INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 23, 607–631 (1996)

AN UNFACTORED IMPLICIT MOVING MESH METHOD FOR THE TWO-DIMENSIONAL UNSTEADY N–S EQUATIONS

K. J. BADCOCK

Aerospace Engineering Department, University of Glasgow, Glasgow G12 8QQ, U.K.

AND

A. L. GAITONDE

Aerospace Engineering Department, University of Bristol, Bristol BS8 1TR, U.K.

SUMMARY

An unfactored implicit time-marching method for the solution of the unsteady two-dimensional Reynolds-averaged thin layer Navier–Stokes equations is presented. The linear system arising from each implicit step is solved by the conjugate gradient squared (CGS) method with preconditioning based on an ADI factorization. The time-marching procedure has been used with a fast transfinite interpolation method to regenerate the mesh at each time step in response to the motion of the aerofoil. The main test cases examined are from the AGARD aeroelastic configurations and involve aerofoils oscillating rigidly in pitch. These test cases have been used to investigate the effect of various parameters, such as CGS tolerance and laminar to turbulent transition location, on the accuracy and efficiency of the method. Comparisons with available experimental data have been made for these cases. In order to illustrate the application of the mesh generator and flow solver to more general flows where the aerofoil deforms, results for an NACA 0012 aerofoil with an oscillating trailing edge flap are also shown.

KEY WORDS: unsteady flow; implicit methods; mesh generation

1. INTRODUCTION

An important part of the design process for an aircraft is the demonstration that the flight envelope lies within the boundaries of aeroelastic stability. Owing to the cost of aeroelastic wind tunnel experiments, computational tools have an important role to play in this area. However, time-accurate simulation techniques for unsteady flow lag well behind the development of codes for steady flows, where convergence acceleration by multigrid, Newton iterations or a variety of other methods has reached an advanced stage and studies of three-dimensional flows governed by the Reynolds-averaged Navier–Stokes equations are being reported in the literature with increasing frequency.

The basis of most aeroelastic and many unsteady aerodynamic studies has been the low-frequency transonic small-disturbance equations. The code LTRAN, developed by Ballhaus and Goorjian,[1] which uses an alternating direction implicit (ADI) approach to solve the equations in multiple dimensions, provided a major breakthrough in computational aerodynamics.

One of the principal disadvantages of a code such as LTRAN is its inability to cope with strong shock waves. To model strong shocks satisfactorily, the lowest level of model required is the Euler equations. An early study of the unsteady Euler equations was published by Magnus and Yoshihara[2]

CCC 0271–2091/96/060607–25

Received August 1994
Revised January 1995

in 1975 based on an explicit algorithm. As computational methods have matured and computing facilities improved, studies of the unsteady Euler equations using both explicit and implicit methods have become increasingly common. Explicit methods are simple to code and vectorize, but for many problems the allowable time step for stability is much smaller than that required for accuracy. For unsteady flows, where most of the acceleration techniques developed to speed up steady flow calculations cannot be used (as they destroy time accuracy), this results in a requirement for a very large number of time steps. Studies based on explicit schemes include those published by Venkatakrishnan and Jameson,[3] Kandil and Chang[4] and Batina *et al.*[5] Implicit schemes allow much larger time steps, but the work required per time step may be large, particularly in three dimensions. Implicit studies include unfactored methods, e.g. those of Brenneis and Eberle,[6,7] and methods which use a Beam–Warming approximate factorization, e.g. those of Guruswamy[8] and Guruswamy and Obayashi.[9] More recently a dual-time approach has been developed by Jameson.[10] This uses an implicit real-time discretization, but at each real time step it marches the solution in pseudotime to a steady state through an explicit time-marching scheme. The acceleration techniques of steady flow calculations can be used, since the marching is in pseudotime.

The Navier–Stokes equations have remained relatively untouched for unsteady applications despite the need to model fully separated and mixed separated–attached flows. The grid density required to resolve the viscous flow features means that for explicit methods the stability limit leads to an excessively high number of time steps. Therefore most studies of the Navier–Stokes equations for unsteady flows published to date have used implicit methods. Notable calculations include the study of aileron buzz published by Steger and Bailey[11] and those by Levy and co-workers,[12,13] who studied shock-induced oscillations over a rigid aerofoil, and Chyu *et al.*[14] who examined the flow over a pitching NACA 64A010 aerofoil—a flow that has become a standard test case for Euler and Navier–Stokes codes. From 1990 Guruswamy and coworkers have reported results obtained with the three-dimensional Euler/Navier–Stokes code ENSAERO.[8,9,15–17] Applications include the aeroelastic flow over F-5 and delta wings. In 1990 an application of the Lockheed code ENS3D was reported to examine the flutter response of a wing–fuselage section. These studies are mostly based on the Beam–Warming approximate factorization method which utilizes central differences and artificial dissipation, although there is an indication that upwind methods are becoming increasingly popular. The dual-time approach has recently been extended to the Navier–Stokes equations by Arnone *et al.*,[18] who investigated vortex shedding, shock buffeting and wake effects on a rotor blade. This approach is likely to be competitive with traditional implicit methods.

There is still considerable scope for the development of new methods for the solution of the Navier–Stokes equations for a range of important unsteady flows, e.g. calculation of vortex-induced instabilities.[8] Owing to the intensive computing requirements for such flows, efficient numerical algorithms together with improved computing power are needed before calculations for three-dimensional geometries become routine. This paper describes current progress in the development of an implicit code capable of solving turbulent and aeroelastic flows. The flow solver is implemented in conjunction with a moving grid algorithm and is based on upwind methods and conjugate gradient solvers. Upwind schemes of the flux vector splitting or approximate Riemann solver type developed for the Euler equations led to very satisfactory ways of treating shock waves. When these methods are incorporated in Navier–Stokes codes, they allow the boundary layers to be well resolved owing to their low numerical dissipation. The large sparse linear system which arises for most steady and unsteady implicit methods must be solved efficiently. This is usually achieved by some form of approximate factorization which introduces an additional source of error into the solution procedure, with consequences for accuracy and efficiency. Despite this, the CFD manifestation of the alternating direction implicit (ADI) factorization, the Beam–Warming method, has proved very popular and successful with its incorporation in codes such as LTRAN and ARC3D. More recently an

approximate LU factorization based on flux vector splitting has been successfully used. Conjugate gradient methods provide an efficient method for solving the exact linear system to a required tolerance and they have been used successfully for steady flow problems. The preconditioning strategy is crucial to their success and in Reference 19 the ADI factorization was used as a preconditioner. This method, called AF-CGS (approximate factorization conjugate gradient squared), proved successful on a number of steady aerofoil test problems. A similar approach has been used in Reference 20 for the steady Euler equations and in Reference 21 for steady turbulent flows. Here the AF-CGS method is extended to unsteady turbulent flows.

The treatment of the mesh is a major consideration for unsteady aerodynamic and aeroelastic analysis, where the mesh must move to conform to the instantaneous body shape. Rigid body motions can be modelled without mesh regeneration or adaption by moving the mesh rigidly in response to the motion of the body. However, this approach is no longer an option if the body deforms as in an aeroelastic problem or if the outer boundaries of the mesh are fixed multiblock boundaries. An efficient transfinite interpolation method was used in Reference 22 to regenerate the mesh at each time step around an aerofoil with outer boundaries which were either fixed or undergoing a prescribed motion. The same grid generation procedure has been used in this study. In the work described here the speeds of all mesh points, which are required by the flow solver, have been calculated using a first-order finite difference procedure, whereas in Reference 22 the velocities of mesh points were generated by transfinite interpolation of the speeds of the mesh points on the aerofoil surface. The latter method was used for aerofoils either oscillating in pitch or with oscillating trailing edge flaps. For the pitching cases analytic expressions were derived for the speeds of the aerofoil mesh points, whereas for the oscillating flap cases the speeds of the mesh points on the aerofoil were approximated using a first-order finite difference procedure. For general aeroelastic problems, where no analytical description of the aerofoil motion would be available *a priori*, one of the two finite difference approaches described above would be required. This moving mesh method has already been extended to 3D flows about wings.[23]

Comparisons between results computed by the present method and experimental data are shown for pitching aerofoil flows, with the test cases being selected from the AGARD standard aeroelastic configurations. In addition, results from two oscillating flap problems are given and compared with solutions of the Euler equations calculated using the method described in Reference 23.

2. AF-CGS METHOD

The thin layer Reynolds-averaged Navier–Stokes equations in generalized curvilinear co-ordinates (ξ, η) with η normal to the surface are given in non-dimensional form by

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}}{\partial \xi} + \frac{\partial \mathbf{g}}{\partial \eta} = \frac{\partial \mathbf{s}}{\partial \eta}, \tag{1}$$

where

$$\mathrm{w} = J^{-1}\begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \qquad \mathrm{f} = J^{-1}\begin{bmatrix} \rho U \\ \rho u U + \xi_x p \\ \rho v U + \xi_y p \\ U(e+p) - \xi_t p \end{bmatrix}, \qquad \mathrm{g} = J^{-1}\begin{bmatrix} \rho V \\ \rho u V + \eta_x p \\ \rho v V + \eta_y p \\ V(e+p) - \eta_t p \end{bmatrix},$$

$$\mathbf{s} = Re^{-1}J^{-1}\begin{bmatrix} 0 \\ \mu m_1 u_\eta + (\mu/3)m_2\eta_x \\ \mu m_1 v_\eta + (\mu/3)m_2\eta_y \\ \mu m_1 m_3 + (\mu/3)m_2(\eta_x u + \eta_y v) \end{bmatrix},$$

with

$$\mu = \mu_L + \mu_T, \qquad U = \xi_t + \xi_x u + \xi_y v, \qquad V = \eta_t + \eta_x u + \eta_y v,$$
$$m_1 = \eta_x^2 + \eta_y^2, \qquad m_2 = \eta_x u_\eta + \eta_y v_\eta, \qquad m_3 = (u^2 + v^2)_\eta/2 + Pr^{-1}(\gamma - 1)^{-1}(c^2)_\eta$$

and J the determinant of the Jacobian of the transformation $x = x(\xi, \eta, t)$, $y = y(\xi, \eta, t)$. Here ρ, u, v, e, p, Re, c and γ denote density, the two components of velocity, energy, pressure, the Prandtl number, the Reynolds number, the speed of sound and the (constant) ratio of the specific heats respectively. The viscosity μ is composed of a part due to the natural viscosity of the fluid, μ_L, and a term to account for turbulence, μ_T. Sutherland's law is used to describe the variation in the fluid viscosity with temperature and the Baldwin–Lomax model[24] is used to provide a value for the turbulent viscosity. The transition point is generally fixed, based on either experimental practice or experimental observation. However, where the location of transition is not available from the experiment, calculations have been performed with various assumed transition points to assess the effect on the solutions.

Any accurate method for computing a flow should be able to calculate steady flow without the presence of any bodies. If moving meshes are used, then a geometric conservation law (GCL) is required to prevent mass, momentum or energy being produced unphysically by the numerical procedure.[25] This GCL has a similar integral form to the mass conservation law and leads to a set of equations which must be solved for the cell volumes using the same integration scheme as used for the flow equations themselves. However, several authors, including Guruswamy,[16] have found that for many test cases where the changes in cell areas at each time step are very small, the error in disregarding the GCL is less than other possible errors in the solution procedure. In this study the additional computational effort has been avoided as the GCL has not been applied. However, for applications of the present work this is a matter which requires further investigation.

To solve the system of partial differential equations (1), a finite volume scheme is used. Osher's method is used for the convective terms and the formulation for general geometries is given in Reference 26. A MUSCL interpolation is used to provide second- or third-order accuracy and the Von Albada limiter prevents spurious oscillations from occurring around shock waves. Central differencing is employed for the viscous terms. Far-field boundary conditions are imposed by Riemann invariants and no vortex correction is applied owing to the unsteadiness of the flow. One implicit step can be written as

$$\left[I + \Delta t\left(\frac{\partial \mathbf{R}_\xi}{\partial \mathbf{w}}\right)^n + \Delta t\left(\frac{\partial \mathbf{R}_\eta}{\partial \mathbf{w}}\right)^n\right]\Delta \mathbf{w}^n = -\Delta t(\mathbf{R}_\xi^n + \mathbf{R}_\eta^n), \tag{2}$$

where $\mathbf{R}_\xi$ and $\mathbf{R}_\eta$ are terms arising from the spatial discretization in the ξ- and η-direction respectively and

$$\frac{\partial \mathbf{f}}{\partial \xi} \approx \mathbf{R}_\xi, \qquad \frac{\partial(\mathbf{g} - \mathbf{s})}{\partial \eta} \approx \mathbf{R}_\eta, \qquad \Delta \mathbf{w} = \mathbf{w}^{n+1} - \mathbf{w}^n.$$

Here the turbulent viscosity is treated as being constant at its value at time level n in the calculation of $\partial \mathbf{R}_\eta / \partial \mathbf{w}$, because a linearization of this term would lead to a large number of additional terms in the coefficient matrix on the left-hand side of (2). This explicit treatment of the turbulent viscosity was not found to compromise the stability of the implicit time stepping. The ADI factorization of (2) is

$$\left[I + \Delta t\left(\frac{\partial \mathbf{R}_\xi}{\partial \mathbf{w}}\right)^n\right]\left[I + \Delta t\left(\frac{\partial \mathbf{R}_\eta}{\partial \mathbf{w}}\right)^n\right]\Delta \mathbf{w}^n = -\Delta t(\mathbf{R}_\xi^n + \mathbf{R}_\eta^n). \tag{3}$$

The ADI factorization given in (3) has been widely used because it is easier to solve than equation (2), as each of the factors is typically block diagonal or block pentadiagonal. However, the solution of the ADI system is not an exact solution of (2) and in practice the factorization error (the term neglected in using equation (3) rather than equation (2)) leads to a stability limit on the time step and introduces another source of error into the calculation.

An alternative approach is therefore adopted in the present work: equation (2) is solved to a required tolerance using a conjugate gradient method. The first major issue that arises from this approach is the construction of the matrix on the left-hand side of (2). Here the symbolic algebra package *Reduce*™ has been used to achieve a fully analytic evaluation of the matrix. Next a method is required to solve the resulting linear system which avoids factorization errors. This is achieved by using a conjugate gradient method to solve the exact system, with an efficient preconditioner. For the unsteady flows considered here the computation of the preconditioner must not be too computationally intensive, hence an efficient preconditioner based on an approximate factorization is used. This has been successfully applied for steady fluid flow problems.[27,28]

Conjugate gradient methods find an approximation to the solution of a linear system by minimizing a suitable residual error function in a finite-dimensional space of potential solution vectors. Several algorithms are available, including BiCG, CGSTAB, CGS and GMRES. These methods were tested in Reference 27 and it was concluded that the choice of method is not as crucial as the preconditioning. However, the CGS method was found to be the quickest of the three methods that do not require reorthogonalization and is used here. It has the additional advantage that the transpose of A, the matrix on the left-hand side of (2), is not required, which reduces implementation difficulties. The CGS algorithm was derived in Reference 29 and is restated in Reference 20.

Denoting the linear system to be solved at each time step by

$$A\mathbf{x} = \mathbf{b}, \tag{4}$$

we seek an approximation to $A^{-1} \approx C^{-1}$ which yields a system

$$C^{-1}A\mathbf{x} = C^{-1}\mathbf{b} \tag{5}$$

more amenable to conjugate gradient methods. The ADI method provides a fast way of calculating an approximate solution to (4) or, restating this, of forming the matrix vector product

$$C^{-1}\mathbf{b} = \mathbf{x}. \tag{6}$$

Hence, if we use the inverse of the ADI factorization as the preconditioner, then multiplying a vector by the preconditioner can be achieved simply by solving a linear system with the right-hand side given by the multiplicand and the left-hand side matrix given by the approximate factorization. The factors in C can be put in triangular form once at each time step, with the row operations being stored for use at each multiplication by the preconditioner.

The conditioning of the system is adversely affected by increasing the time step. Therefore there is a conflict between reducing the overall number of time steps used to solve the problem and reducing the number of CGS iterations required to solve the linear system at each time step. The behaviour of the CGS solver has been used in the present work to adapt the time step during a calculation. First, a maximum time step is set according to accuracy considerations. Experience of the number of steps per cycle required for calculations at various frequencies guides this choice of maximum time step. The time step is never allowed to increase above this threshold, but it might have to be reduced to enhance the CGS convergence. This is achieved by cutting off the CGS solution after a preset number of iterations if no convergence has been achieved, reducing the time step, recalculating the matrix on the left-hand side of the linear system and restarting the conjugate gradient solution with a linear system which should prove easier to solve. In contrast, if the linear system is solved easily at one step, then the time step is increased for the subsequent calculation, provided that it is kept lower than the maximum specified from an accuracy point of view. Finally, to increase the robustness of the computer code, the calculation is restarted with a lower CGS convergence criterion if negative density or pressure is detected.

3. MESH GENERATION

The solution of the thin layer Navier–Stokes equations on a moving grid requires a new grid along with the resulting grid speeds at each time step. The grid generator used in this study was developed in Reference 22 and is based on the algebraic technique of transfinite interpolation. The procedure effectively interpolates grid points in the computational domain from points on the inner boundary (aerofoil surface) and the outer (far-field or grid block) boundary. At each time step the transformation is given by a vector-valued function

$$F(\xi, \eta, t) = \begin{bmatrix} x(\xi, \eta, t) \\ y(\xi, \eta, t) \end{bmatrix}, \tag{7}$$

where ξ and η are the parametric co-ordinates of the grid and $0 \leqslant \xi \leqslant 1$, $0 \leqslant \eta \leqslant 1$. The grid is generated by partitioning the parametric co-ordinates into uniform intervals and mapping them to the physical domain using the transformation

$$F(\xi, \eta, t) = \alpha_1^0(\eta)F(\xi, 0, t) + \alpha_1^1(\eta)\frac{\partial F(\xi, 0, t)}{\partial \eta} + \alpha_2^0(\eta)F(\xi, 1, t) + \alpha_2^1(\eta)\frac{\partial F(\xi, 1, t)}{\partial \eta}, \tag{8}$$

where the functions $\alpha_i^j(\eta)$ are called blending functions. Here the blending functions described by Eriksson,[30] which give an exponential stretching between the boundary surfaces, are used. These blending functions do not always give sufficient control over the grid spacing. Additional control can be obtained by mapping to a set of intermediate control co-ordinates and then using transfinite interpolation from the control domain to the physical domain. In Reference 22, where grids were required for the solution of the Euler equations, the parametric co-ordinates were mapped to a control domain via $\eta_c = \eta^e$, where the exponent e is related to the mesh spacing on the aerofoil surface. This results in a normal mesh spacing which is a function of the arc length around the aerofoil and allows the mesh to be clustered in the region of the leading and trailing edges, resulting in meshes of good quality for the solution of the inviscid Euler equations. This control mapping is not suitable for the

thin layer Navier–Stokes equations, giving too large a stretching at the aerofoil for the problems considered in this paper. Therefore as an alternative the control mapping is taken to be the well-known hyperbolic tangent stretching function

$$\eta_c = P\eta + (1-P)\left(1 - \frac{\tanh[Q(1-\eta)]}{\tanh(Q)}\right), \tag{9}$$

with P and Q being user-specified constants. This gives satisfactory results for the test cases considered in this study.

In the current application the outer boundary is assumed to be fixed.

4. FLOW RESULTS

4.1. Definition of test cases

Results have been obtained for periodic flows. In each case a starting solution is obtained using a steady state version of the AF-CGS algorithm.[21] The aerofoil is then set in motion impulsively and the calculations are continued until a periodic solution is obtained.

The AGARD test cases[31] act as standard flows for computer code evaluation and verification. In this paper results are presented for several of these test cases for pitching aerofoils. The motion is defined by the angle of attack as a function of time and the centre of rotation, x_c, which is given as the distance along the chord as a percentage of the chord length c. The angle of attack is defined by

$$\alpha(t) = \alpha_m + \alpha_0 \sin(kt), \tag{10}$$

where t is the non-dimensional time, $k = \omega c/V_\infty$ and ω is the dimensional frequency.

The cases presented here are listed in Table I. The calculated solutions are compared with experimental data, which include detailed pressure distributions at a number of points during the pitching cycle as well as normal and moment coefficient values. Note that for case 4, experimental pressure data are only available for the upper surface of the aerofoil.

Table I. Pitching test cases examined

Number	Aerofoil	M_∞	$Re \times 10^6$	α_m	α_0	k	x_c	x_t
1	NACA 0012	0·60	4·8	2·89	2·41	0·1616	0·25	0·10
2	NACA 0012	0·60	4·8	4·86	2·44	0·1620	0·25	0·10
3	NACA 0012	0·755	5·5	0·016	2·51	0·1628	0·25	0·10
4	NACA 64A010	0·796	12·56	0	1·0	0·4080	0·248	0·05

Table II. Flap test cases examined

Number	Aerofoil	M_∞	$Re \times 10^6$	α	δ_0	k	x_t
5	NACA 0012	0·80	4·8	0·0	0·5	0·4000	0·10
6	NACA 0012	0·80	4·8	0·0	5·0	0·4000	0·10

The above cases 1–4 are all for rigid body motions. In order to demonstrate that the mesh generator can be applied to deforming aerofoils, theoretical test cases for an NACA 0012 aerofoil with an oscillating trailing edge flap have been considered as in Reference 22. The flap hinge lies on the upper surface of the aerofoil at $x/c = 0{\cdot}75$. The flap deflection angle δ is then given by

$$\delta = \delta_0 \frac{1 - \cos(kt)}{2}. \tag{11}$$

Two test cases were considered and these are listed in Table II.

4.2. Results

First consider results for the AGARD test cases. Comprehensive experimental data are available for these cases and they have therefore been widely used to check new unsteady methods.

For case 1 the qualitative behaviour of the shock wave which forms on the upward part of the motion* is well predicted; see Figure 1, where solutions on 77 × 30 and 137 × 30 meshes are shown. However, the shock strength is underpredicted and the location is upstream of experiment. The pressure distributions compare less favourably with experiment than the Euler results of References 3 and 22. The results for this case are very sensitive to the location of the transition point, see Figure 2, which suggests that the turbulence model is largely to blame for this. No improvement in the results was noted by mesh refinement in the streamwise or normal directions or by increasing the number of time steps taken per cycle, further suggesting that errors in the solution can be attributed to the turbulence model. The normal force coefficient agrees well with experimental data and the moment coefficient agrees well taken about 0·273 of the chord as in Reference 22 instead of the quarter-chord as quoted for the experimental results; see Figure 10.

Case 2 is well predicted, with good resolution of the shock wave; see Figure 4, where solutions are shown for 77 × 30 and 137 × 30 meshes. The shock strength is slightly underpredicted compared with experiment and is located slightly upstream of the experimental position, but the agreement is much closer than for case 1. No improvement in the results is noted with mesh refinement. The computed normal force coefficient variation is very close to that of the experimental values, but some disagreement is noted for the moment coefficient; see Figure 11. However, given the doubt over the exact location of the moment centre for case 1 (which is part of the same set of experiments) and the good prediction of the pressure distribution throughout the pitching cycle, the experimental values are in doubt.

For case 3, mesh refinement from 77 × 30 to 157 × 40 produces significantly improved shock resolution, with the pressure distribution on the finer mesh closely following the experimental behaviour; see Figure 5. The results are also consistent with the Navier–Stokes results given in Reference 32, which located the shock wave slightly upstream of experiment. The comparison of the moment and normal force coefficients in Figure 12 also follows that given in Reference 32. Poor agreement was noted therein for the viscous results compared with inviscid results and it was argued that the corrected angle of attack should be higher and that when the inviscid solutions located the shock wave downstream of the actual position for the computational conditions, this is closer to the position for the actual experimental conditions. It was also noted that the phase of the inviscid and viscous calculations was identical and hence that inviscid results would be sufficient for aeroelastic calculations at similar flow conditions.

* In this subsection the aerofoil motion is described in terms of the movement of the nose, i.e. the upward motion denotes the case where α is increasing.

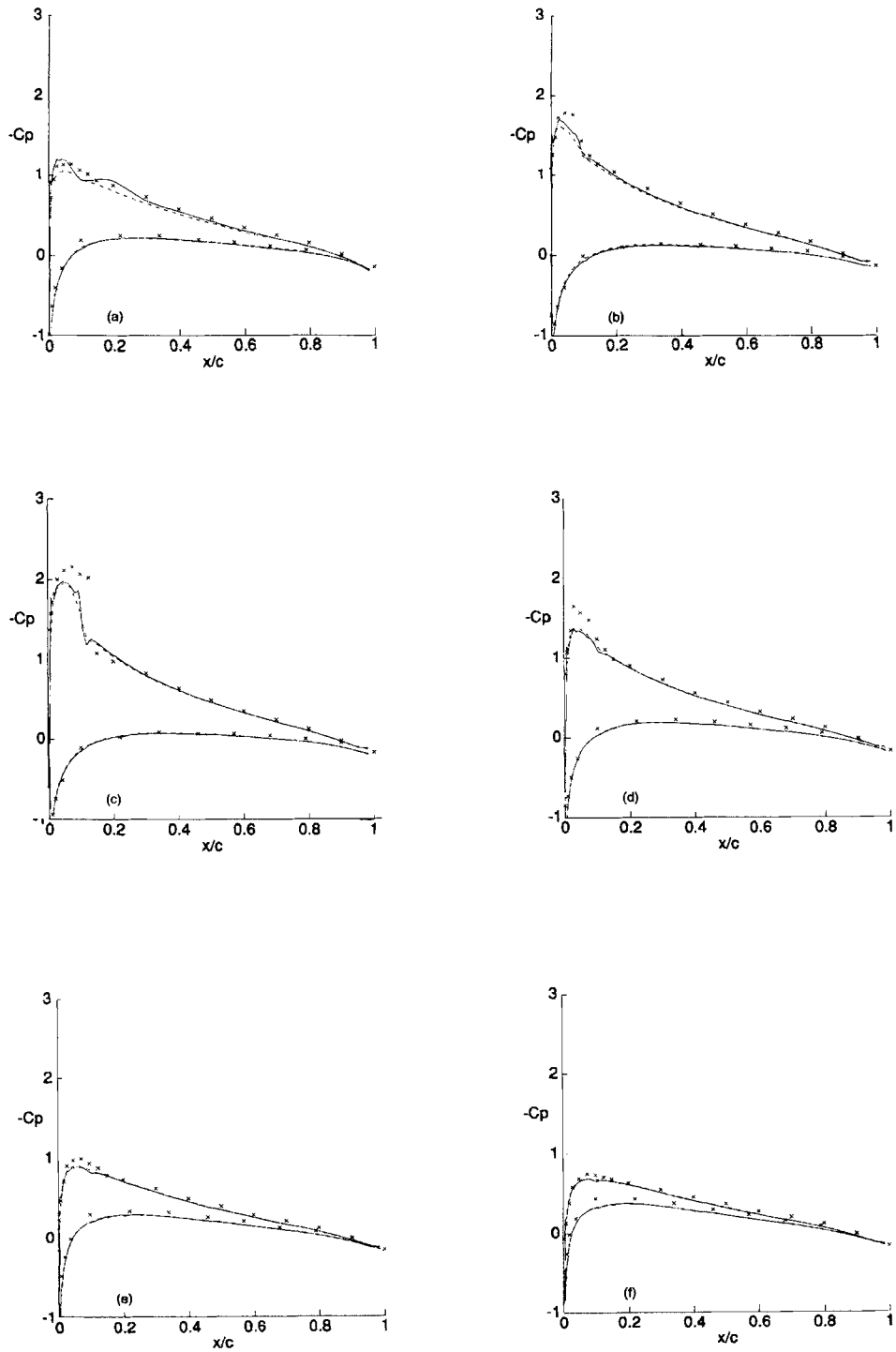

Figure 1. $-C_p$ versus x/c for case 1 with no limiter and transition at 0·1: × , experiment; ———, mesh 77 × 30; ———, mesh 137 × 30; α (deg) = (a) 2·97, (b) 4·56, (c) 5·09, (d) 2·62, (e) 1·16, (f) 1·29

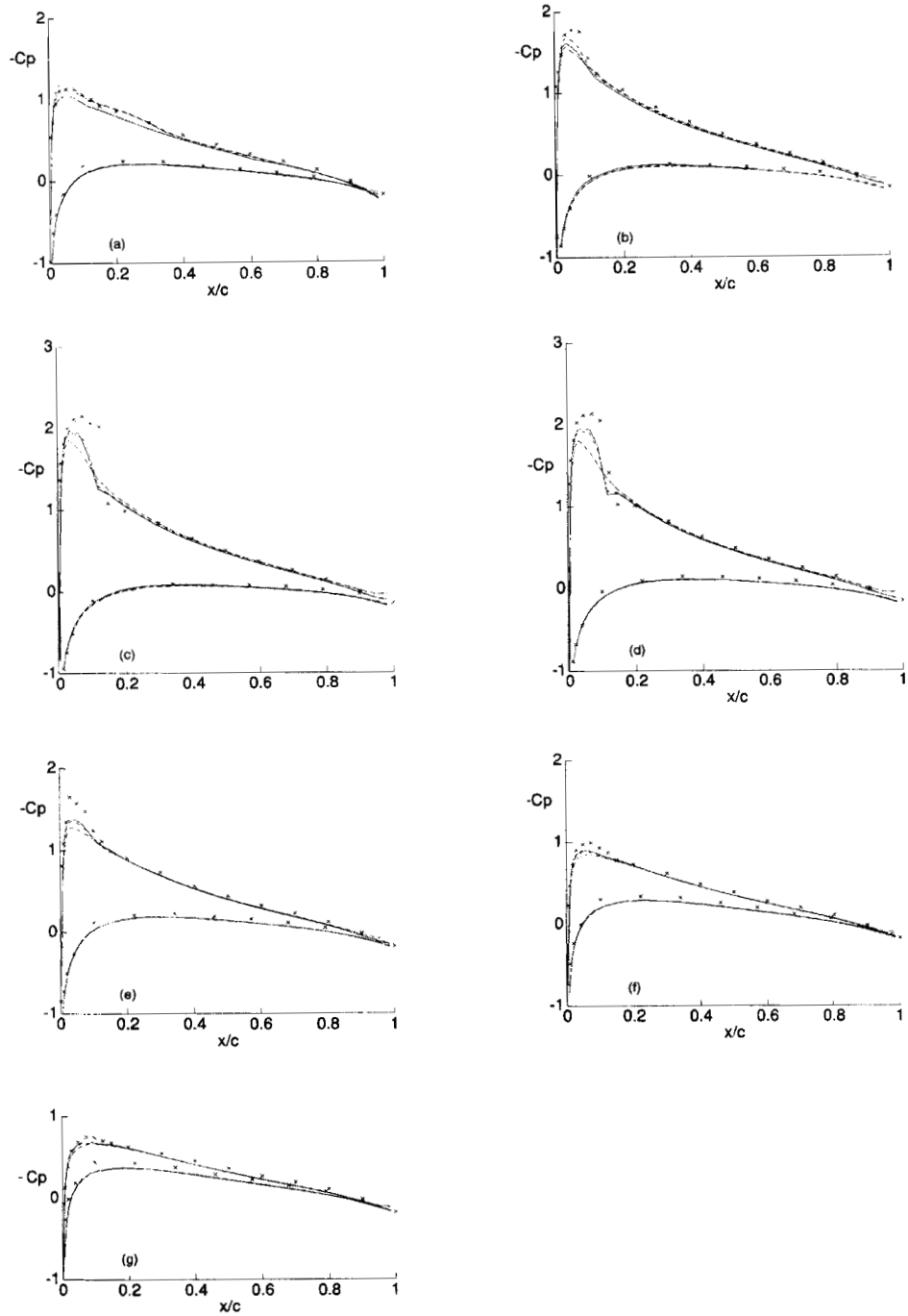

Figure 2. $-C_p$ versus x/c for case 1 with no limiter: × , experiment; transition at (————) 0·15, (— — —) 0·10, (– · –) 0·05; α (deg) = (a) 2·97, (b) 4·56, (c) 5·09, (d) 4·17, (e) 2·62, (f) 1·16, (g) 1·29

Mesh refinement from 77 × 30 to 157 × 40 again gave closer agreement with experiment for case 4, with the shock wave resolution again being significantly improved; see Figure 6. Close agreement with experiment for the normal force is noted, see Figure 13, although a periodic solution has yet to be established. Similar discrepancies with experiment for the moment coefficient are observed as for the Euler solutions in References 3 and 22, suggesting that the integration of the pressure values is in error for the experimental results.

The thin layer Navier–Stokes code has produced good results for the above AGARD test cases. However, they represent mainly attached flows and therefore do not represent a serious challenge for the thin layer Navier–Stokes code.

In addition, the above test cases could all have been treated by rigidly rotating the mesh with the aerofoil. To demonstrate the generality of the mesh regeneration approach, a theoretical problem of an NACA 0012 aerofoil with oscillating trailing edge flap was considered. The results from the thin layer Navier–Stokes solver were compared with those available from a Euler solver to assess viscous effects. The Euler solver was a dual-time method with four-stage Runge–Kutta time stepping.[23]

Test case 5 has a small maximum flap deflection angle of $\delta_0 = 0{\cdot}5°$. The instantaneous pressure distributions from the Euler (109 × 30 grid) and thin layer Navier–Stokes (129 × 40 grid) calculations at $\delta = 0°$, $0{\cdot}25°\uparrow$, $0{\cdot}5°$ and $0{\cdot}25°\downarrow$ during the third cycle of motion are shown in Figure 7. The distributions are essentially very similar, with just a small difference in shock location, indicating that viscous effects are small and that the flow is attached. This is confirmed by examination of the flow field. Mesh refinement did not produce any change in the results.

Test case 6 has a much larger maximum flap deflection angle $\delta_0 = 5°$. Thin layer Navier–Stokes calculations were performed on 129 × 40 and 175 × 60 meshes. Mesh refinement did produce some differences in the pressure distributions. The fine grid distributions are shown in Figure 8 for $\delta = 0°$, $2{\cdot}5°\uparrow$, $5°$ and $2{\cdot}5°\downarrow$, again on the third cycle of motion. Euler solutions are also shown for a 129 × 30 grid and in this case the pressure distributions exhibit differences from the thin layer Navier–Stokes solutions. The flow is dominated by a strong shock wave that develops on the lower surface of the aerofoil in the vicinity of the corner formed by the flap. When the flap is undeflected, the shock lies upstream of the 'flap corner', and as the flap deflects upwards, the shock decreases in strength and moves towards the leading edge. As the flap returns to its undeflected position, the shock strength increases, and at $\delta = 2{\cdot}5°$ the shock has moved far enough back to lie just further aft than the 'flap corner'. Examination of the shear stress (non-dimensionalized by $\rho_\infty V_\infty^2$) shows evidence of flow separation; see Figure 9. For $\delta = 0°$, $2{\cdot}5°\uparrow$ and $5°$ the shear stress indicates only a small region of separation, whereas the $\delta = 2{\cdot}5°\downarrow$ shear stress indicates that the flow over most of the lower surface of the flap is separated. Note that this is predicted by both the 129 × 40 and 175 × 60 meshes. This is confirmed by examination of velocity vectors in cells adjacent to the body surface. The reversed flow is captured over between eight and 16 cells in the η-direction on the 175 × 60 grid.

This last test case does exhibit viscous effects, as confirmed by comparisons with a Euler solution. However, the region of separation is very small. Further tests are required to establish the ability of the code to deal with large-scale separation, with particular attention paid to the turbulence model. As yet it has not been possible to determine a suitable test case which exhibits large-scale separation and for which experimental data are available.

5. EFFECT OF NUMERICAL PARAMETERS

The main numerical parameters of the method are the time step, the limiter details, the CGS tolerance, the transition point, the precision with which the computer code is implemented and the mesh density. The time step affects the efficiency of the CGS solution as well as the accuracy of

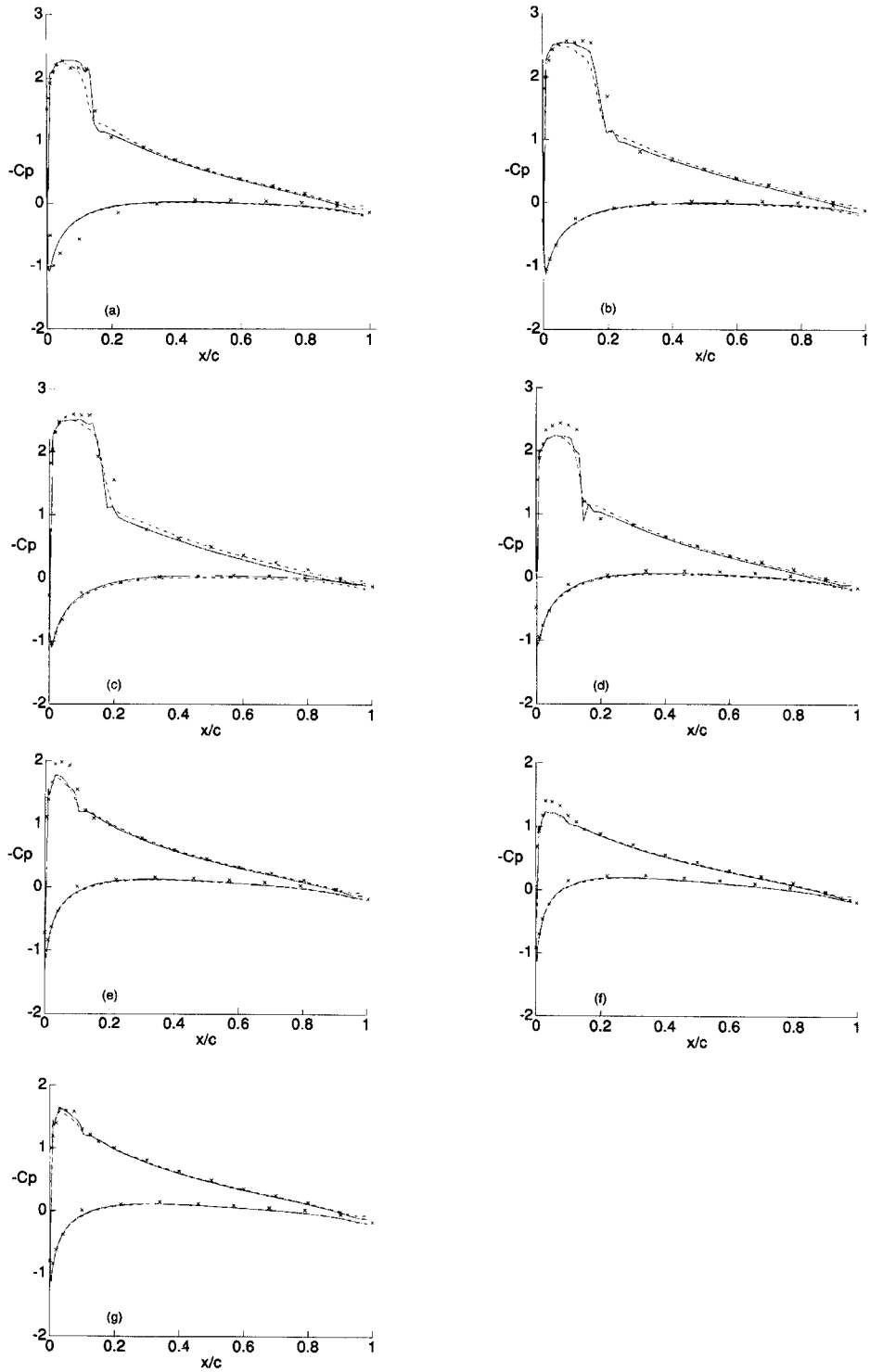

Figure 3. $-C_p$ versus x/c for case 2: ×, experiment; ————, no limiter; — — —, Von Albada limiter; α (deg) = (a) 5·94, (b) 6·96, (c) 6·59, (d) 5·12, (e) 3·51, (f) 2·44, (g) 4·27

the flow solution as discussed in Section 2. In general, the smaller the time step, the easier the linear system is to solve at each time step at the cost of an increased number of steps in the thin layer Navier–Stokes solution. For the results of this paper the maximum number of CGS steps was set at 20. The CGS residual value at which the time step is increased if this has been achieved after one iteration was set at one-tenth of the CGS tolerance. The number of steps per cycle is either increased or decreased by 50 according to whether or not either of these conditions has been met.

The number of time steps required to compute one cycle of each of the AGARD test cases is shown in Table III for various meshes and for single- and double-precision versions of the code. Case 2 proved the most difficult for the CGS solver, with stronger shock waves present than for the other cases. Case 2 was also solved using no flux limiter and the results are shown in Figure 3. It is clear from the figure that the solution obtained with no limiter is very close to that obtained by using one, except for small oscillations around the shock wave. It is also clear from Table III that the unlimited case is much easier to solve, requiring fewer time steps per cycle. However, from the point of view of eliminating spurious solution modes, the limiter should be used where this does not lead to unacceptably high run times. In Reference 32, 15,000 time steps were required to solve one cycle of case 3, and in Reference 33, 2600 steps per cycle were needed for case 4, both studies used the Beam–Warming factorization. This is an order of magnitude greater than the number required in the present work. The most costly part of the present method is the Jacobian calculation and an implicit step using an ADI solution is only 15 per cent cheaper than one based on CGS. Hence the present results would represent a substantial reduction in CPU time based on the quoted number of time steps given above. A time step refinement study for case 4 showed that the results obtained for the number of steps per cycle is shown in Table III have converged with respect to the time step. This indicates that the time step is still restricted by some factor other than accuracy, in this case CGS convergence. However, the restriction is much less severe than for ADI.

The location of the transition point was found to have a significant effect on the results for case 1. Transition was free in the experiments, but the location for the computation has to be fixed as part of the simulation. Three different points were tried at 0·05, 0·1 and 0·15 and the results are shown in Figure 2. It is clear that fixing the point at 0·1 or 0·15 gives much better shock wave resolution compared with the case where transition is at 0·05. However, there is no rational justification for fixing the transition point in this manner and this is one of several weaknesses of the Baldwin–Lomax model which is a problem common to all turbulence models. The results for the other cases were not found to be as sensitive to the location of transition as case 1 is.

The effect of the precision of arithmetic on conjugate gradient methods is well known. Rounding errors can significantly reduce the convergence rate. This is seen in Table III, where the number of steps required by the double-precision version is always less than or equal to the number required by the single-precision case, since the double-precision code can sometimes solve the linear system in a fixed number of steps when no convergence is achieved for the single-precision case. However, the double-precision code requires more memory and CPU time per step and in general the decrease in the number of time steps is not great enough to offset these disadvantages. Nevertheless, for particularly complicated flows the double-precision version might prove significantly more efficient.

The CGS tolerance is a crucial parameter since it determines the stability of the time-stepping scheme. For all the cases considered the tolerance was set at one-tenth of the residual corresponding to the solution obtained by approximate factorization. The only cases which subsequently suffered negative density or pressure were cases 1 and 2 and these were successfully solved after the solution was reset and the tolerance was halved to one-twentieth. The automatic resetting procedure provides a convenient way to quickly decide on a suitable CGS tolerance for an unfamiliar class of problems and adds to the robustness of the code.

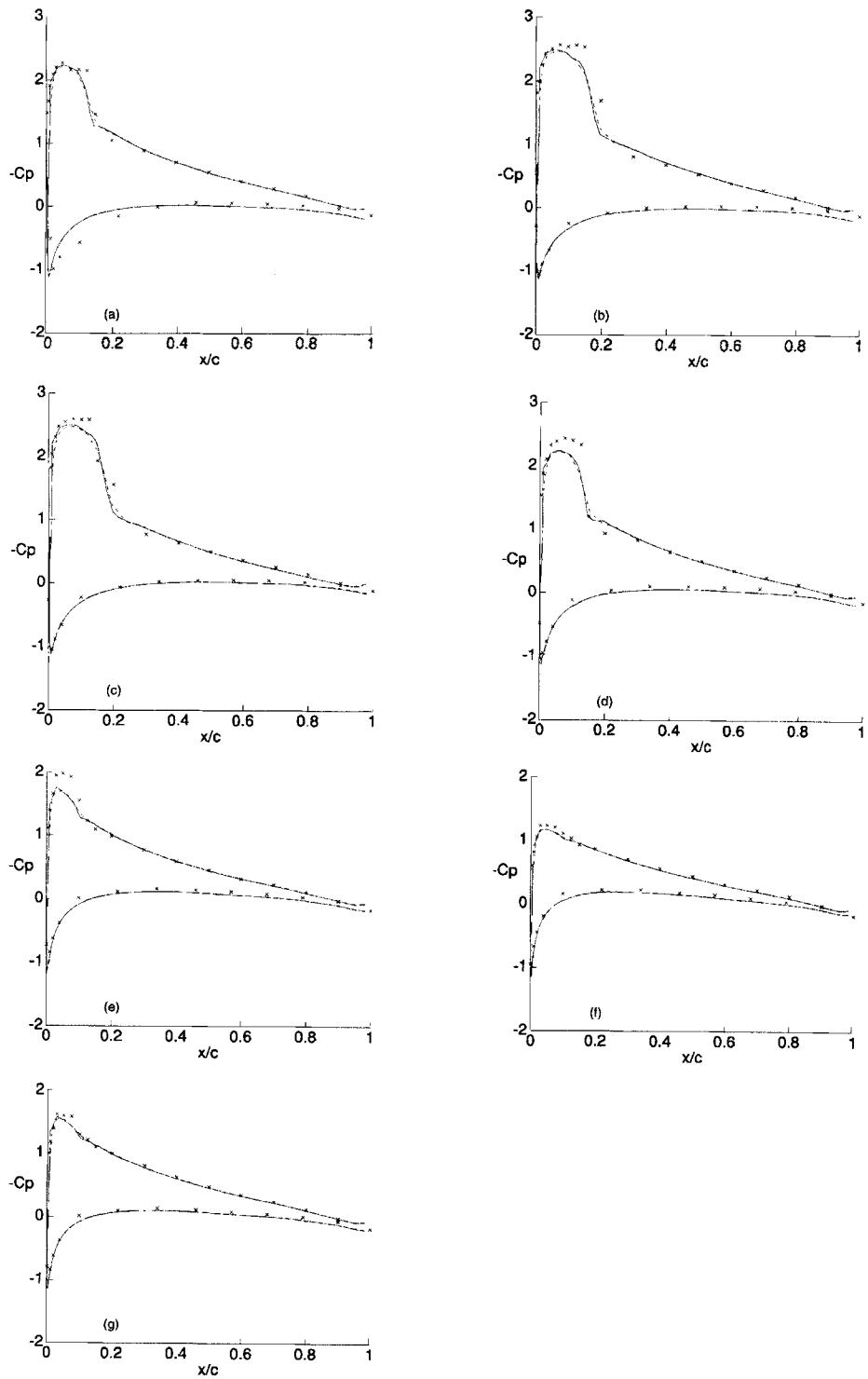

Figure 4. $-C_p$ versus x/c for case 2: ×, experiment; — — —, mesh 77 × 30; ———, mesh 137 × 30; α (deg) = (a) 5·94, (b) 6·96, (c) 6·59, (d) 5·12, (e) 3·51, (f) 2·67, (g) 4·27

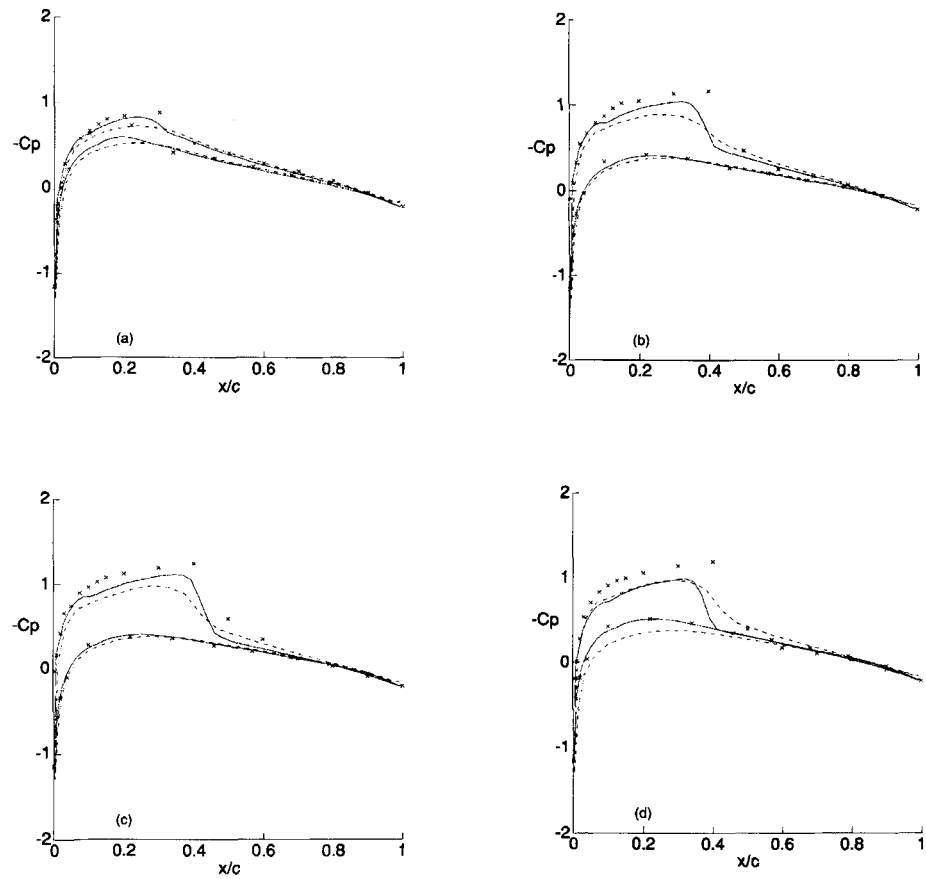

Figure 5. $-C_p$ versus x/c for case 3: ×, experiment; — — —, mesh 77 × 30; ———, mesh 157 × 40; α (deg) = (a) 1·07, (b) 2·34, (c) 2·02, (d) 0·53, (e) −1·24, (f) −2·41, (g) −2·01, (h) −0·56

Finally, we examine the effect of mesh refinement. The performance of conjugate gradient methods in general is degraded as the size of the system increases. This feature is clear from the results in Table III, where it is shown that smaller time steps are required on finer meshes for the linear solver to converge within the specified number of iterations. This feature is undesirable, but the degradation is not prohibitively large for the test cases examined herein.

6. CONCLUSIONS

An unfactored implicit method for the Reynolds-averaged thin layer Navier–Stokes equations on moving meshes has been developed. Provisional comparisons of the method with approximately factored methods, which are widely used for viscous flow problems, are favourable. The ability to control, through the conjugate gradient tolerance, the error incurred in the solution of the linear

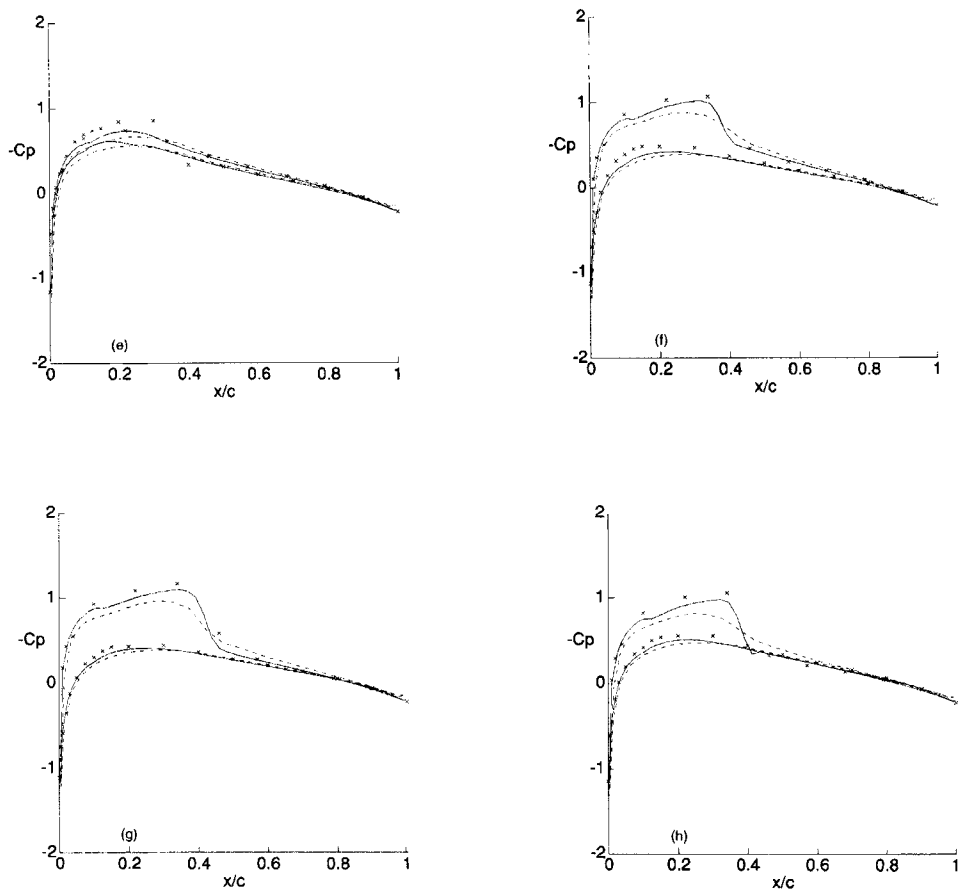

Figure 5. (*continued*)

system at each time step allows larger time steps to be taken compared with approximately factored methods. Further work is required to evaluate the AF-CGS method against a well-developed approximate factorization method.

The incorporation of a general moving mesh method based on transfinite interpolation has improved the generality of the unsteady AF-CGS code. The outer boundary of the grid pursues a motion which is independent of the component it surrounds (here the outer boundary is fixed), which provides several benefits. First, the method is applicable to aerolastic calculations, whereas previous versions of AF-CGS were restricted to rigid body flow calculations. Second, the present scheme could be incorporated easily into a multiblock method, which is popular for computing flows about complex configurations. The transfinite interpolation procedure was initially developed for Euler calculations and good interpolation functions and grid parameters have been identified for this application. This procedure was adapted for this study to generate grids for the Navier–Stokes calculations by using the well-known hyperbolic tangent stretching function and more work is required to assess and optimize the grid quality for viscous applications. There is also scope for the

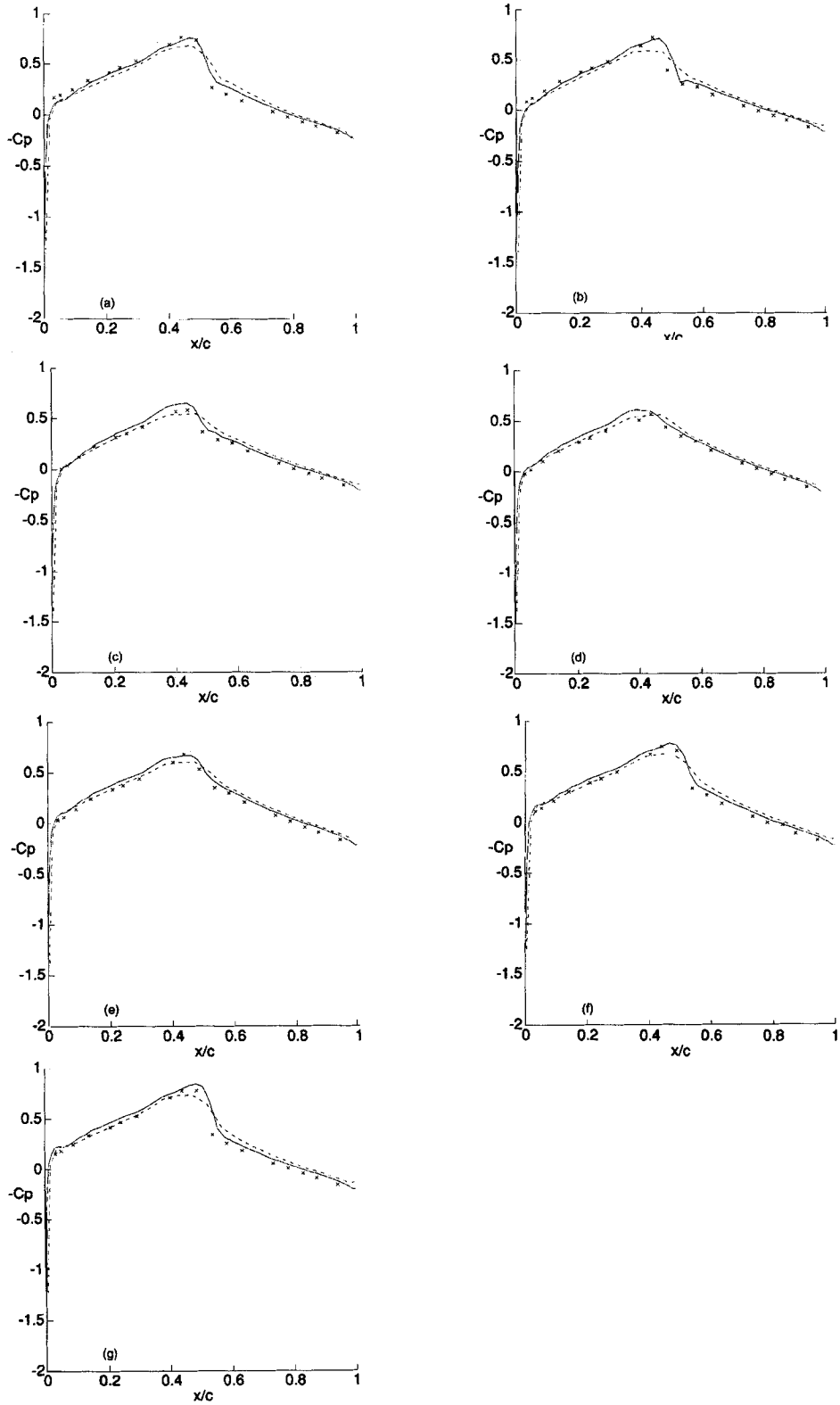

Figure 6. $-C_p$ versus x/c for case 4: ×, experiment; — — —, mesh 77 × 30; ———, mesh 157 × 40; α (deg) = (a) 0·21, (b) 0·87, (c) 1·00, (d) 0, (e) 0·73, (f) −1·00, (g) −0·59

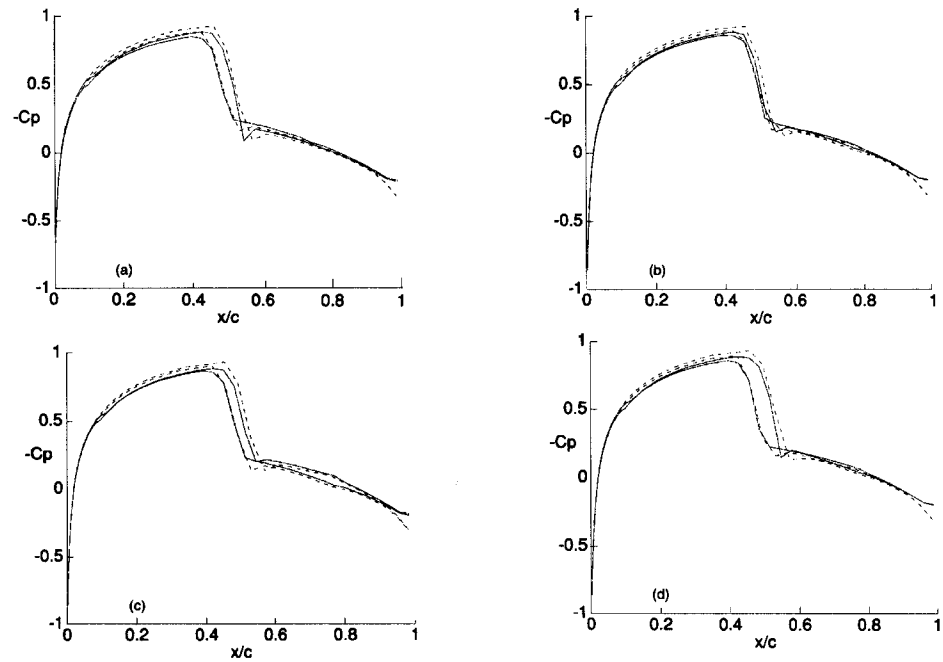

Figure 7. $-C_p$ versus x/c for case 5: ———, Euler; ————, thin layer Navier–Stokes; δ (deg) = (a) 0, (b) 0·25, (c) 0·5, (d) 0·25

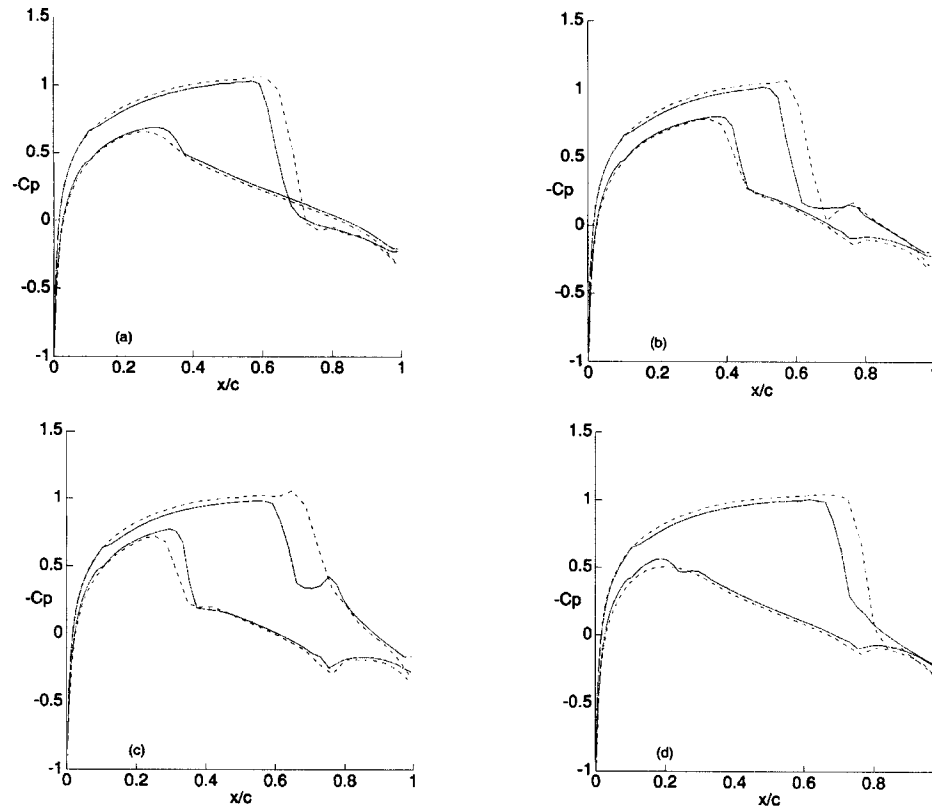

Figure 8. $-C_p$ versus x/c for case 6: ———, Euler; ————, thin layer Navier–Stokes; δ (deg) = (a) 0, (b) 2·5, (c) 5, (d) 2·5

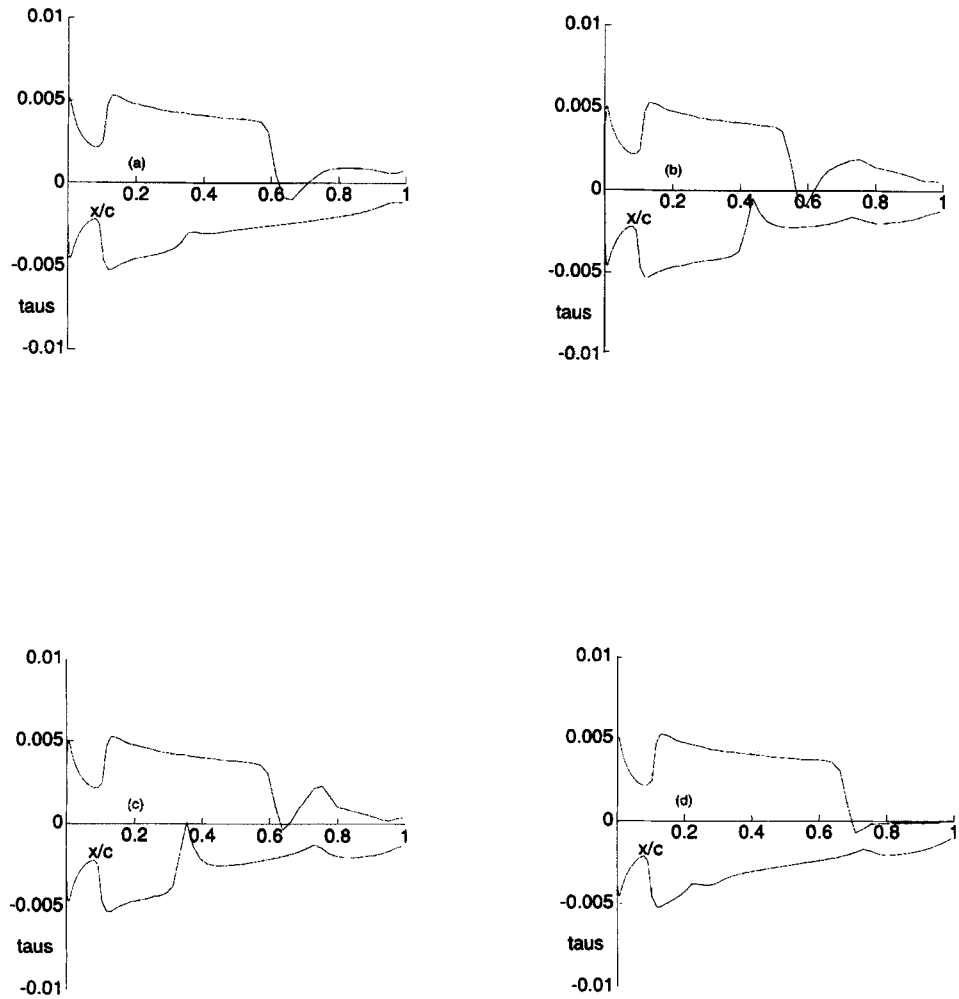

Figure 9. Shear stress τ_s versus x/c for case 6: δ (deg) = (a) 0, (b) 2·5, (c) 5, (d) 2·5

development of a solution-adaptive version of the grid generation scheme, which might enhance the current method.

The method was verified for several AGARD test cases; satisfactory agreement was noted with both experiment and previous computations. These cases represent attached flow and so are not in general representative of the flow problems which need to be studied by solving the thin layer Navier–Stokes equations. The oscillating flap test case with $\delta_0 = 5°$ shows evidence of a very small region of separated flow and has significantly different pressure distributions from those of a Euler code, which indicates that viscous effects play a role in this case. It is anticipated that in cases with more significant regions of separated flow the main problems will be associated with the turbulence model and not with the solution algorithm.

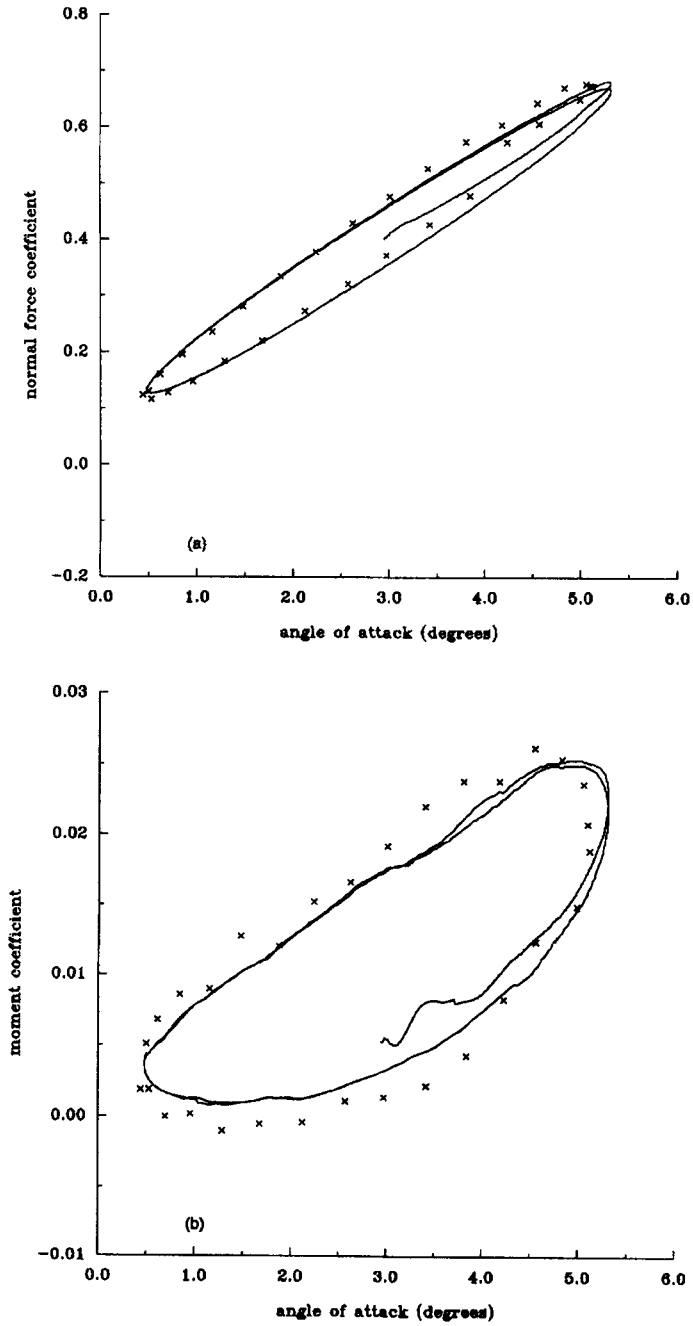

Figure 10. Integrated (a) normal force and (b) moment coefficients as a function of angle of attack for case 1: ×, experiment; ———, computed

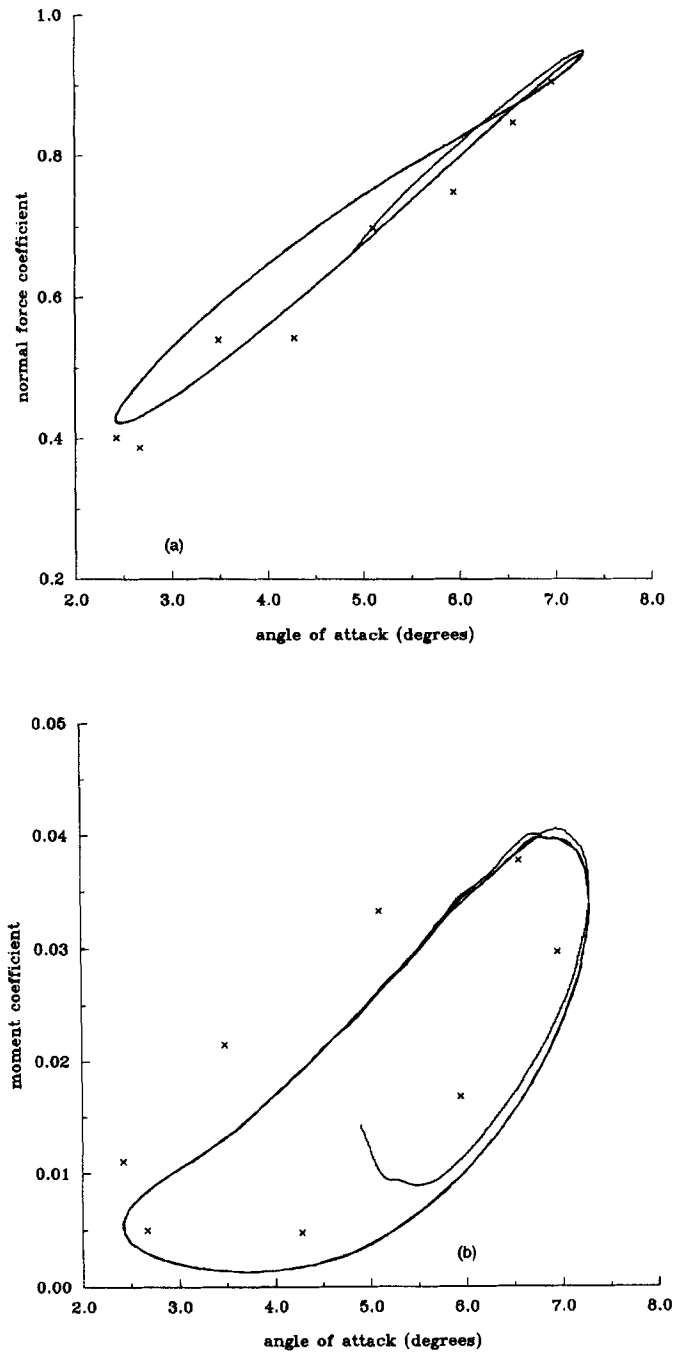

Figure 11. Integrated (a) normal force and (b) moment coefficients as a function of angle of attack for case 2: ×, experiment; ————, computed

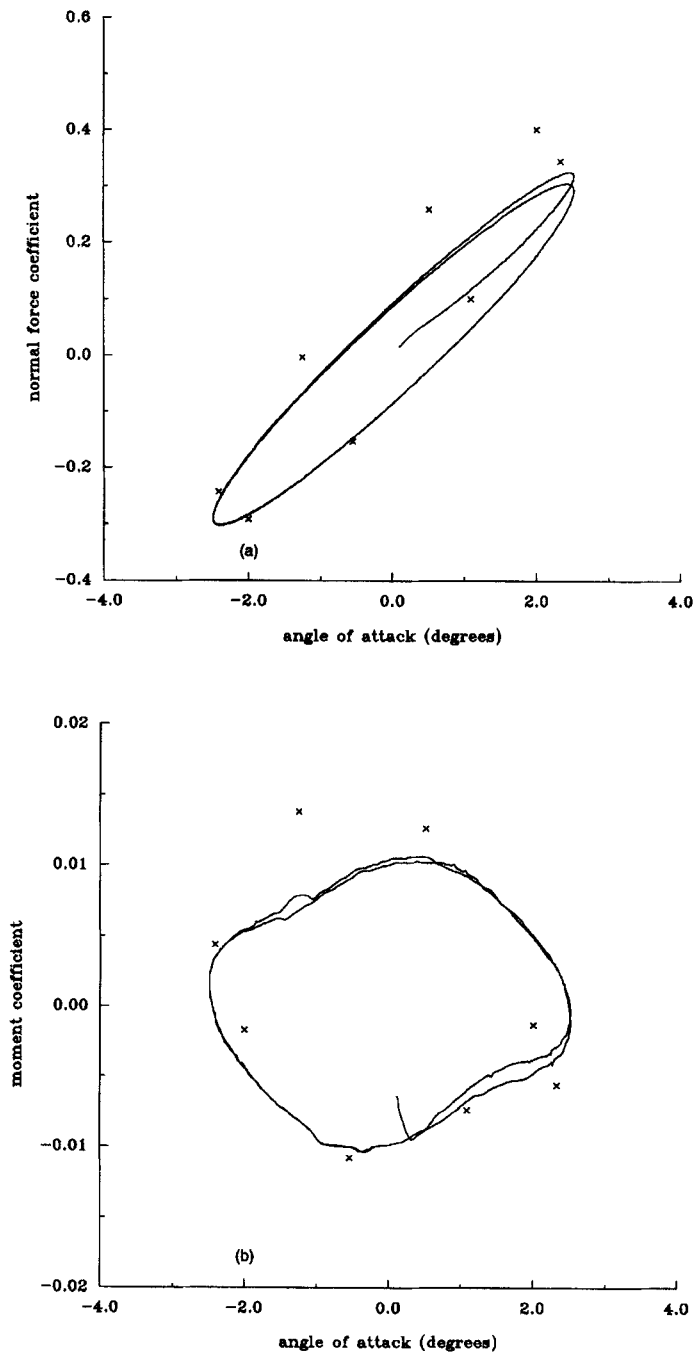

Figure 12. Integrated (a) normal force and (b) moment coefficients as a function of angle of attack for case 3: ×, experiment; ————, computed

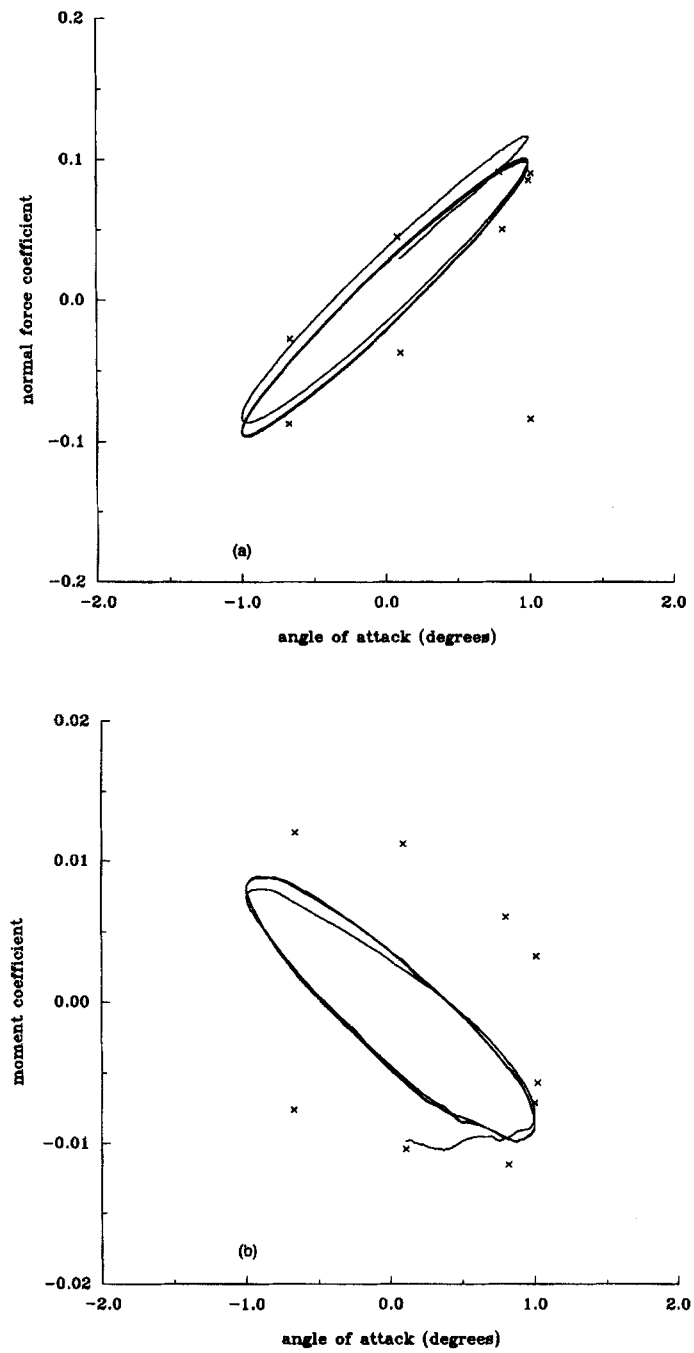

Figure 13. Integrated (a) normal force and (b) moment coefficients as a function of angle of attack for case 4: ×, experiment; ————, computed

Table III. Number of time steps required to compute one cycle

Case	Mesh	Limiter	CGS tolerance	Single precision	Double precision
1	77 × 30	No	0·05	796	—
	137 × 30	No	0·05	1208	—
	77 × 60	No	0·05	1094	—
2	77 × 30	Yes	0·05	1980	1334
	77 × 30	No	0·05	1178	922
	137 × 30	Yes	0·05	4660	2345
	137 × 30	No	0·05	1713	1299
3	77 × 30	Yes	0·1	569	464
	157 × 40	Yes	0·1	791	656
4	77 × 30	Yes	0·1	178	150
	157 × 40	Yes	0·1	418	320

Work is currently in progress to couple the flow code to a structural model to study the aeroelastic response of an aerofoil. The extension of the method to 3D wing calculations is currently underway. The straight generalization of the method is unpromising owing to the poor quality of the ADI factorization and to the large memory requirements in 3D. To overcome these problems, a two-factor method has been developed which has one block diagonal factor and one factor of a more complicated sparsity pattern which is solved by the approach described in this paper. Preliminary results have been promising.

ACKNOWLEDGEMENTS

K.J.B. is funded under SERC/MOD contract GR H 47371. A.L.G. is a Lloyds of London Tercentenary Foundation Research Fellow.

REFERENCES

1. W. F. Ballhaus and P. M. Goorjian, 'Implicit finite-difference computations of unsteady transonic flows about aerofoils', *AIAA J.*, **15**, 1728–1735 (1977).
2. R. Magnus and H. Yoshihara, 'Unsteady transonic flows over an aerofoil', *AIAA J.*, **13**, 1622–1628 (1975).
3. V. Venkatakrishnan and A. Jameson, 'Computation of unsteady transonic flows by the solution of the Euler equations', *AIAA J.*, **26**, 974–981 (1988).
4. O. A. Kandil and H. A. Chang, 'Computation of steady and unsteady vortex-dominated flows with shock waves', *AIAA J.*, **26**, 524– 531 (1988).
5. J. T. Batina, E. M. Lee, W. L. Kleb and R. D. Rausch, 'Unstructured- grid methods development for unsteady aerodynamic and aeroelastic analyses', in *Transonic Unsteady Aerodynamics and Aeroelasticity*, AGARD, 1991.
6. A. Brenneis and A. Eberle, 'Evaluation of an unsteady implicit Euler code against two and three-dimensional standard configurations', in *Transonic Unsteady Aerodynamics and Aeroelasticity*, AGARD, 1991. pp 10.1–10.150.
7. A. Brenneis and A. Eberle, 'Application of an implicit relaxation method solving the Euler equations for time-accurate unsteady problems', *J. Fluids Eng.*, **112**, 510–520 (1990).
8. G. P. Guruswamy, 'Unsteady aerodynamic and aeroelastic calculations for wings using Euler equations', *AIAA J.*, **28**, 461–469 (1990).
9. G. P. Guruswamy and S. Obayashi, 'Transonic aeroelastic computations on wings using Navier–Stokes equations', in *Transonic Unsteady Aerodynamics and Aeroelasticity*, AGARD, 1991.
10. A. Jameson, 'Time dependent calculations using multigrid with applications to unsteady flows past aerofoils and wings', *AIAA Paper 91-1596*, 1991.
11. J. L. Steger and H. E. Bailey, 'Calculation of transonic aileron buzz', *AIAA J.*, **18**, 249–255 (1980).
12. L. L. Levy, 'Experimental and computational steady and unsteady transonic flows about a thick aerofoil', *AIAA J.*, **16**, 564–572 (1978).

13. J. B. McDevitt, L. L. Levy and G. S. Deiwert, 'Transonic flow about a thick circular-arc aerofoil', *AIAA J.*, **14**, 606–613 (1976).
14. W. J. Chyu, S. S. Davis and K. S. Chang, 'Calculation of unsteady transonic flow over an aerofoil', *AIAA J.*, **19**, 684–690 (1981).
15. N. M. Chaderjian and G. P. Guruswamy, 'Transonic Navier–Stokes computations for an oscillating wing using zonal grids', *J. Aircraft*, **29**, 326–335 (1992).
16. G. P. Guruswamy, 'Navier–Stokes computations on swept-tapered wings, including flexibility', *J. Aircraft*, **29**, 588–597 (1992).
17. S. Obayashi, G. P. Guruswamy and P. M. Goorjian, 'Streamwise upwind algorithm for computing unsteady transonic flows past oscillating wings', *AIAA J.*, **29**, 1668–1677 (1991).
18. A. Arnone, M.-S. Liou and L. A. Povinelli, 'Multigrid time-accurate integration of Navier–Stokes equations', *AIAA Paper 93-3361*, 1993.
19. K. J. Badcock, 'An efficient unfactored implicit method for unsteady aerofoil flows', *GU Aero Rep. 9313*, 1993.
20. M. Vitaletti, 'Solver for unfactored schemes', *AIAA J.*, **29**, 1003–1005 (1991).
21. K. J. Badcock, I. C. Glover and B. E. Richards, 'A preconditioner for steady two-dimensional turbulent flow simulation', *Int. J. Numer. Methods Heat Transfer Fluid Flow*, in press (1996). Vol 6 pp 79–93.
22. A. L. Gaitonde and S. P. Fiddes, 'A moving mesh system for the calculation of unsteady flows', *AIAA Paper 93-0641*, 1993.
23. A. L. Gaitonde, 'A dual-time method for the solution of the unsteady Euler equations', *Aeronaut. J.*, **98**, 283–291 (1994).
24. B. S. Baldwin and H. Lomax, 'Thin layer approximation and algebraic model for separated turbulent flow', *AIAA Paper 78-257*, 1978.
25. 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).
26. S. Osher and S. R. Chakravarthy, 'Upwind schemes and boundary conditions with applications to Euler equations in general coordinates', *J. Comput. Phys.*, **50**, 447–481 (1983).
27. K. J. Badcock, 'Newton's method for laminar aerofoil flows', *GU Aero Rep. 9310*, 1993.
28. V. Venkatakrishnan, 'Preconditioned conjugate gradient methods for the compressible Navier–Stokes equations', *AIAA J.*, **29**, 1092–1100 (1990).
29. P. Sonneveld, 'CGS: a fast Lanczos-type solver for nonsymmetric linear systems', *SIAM J. Stat. Comput.*, **10**, 36–52 (1989).
30. I.-E. Eriksson, 'Generation of boundary-conforming grids around wing- -body configurations using transfinite interpolation', *AIAA J.*, **20**, 1313–1320 (1982).
31. AGARD, 'Compendium of unsteady aerodynamic measurements', *Tech. Rep. 702*, 1982.
32. M. P. Thomadakis and S. Tsangaris, 'On the prediction of transonic unsteady flows using second order time accuracy', in *Computational Fluid Dynamics '92*, 1992. Elsevier pp 711–718 edited C Hirzch et al.
33. W. J. Chyu and S. S. Davis, 'Numerical studies of unsteady transonic flow over an oscillating aerofoil', in *Transonic Unsteady Aerodynamics and Its Aeroelastic Applications*, AGARD, 1984. pp 3.1–3.21.