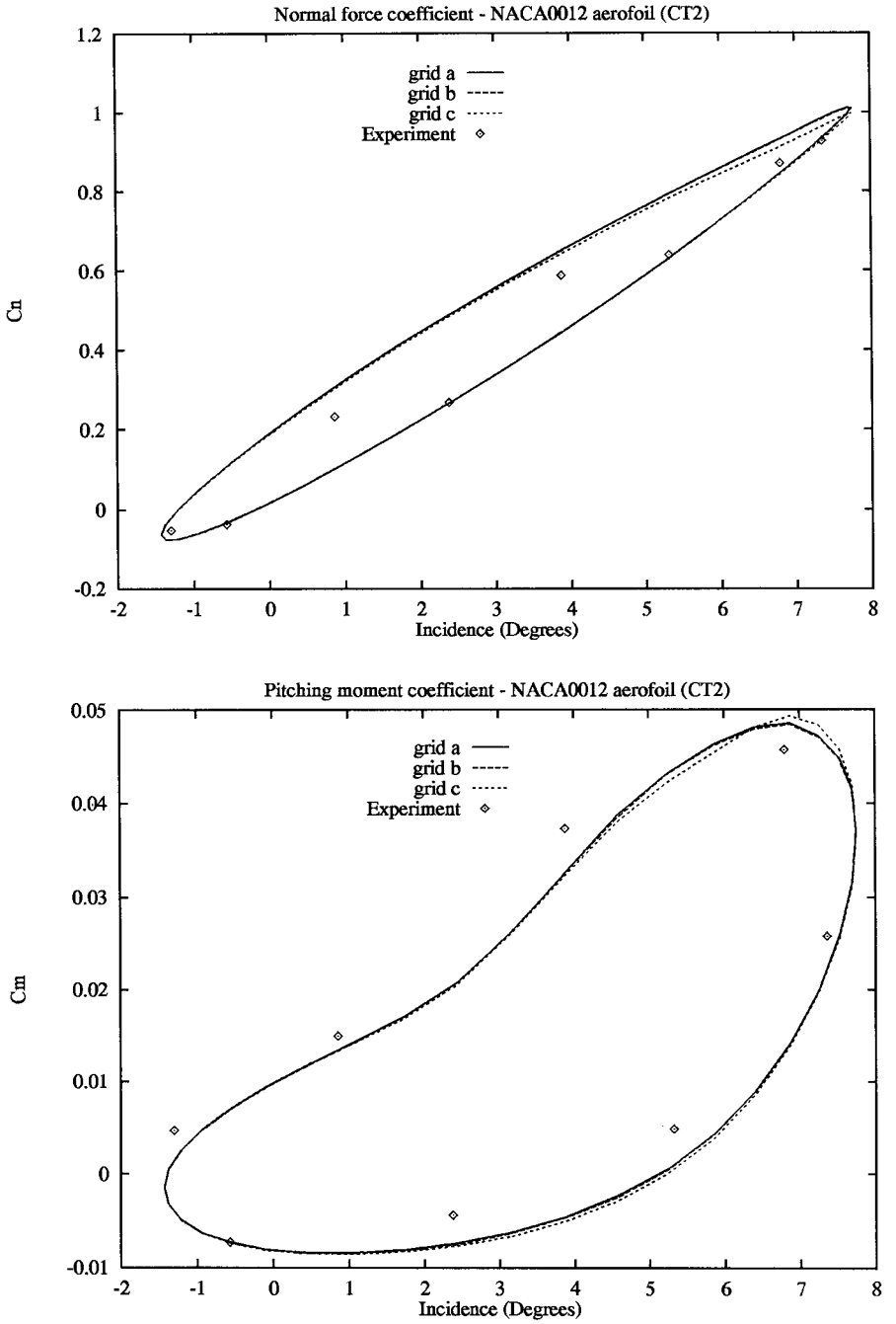


Figure 5. Effect of the size of the deformation region on solution accuracy for case CT2.

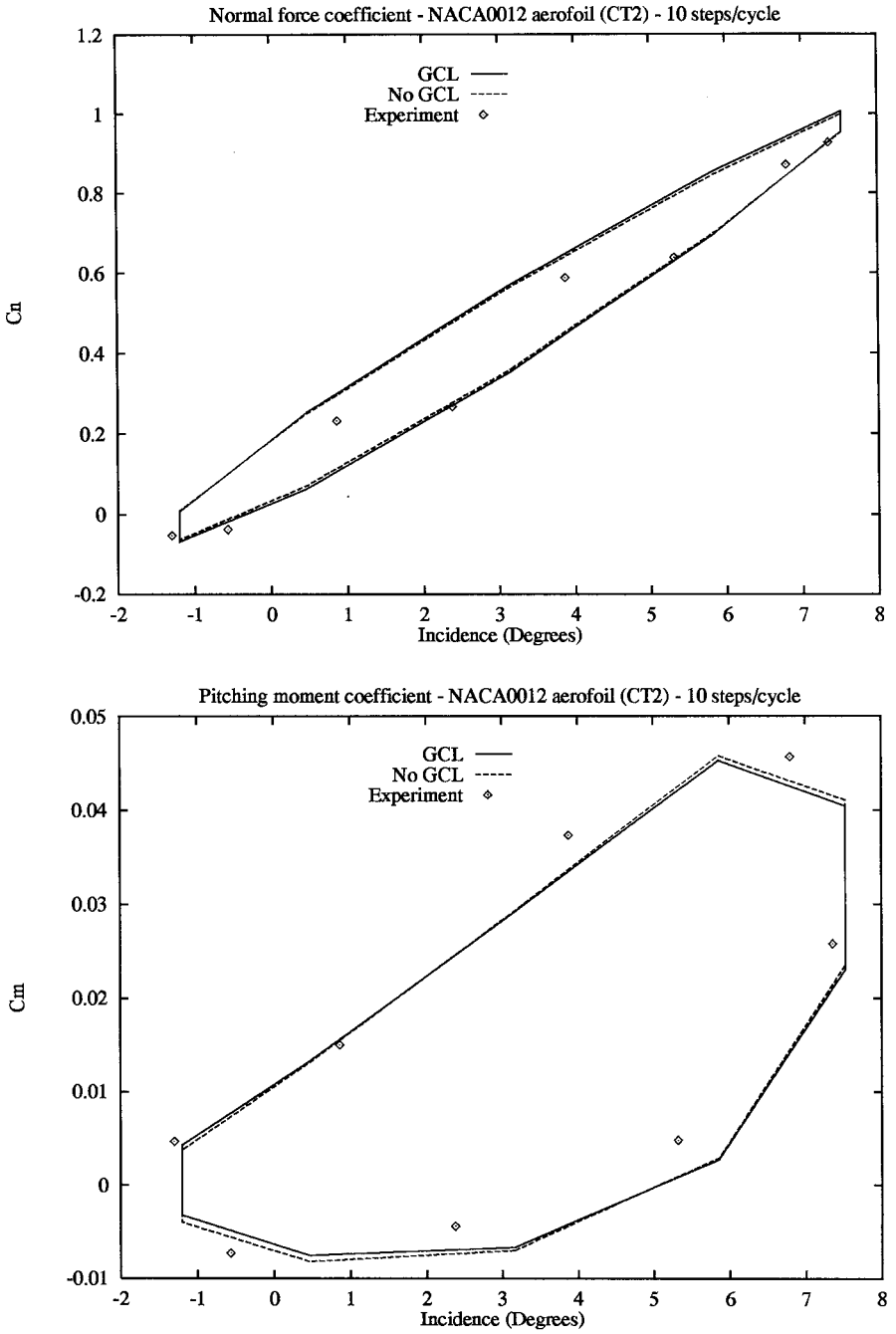


Figure 6. Effect of the GCL solution accuracy for case CT2.

7. MULTI-ELEMENT DEMONSTRATION: WILLIAMS AEROFOIL

In order to demonstrate the performance of the method for a genuinely multi-block case, we consider here a demonstration case for the Williams aerofoil (Configuration B) [29] with an oscillating flap. The free-stream Mach number for this test case is $M_\infty = 0.58$. Several test cases were considered for this configuration with various amplitudes for the flap deflection up to 14°. The mesh consists of 15 blocks with 11228 grid points (10367 mesh cells). The original undeformed grid for the Williams aerofoil is shown in Figure 7, together with the block topology and the block numbering. The blocks that are allowed to deform are selected by the user by means of a graphical interface. The choice is made to retain the overall quality of the grid. For this particular problem, a minimum of four moving blocks for very small flap deflections and up to a maximum of 12 moving blocks for larger amplitudes (over 10°) were considered. Figures 8 and 9 show a close up view of the flap region for various combinations of deforming blocks. The deformed grids are shown here for a flap deflection of 14°. When only the blocks adjacent to the moving surface (i.e. oscillating flap) are deforming, the local distortion is important where the flap motion is biggest at the trailing edge. Some extra distortion of the grid towards the leading edge of the flap is incurred when block 5 is allowed to deform since this alters the point distribution along the boundary between blocks 4 and 5. However, the change in point distribution along the boundary between blocks 4 and 5 allows the grid quality to be maintained in the area of largest flap movement, at the flap trailing edge. This again suggests that for large and general motions, allowing the boundary changes to be absorbed over a large region of the grid is the best strategy.

We present here the results obtained for an oscillating flap, where the periodic motion of the flap is defined by its deflection angle as

$$\alpha(t) = \alpha_m + \alpha_0 \sin(\omega t), \quad \text{with } \begin{cases} \alpha_m = 7.0° \\ \alpha_0 = 7.0° \end{cases},$$

where α_m is the mean incidence, α_0 is the amplitude of the flap oscillation and $k = \omega c/2U_\infty = 0.0814$ is the reduced frequency. The deflection of the flap is counted with respect to its position for the original Williams aerofoil (Configuration B), counted positive when the flap is deflected nose up and negative when deflected nose down. The flap is rigidly rotated about the point situated in front of its leading edge with co-ordinates $x = 0.98$ and $y = -0.07$, where the non-dimensionalization is defined here with respect to the chord of the main aerofoil.

An initial solution was first obtained by solving a steady state problem at the mean deflection angle and the flap was then set in motion to solve the unsteady problem for the oscillating flap. The calculations are continued until a periodic solution is obtained, usually after two or three cycles.

The results are shown in Figure 10 in terms of lift coefficient and pitching moment coefficient calculated about the quarter-chord of the main aerofoil. Results are shown here for various combinations of moving blocks. As expected, the solution obtained with the minimum number of deforming blocks is slightly different from the three other combinations, due to the greater local distortion generated in the near flap region. However, consistent results are obtained on all grids. Comparison of the solution between GCL and algebraic computation of

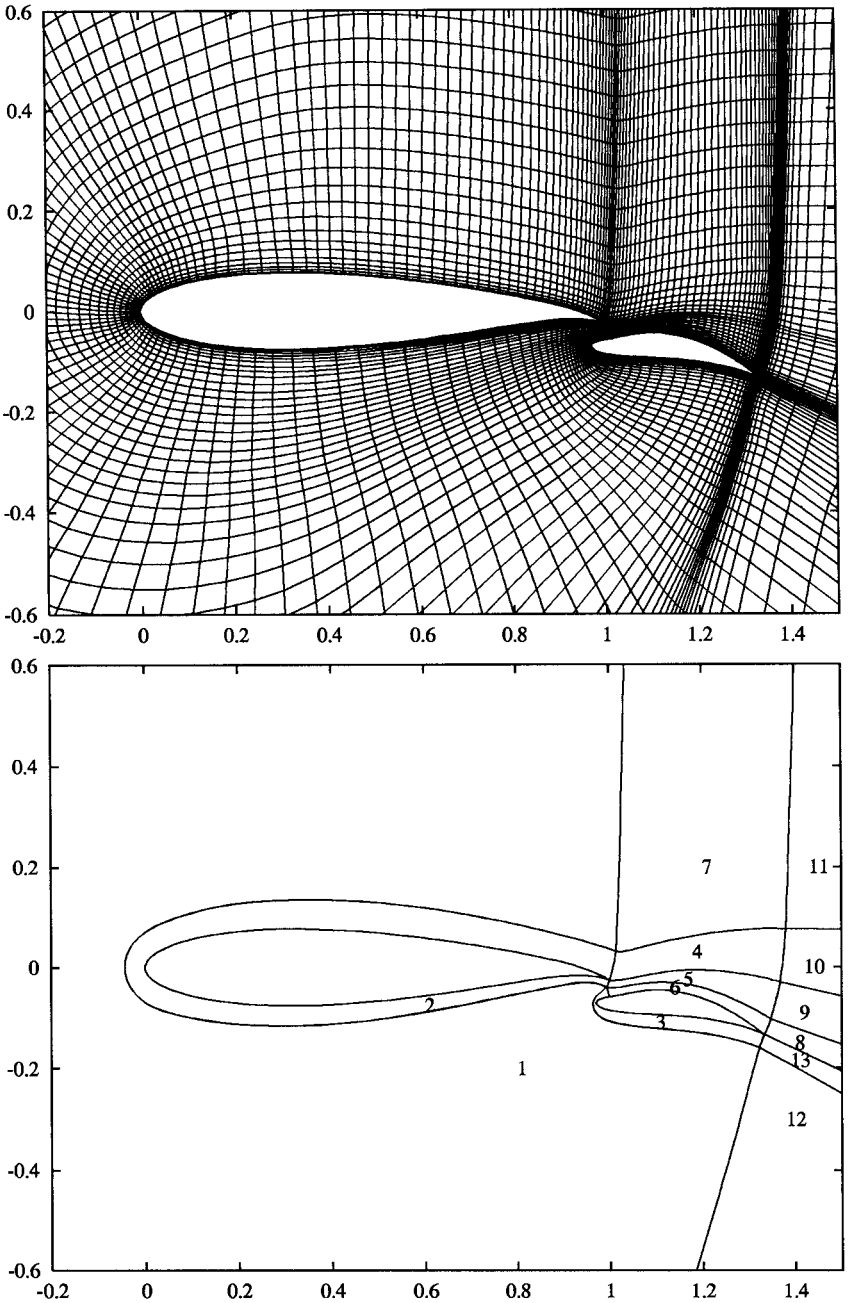


Figure 7. Multi-block grid for the original Williams aerofoil (top) and grid topology and block numbering (bottom).

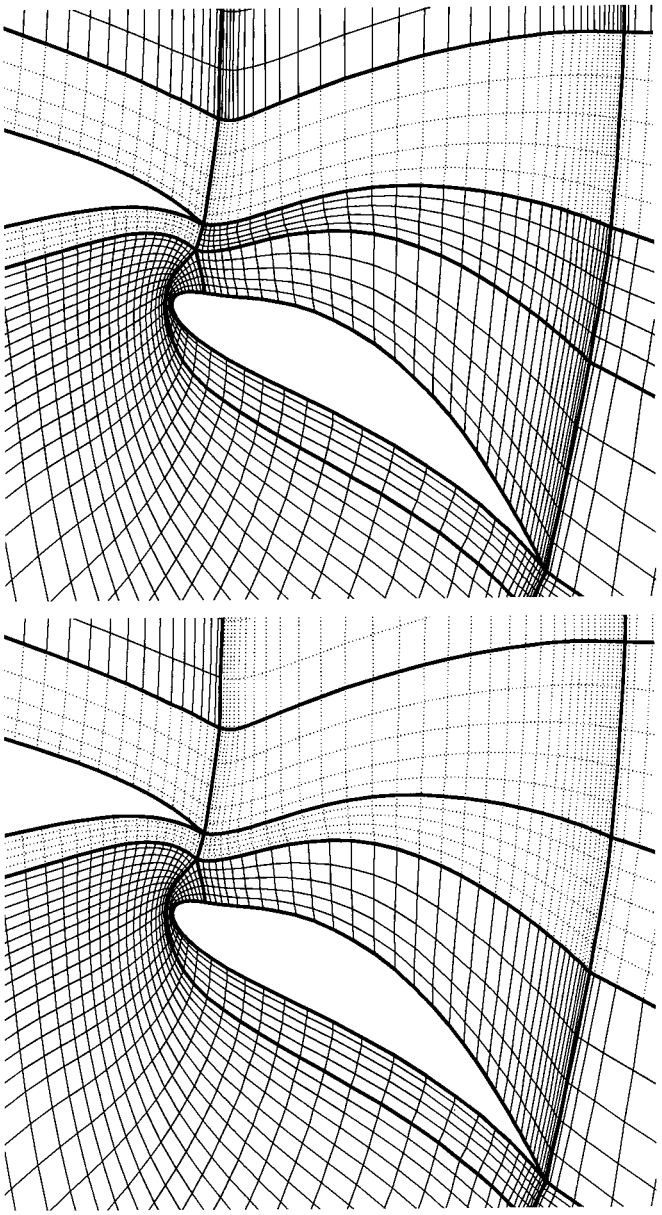


Figure 8. Deformed grid for the Williams aerofoil: six moving blocks (top) and eight moving blocks (bottom). Moving blocks are shown with solid lines and fixed blocks with dotted lines.

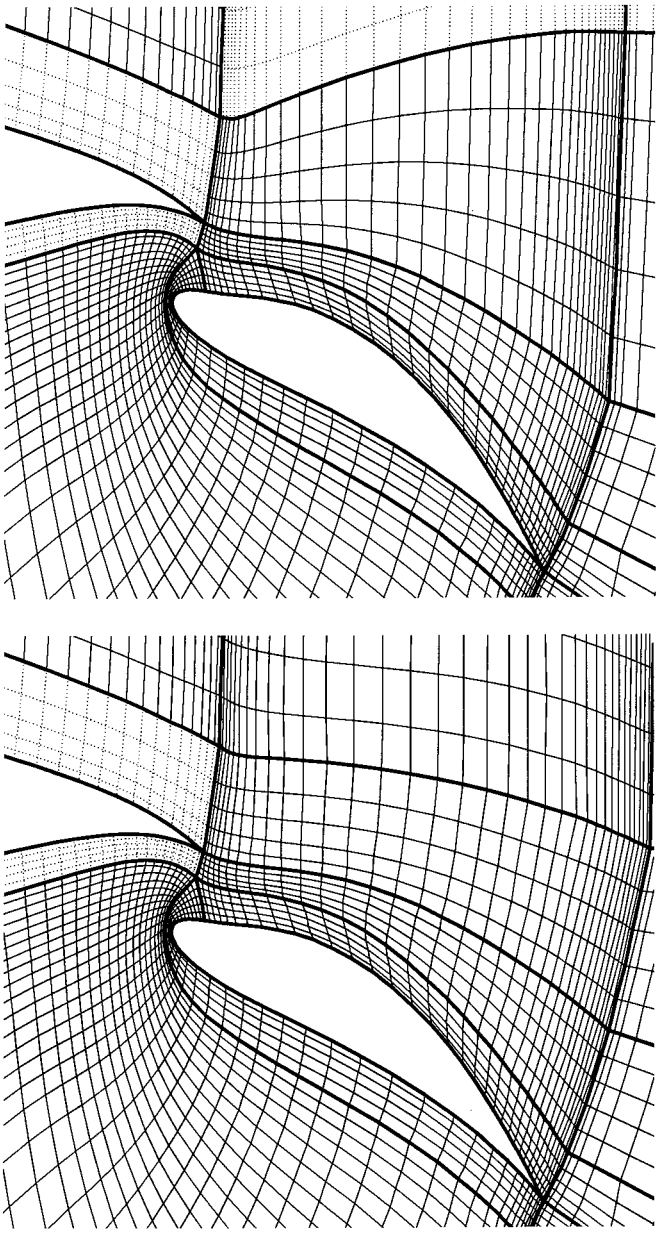


Figure 9. Deformed grid for the Williams aerofoil: ten moving blocks (top) and 12 moving blocks (bottom). Moving blocks are shown with solid lines and fixed blocks with dotted lines.

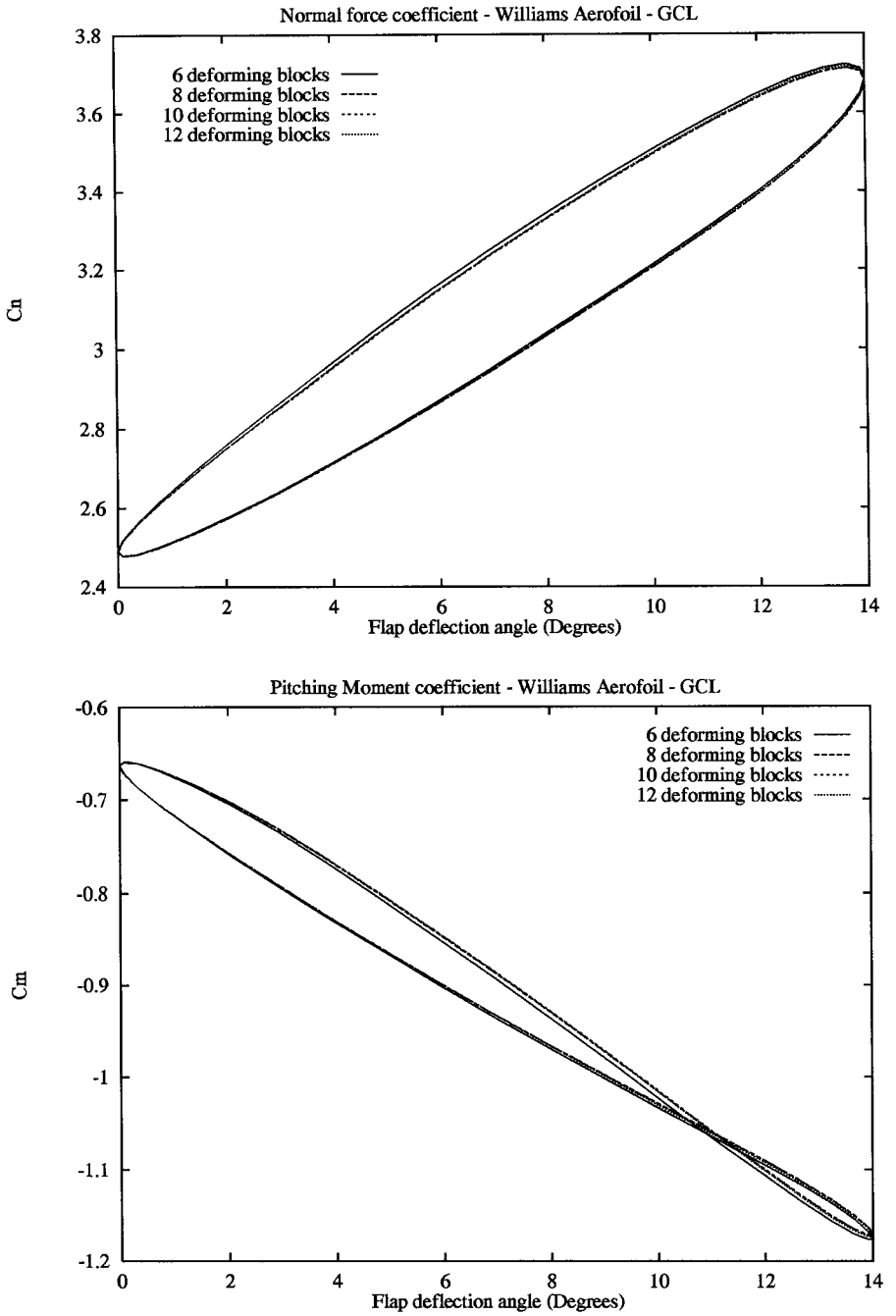
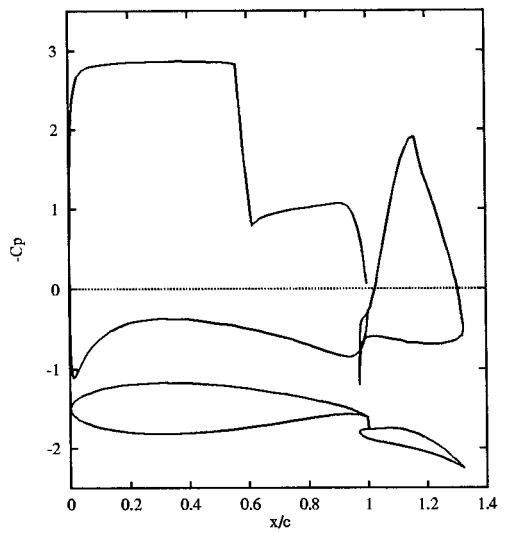
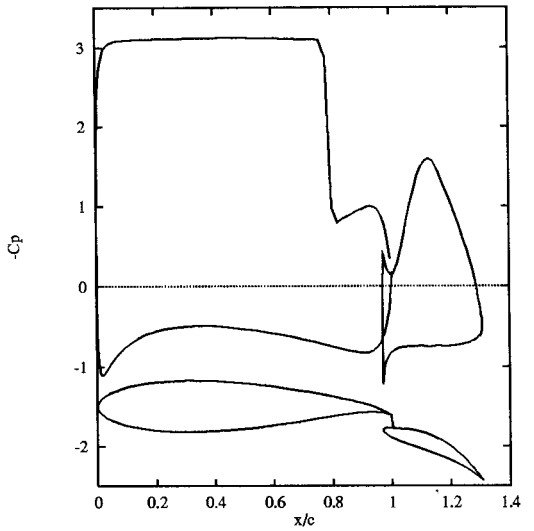


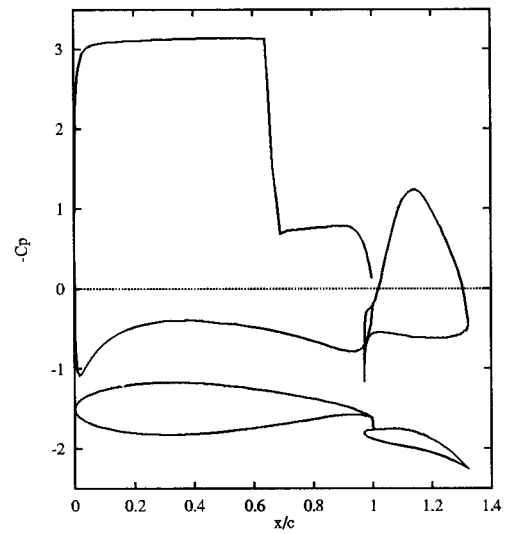
Figure 10. Williams aerofoil: lift coefficient (top) and pitching moment coefficient (bottom).



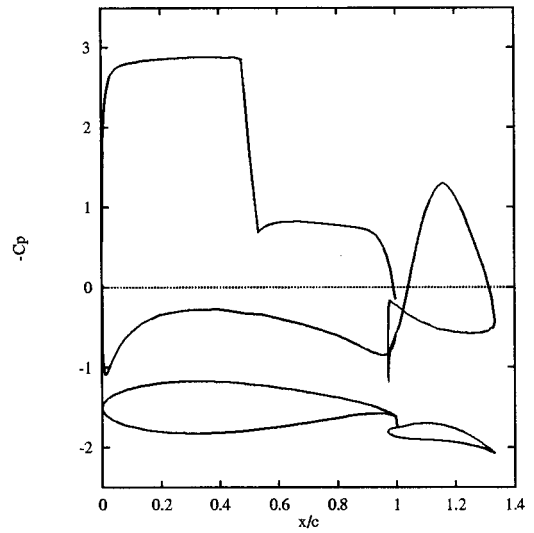
(1) $\alpha = 7.0^o \nearrow$



(2) $\alpha = 14.0^o$



(3) $\alpha = 7.0^o \searrow$



(4) $\alpha = 0.0^o$

Figure 11. Pressure distribution for Williams aerofoil with oscillating flap.

the cell areas showed no difference at all for this particular problem even for very large time steps, further highlighting the good properties of the grid deforming technique employed here. The results are plotted here for 40 steps per cycle.

Figure 11 show the corresponding C_p distributions at four different deflection angles during the cycle. As the flap is deflected downward (nose up), the shock wave located on the upper surface of the main aerofoil moves downstream and the pressure plateau upstream of the shock increases slightly, resulting in a significant increase of lift. At maximum flap deflection (14°), the shock is located at approximately 70% of the main aerofoil chord, whereas at minimum deflection (corresponding to a zero deflection angle), the shock is situated at about 50% chord. The C_p distributions also show that the extra lift generated by the flap itself increases significantly as the flap moves downward (nose up). For this particular test case, the lift coefficient varies by about $\pm 20\%$ during the cycle.

Figure 12 shows some pressure contours at four different angles, showing clearly the displacement of the strong shock wave on the upper surface of the main aerofoil.

8. CONCLUSIONS

A method for grid deformation is described. This method, based on the TFI of the displacement of block boundaries, provides an efficient way of deforming the grid during an unsteady flow calculation. The method is flexible and is independent of the initial grid generation. Good quality grids have been obtained for two types of inviscid test case, including an aerofoil oscillating in pitch and a multi-element aerofoil with an oscillating flap.

The general results obtained for these test cases have shown that the deformation of the grid has no significant effect on the solution accuracy when compared with a rigidly rotating grid, therefore suggesting that the grid deformation technique employed here and based on a TFI algorithm is a suitable approach to handle deforming grids for similar cases. The test cases indicated that the interpolation effectively rotates the grid points close to the moving surface if the movement is absorbed over a large region in the overall grid. This property is desirable when considering more general aeroelastic or free surface motions.

A GCL was incorporated into the flow solver but was found to have no significant effect, even for very large time steps.

Extensions of this work involve evaluating the method for viscous calculations and for more general motions (e.g. deforming aerofoils). Current applications of the method include

- aeroelastic problems,
- free surface problems,
- grid treatment for icing problems,
- regeneration of grids for different aerofoils based on a high quality grid for a different aerofoil,
- dynamic design optimization.

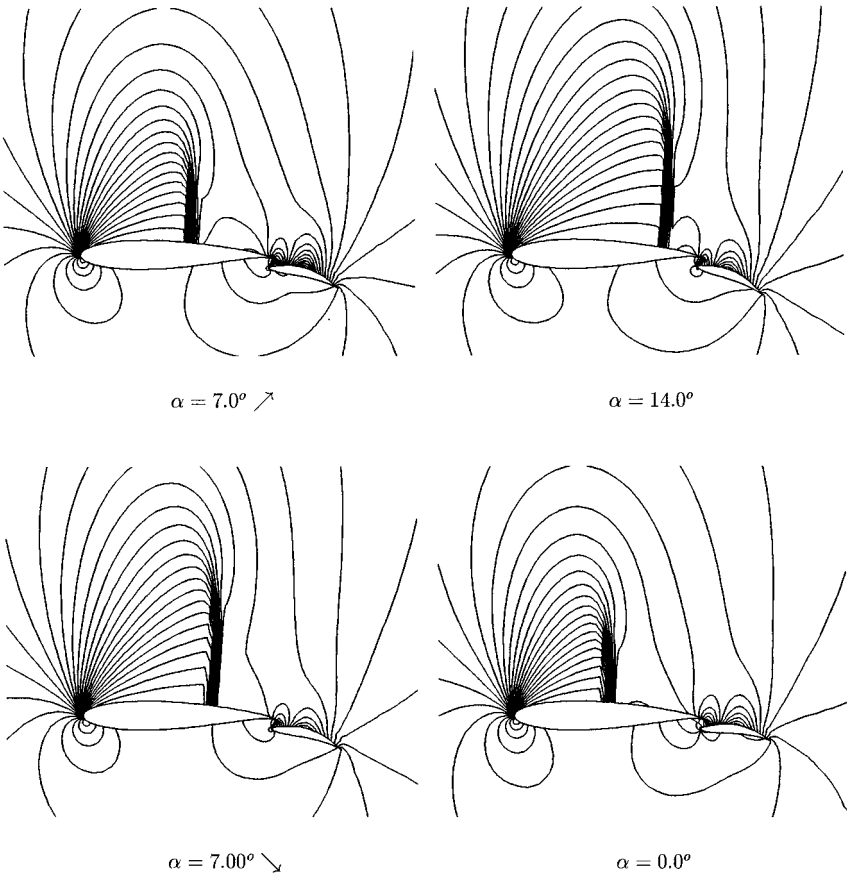


Figure 12. Pressure contours for Williams aerofoil with oscillating flap.

ACKNOWLEDGMENTS

This work has been partially sponsored by British Aerospace Contract SPO S 0104458, Defence Evaluation and Research Agency (DERA) contract FRN1c/407, Engineering and Physical Sciences research Council/Ministry of Defence Grant GR/K55455 and the Office of Science and Technology LINK award Grant DERA ASF/2351U.

APPENDIX A. NOMENCLATURE

c aerofoil chord
C_m pitching moment coefficient
C_n normal force coefficient

C_p pressure coefficient
$\mathrm{d}\boldsymbol{x}$ Displacement vector
E specific total energy
f_1 Interpolation function
$\boldsymbol{F}$, $\boldsymbol{G}$ convective fluxes
k reduced frequency
M_∞ free-stream Mach number
$\boldsymbol{n}$ normal vector
p static pressure
$\boldsymbol{R}_{i,j}$ flux residual
s stretching function
t time
u, v Cartesian velocity components
U, V contravariant velocities
U_∞ free-stream velocity
$V_{i,j}$ cell volume
$\boldsymbol{v}$ velocity vector
$\boldsymbol{W}$ vector of conservative variables
$\boldsymbol{x}$ position vector
x, y Cartesian co-ordinates
x_m moment centre
x_t, y_t grid speeds

Greek letters

α angle of attack
α_m mean angle
α_0 amplitude of oscillation
Δt real time step
η, ξ generalized co-ordinates
ω angular frequency
ρ density
Φ,Ψ blending functions
Ω control volume
Σ boundary surface

Subscripts

i, j computational cell
∞ free-stream conditions
t time derivative
x, y spatial derivatives

Superscripts

n time level of the approximation in real time

REFERENCES

1. A. Rizzi, P. Eliasson, I. Linblad, C. Hirsch, H. Lacor and J. Haeuser, 'The engineering of multiblock/multigrid software for Navier–Stokes flows on structured meshes', *Comput. Fluids*, **22**, 341–367 (1992).
2. N.P. Weatherhill and C.R. Forsey, 'Grid generation and flow calculations for aircraft geometries', *J. Aircraft*, **22**, 855–860 (1985).
3. G. Freskos and O. Penanhoat, 'Numerical simulation of the flow field around supersonic air-intakes', *J. Eng. Gas Turbines Power*, **116**, 116–123 (1994).
4. F. Ghaffari, J.M. Luckring, J.L. Thomas, B.L. Bates and R.T. Biedron, 'Multiblock Navier–Stokes solutions about the F/A-18 wing-LEX-fuselage configuration', *J. Aircraft*, **30**, 293–303 (1993).
5. P.A. Gnoffo, K.J. Weilmuenster and S.J. Alter, 'Multiblock analysis for shuttle orbiter re-entry heating form Mach 24 to Mach 12', *J. Spacecraft Rockets*, **31**, 367–377 (1994).
6. G. Kalitzin, A.R.B. Gould and J.J. Benton, 'Application of two-equation turbulence models in aircraft design', *AIAA 96-0327*, 34th Aerospace Sciences Meeting and Exhibit, Reno, NV, 15–18 January 1996.
7. M. Kathong, R.E. Smith and S.N. Timari, 'Application of multiple grids topology to supersonic internal/external flow interactions', *J. Aircraft*, **27**, 242–252 (1990).
8. T.E. Nelson, D.W. Zingg and J.W. Johnston, 'Compressible Navier–Stokes computations of multielement airfoil flows using multiblock grids', *AIAA J.*, **32**, 506–511 (1994).
9. S. Rill and K. Becker, 'Simulation of transonic flows over twin-jet transport aircraft', *J. Aircraft*, **29**, 640–646 (1992).
10. Th. Streit, 'Euler and Navier–Stokes solutions for supersonic flow around a complex missile', *J. Spacecraft Rockets*, **31**, 600–608 (1994).
11. K. Nakahashi and G.S. Deiwert, 'Self adaptive grid method with application to airfoil flow', *AIAA J.*, **25**, 513–520 (1987).
12. A.L. Gaitonde and S.P. Fiddes, 'A moving mesh system for the calculation of unsteady flows', *AIAA Paper 93-0641*, 1993.
13. A.L. Gaitonde, 'A dual-time method for the solution of the 2D unsteady Navier–stokes equations on structured moving meshes', *Report No. 716*, Department of Aerospace Engineering, University of Bristol, May 1995.
14. A.L. Gaitonde and S.P. Fiddes, 'A three-dimensional moving mesh method for the calculation of unsteady transonic flows', *Aeronaut. J. R. Aeronaut. Soc.*, **99**, 150–160 (1995).
15. A. Jameson, 'Time dependent calculations using multigrid with applications to unsteady flows past airfoils and wings', *AIAA Paper 91-1596*, June 1991.
16. F. Cantariti, M. Woodgate and K.J. Badcock, 'Approximate Jacobians for the Euler and Navier–Stokes equations', *Aero Report 9705*, University of Glasgow, February 1997.
17. L. Dubuc, F. Cantariti, M. Woodgate, B. Gribben, K.J. Badcock and B.E. Richards, 'Solution of the Euler unsteady equations using an implicit dual time method', *AIAA J.*, **36**, 1417–1424 (1998).
18. P.D. Thomas and C.K. Lombard, 'Geometric conservation law and its application to flow computations on moving grids', *AIAA J.*, **17**, 1030–1037 (1979).
19. C.B. Allen, 'Central-difference and upwind-biased schemes for steady and unsteady Euler aerofoil computations', *Aero. J.*, **99**, 52–62 (1995).
20. C.B. Allen, 'Adaption by grid motion for unsteady Euler aerofoil flows', *AGARD Paper 36*, Proceedings 77th Fluid Dynamics Panel Meeting and Symposium, Progress and Challenges in CFD Methods and Algorithms, Seville, October 1995, pp. 36.1–36.10.
21. A.L. Gaitonde, 'A dual-time method for the solution of the unsteady Euler equations', *Aero. J.*, **98**, 283–291 (1994).
22. M. Lesoinne and C. Farhat, 'Geometric conservation laws for aeroelastic computations using unstructured dynamic meshes', *AIAA Paper 95-1709-CP*, 1995.
23. V. Venkatakrishnan and D.J. Mavriplis, 'Implicit method for the computation of unsteady flows on unstructured grids', *AIAA Paper 95-170-CP*, 1995.
24. A.L. Gaitonde and S.P. Fiddes, 'A comparison of a cell-centre method and a cell-vertex method for the solution of the two-dimensional unsteady Euler equations on a moving grid', *J. Aerosp. Eng.*, **209** (1995).
25. A.L. Gaitonde and D.P. Jones, 'Parallel implementation of a dual-time moving mesh method for the 2D unsteady Navier–Stokes equations', *Report No. 736*, University of Bristol, December 1995.

26. R.M. Gatiganti, K.J. Badcock and L. Dubuc, 'Solution to steady and unsteady incompressible Navier–Stokes equations with a free surface', *Euromech 374*, *Conf. on Recent Computational Developments in Steady and Unsteady Naval Hydrodynamics*, Poitiers, Futuroscope, 27–29 April 1998.
27. W.J. Gordon and C.A. Hall, 'Construction of curvilinear coordinate systems and applications of mesh generation', *Int. J. Numer. Methods Eng.*, **7**, 461–477 (1973).
28. R.H. Landon, 'NACA0012, oscillatory and transient pitching', in *Compendium of Unsteady Aerodynamic Measurements*, *AGARD-R-702*, 1982.
29. B.R. Williams, 'An exact test case for the plane potential flow about two adjacent lifting aerofoils', *ARC R&M 3717*, 1973.

