AIAA JOURNAL
Vol. 34, No. 3, March 1996

Implicit Time-Stepping Methods for the Navier–Stokes Equations

K. J. Badcock* and B. E. Richards†
University of Glasgow, Glasgow G12 8QQ, Scotland, United Kingdom

A novel two-factor approximate factorization for three-dimensional flows is described. The method uses a preconditioned conjugate gradient solver for one of the factors, and this solution is based on a method developed for two-dimensional flows. This two-dimensional method is described, and its performance is summarized for steady and unsteady flows. The two-factor method is then given. Larger time steps can be used with this method than is possible with the more conventional three-factor method. Test results for the transonic steady flow over the ONERA M6 wing are given.

I. Introduction

A FUNDAMENTAL problem of aerospace engineering is to determine the motion of a vehicle released into a flow. For some flow conditions, the fluid–structure interaction can have terminal consequences for the flight vehicle, and hence this interaction must be investigated in detail during the design stage. Computational tools have a major role to play in this process due to the expense of flutter wind-tunnel tests.

A hierarchy of flow models exists that can describe increasingly complicated flow features. For general flutter calculations, the capability to model shock waves and flow separation is important because of the influence these features can have on the structural response. Examples include the flutter dip in the transonic regime and vortex-induced instabilities. Therefore, the appropriate level of modeling for general flutter calculations is the Navier–Stokes equations. However, the computational cost of solving these equations at present precludes their routine use in the flutter analysis of complex flight vehicles.

There is potential for increasing the efficiency of turbulent flow simulations by the development of improved numerical methods. A popular way of solving the Navier–Stokes equations is by the alternating direction implicit (ADI) method. In two dimensions, with the spatial discretization of convection based on central differencing, this approach yields an unconditionally stable method, whereas in three dimensions the extension is unconditionally unstable in the absence of artificial dissipation. For upwind differencing the stability limit on the time step is very restrictive. Hence, the generalization to three dimensions of the most popular method for simulating two-dimensional unsteady viscous flows is unpromising. However, this method is widely used in three dimensions with examples including the NASA Ames ENSAERO code, which has been used for aeroelastic studies,[1–3] the study of vortical flows over delta wings,[4,5] the NASA code CFL3D for aeroelastic analysis,[6] and to simulate viscous flow over oscillating wings.[7]

An alternative to ADI is the lower-upper factorization method, which retains its favorable stability properties in three dimensions.[8] Subiteration to remove factorization error effects was also used to good effect. Another possible alternative is the dual time method with convergence acceleration techniques such as multigrid applied in a pseudotime at each physical time step (for example, see Ref. 9 for incompressible flows). In Ref. 10 a hybrid approach was proposed that uses Gauss–Seidel relaxation in one coordinate direction and a two-factor approximate factorization in the remaining cross-flow plane.

In the current paper an alternative for three-dimensional flows is proposed that involves a novel two-factor method that builds on work on an unfactored method for two-dimensional flows. The two-dimensional method is based on a preconditioned conjugate gradient linear iterative solver with the ADI factorization serving as the preconditioner. Application of this method has been made to steady aerofoil flows[11,12] and to aerofoil flows with forced pitching and flap movement.[13] Parallel computing for aerofoil flows has been considered in Ref. 14, and the parallel implementation of the two-dimensional method, together with the method of this paper, is discussed in Ref. 15.

This paper is organized as follows. First, the two-dimensional method is discussed, and its performance for a steady-state flow over an aerofoil is summarized. Then, a novel two-factor method for three-dimensional flows that builds on the two-dimensional method is presented. Sample results for the steady flow over the ONERA M6 wing at transonic conditions are given.

II. Two-Dimensional Method

The two-dimensional thin-layer Reynolds-averaged Navier–Stokes equations in generalized curvilinear coordinates (ξ, η) with η normal to the surface can be denoted in nondimensional conservative form by

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

where $\boldsymbol{w}$ denotes the vector of conserved variables, $\boldsymbol{f}$ the convective streamwise flux, $\boldsymbol{g}$ the convective normal flux, and $\boldsymbol{s}$ the normal viscous flux.

One implicit step, updating the primitive variables $\mathcal{P}$, can be written as

$$\left(\frac{\partial \boldsymbol{w}}{\partial \mathcal{P}} + \Delta t \frac{\partial \boldsymbol{R}_\xi^n}{\partial \boldsymbol{w}} + \Delta t \frac{\partial \boldsymbol{R}_\eta^n}{\partial \boldsymbol{w}}\right)\delta \mathcal{P} = -\Delta t\left(\boldsymbol{R}_\xi^n + \boldsymbol{R}_\eta^n\right) \tag{2}$$

where $\boldsymbol{R}_\xi$ and $\boldsymbol{R}_\eta$ are terms arising from the spatial discretization in the ξ and η directions, respectively, and

$$\frac{\partial \boldsymbol{f}}{\partial \xi} \approx \boldsymbol{R}_\xi \qquad \frac{\partial (\boldsymbol{g} - \boldsymbol{s})}{\partial \eta} \approx \boldsymbol{R}_\eta$$

$$\delta \mathcal{P} = \mathcal{P}^{n+1} - \mathcal{P}^n$$

In the present work the spatial terms are discretized using Osher's flux approximation with MUSCL interpolation and the Von Albada limiter for the convective terms and central differencing for the viscous fluxes. The Baldwin–Lomax turbulence model is employed to provide a turbulent contribution to the viscosity, but this is not linearized in time in the present work; i.e., turbulence contributions

Received Feb. 8, 1995; revision received May 2, 1995; accepted for publication May 3, 1995; presented as Paper 95-1711 at the AIAA 12th Computational Fluid Dynamics Conference, San Diego, CA, June 19–22, 1995.

*Lecturer, Department of Aerospace Engineering.

†Professor, Department of Aerospace Engineering. Associate Fellow AIAA.

only appear on the right-hand side of Eq. (2). This has been found not to degrade the stability properties of the methods examined in this paper.

The alternating direction implicit version of Eq. (2) is

$$\left(\frac{\partial w}{\partial \mathcal{P}} + \Delta t \frac{\partial R_\xi^n}{\partial w}\right)\left(\frac{\partial w}{\partial \mathcal{P}}\right)^{-1}\left(\frac{\partial w}{\partial \mathcal{P}} + \Delta t \frac{\partial R_\eta^n}{\partial w}\right)\delta \mathcal{P}^n = R_{\exp} \quad (3)$$

where

$$R_{\exp} = -\Delta t\left(R_\xi^n + R_\eta^n\right)$$

The ADI factorization that appears on the left-hand side of Eq. (3) has been widely used to approximate a solution to the system (2) because the banded structure of each of the factors makes it relatively easy to solve. However, the solution of the ADI system is not an exact solution of Eq. (2), and in practice the factorization error [the error introduced by solving Eq. (3) rather than Eq. (2)] leads to a practical limit on the time step and introduces another source of error into the calculation. This motivates the use of a preconditioned conjugate gradient solution of the unfactored system.

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 Ref. 16, 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. CGS has the additional advantage that the transpose of the matrix on the left-hand side of Eq. (2) is not required, reducing implementation difficulties. The CGS algorithm was derived in Ref. 17 and is restated in Ref. 18.

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

$$Ax = b \quad (4)$$

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

$$C^{-1}Ax = C^{-1}b \quad (5)$$

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

$$C^{-1}b = x \quad (6)$$

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 are put in triangular form once at each time step with the row operations being stored for use at each multiplication by the preconditioner. This roughly doubles the storage requirements of the method.

To illustrate the performance of this method we present results for flow over an RAE2822 aerofoil at a freestream Mach number of 0.73, an angle of attack of 2.73 deg, and a Reynolds number of 6.5×10^6. A study of convergence criteria for this case was carried out, which suggested that a good general flow solution can be obtained after a 2.5-order reduction in the residual from the value at freestream conditions and that the integrated coefficients are well converged after a four-order reduction.[11] The comparison of convergence rates for the present method (called AF-CGS), straight ADI, and an explicit local time-stepping method is shown in Fig. 1 In this paper a work unit is defined as the CPU time for one residual evaluation (i.e., the time to calculate the left-hand side of Eq. (2) or roughly the time for one explicit time step). It is clear that the new method yields an improved convergence rate of around 25% over ADI for the present problem. The Courant–Friedrichs–Lewy (CFL) number that yields fastest convergence for the AF-CGS method is 35, but CFL numbers of up to 50 can be used. The largest CFL number that yields a solution for ADI is 18, and hence removing the factorization error allows the use of larger time steps. A further reduction in the time to convergence by a factor of 5 has been achieved by mesh sequencing, but these results are excluded here to allow a comparison of the solution algorithms alone.

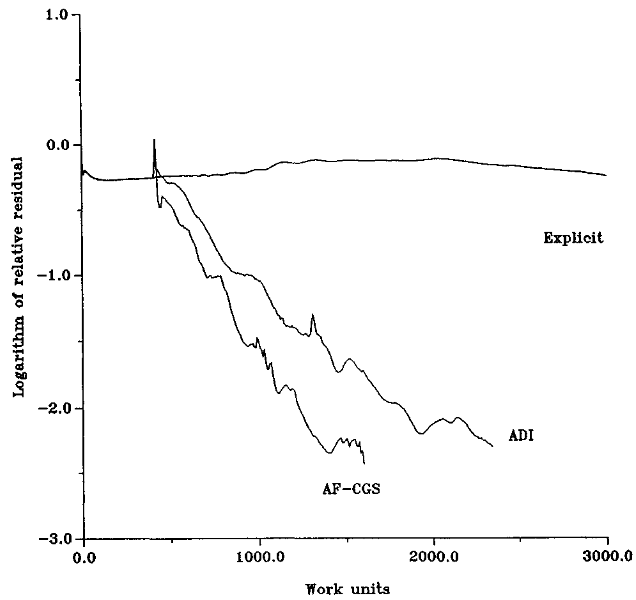

Fig. 1 Convergence histories for the AF-CGS, ADI, and explicit methods.

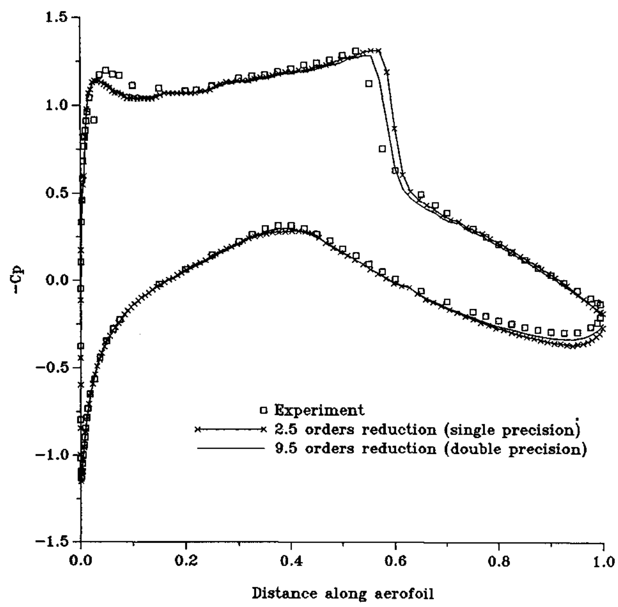

Fig. 2 Computed pressure distribution compared with the experimental distribution.

A similar approach has been used for unsteady flows over pitching and plunging aerofoils and aerofoils with moving flaps.[13] The main conclusion from this work was that AF-CGS does not allow the choice of time step from purely accuracy considerations because of the need to limit the time step to ensure the reasonable performance of the linear solver. However, AF-CGS does allow for larger time steps and a reduced computational cost when compared with ADI. For one particular case the stability restriction on the size of the time step is a global CFL number of 1000. The average CFL number during one cycle for the unfactored method is around 2000 for the unfactored method, translating into a saving in CPU time of around 25%.

III. Three-Dimensional Extensions

The extension of the method to three dimensions is complicated by two considerations. First, computer storage becomes a limiting factor due to the need to store large Jacobian matrices. Second, the

ADI factorization in three dimensions is significantly worse than in two dimensions, making its use as a preconditioner less favorable. This fact, however, means that there are increased gains to be made in three dimensions by the use of an alternative to ADI.

One step of the method considered can be written as

$$\left(\frac{\partial w}{\partial \mathcal{P}} + \Delta t\frac{\partial R_x}{\partial \mathcal{P}} + \Delta t\frac{\partial R_y}{\partial \mathcal{P}}\right)\frac{\partial w}{\partial \mathcal{P}}^{-1}\left(\frac{\partial w}{\partial \mathcal{P}} + \Delta t\frac{\partial R_z}{\partial \mathcal{P}}\right)\delta\mathcal{P} = R_{\exp} \tag{7}$$

where

$$R_{\exp} = -\Delta t(\boldsymbol{R}_x + \boldsymbol{R}_y + \boldsymbol{R}_z)$$

This two-factor step can be loosely described as unfactored in each spanwise slice and approximately factored in the spanwise direction. A stability analysis[12] has shown that the method has similar stability properties to the two-factor ADI method in two dimensions, representing a significant improvement on the behavior of the three-factor method in three dimensions. The linear system resulting from the first factor in Eq. (7) has a more complicated structure than the block pentadiagonal systems that are encountered for each factor in the three-factor method. However, this system can be solved using a direct generalization of the method described earlier for two dimensions; i.e., we solve the system

$$C^{-1}Ax = C^{-1}b \tag{8}$$

by the CGS method where

$$A = \left(\frac{\partial w}{\partial \mathcal{P}} + \Delta t\frac{\partial R_x}{\partial \mathcal{P}} + \Delta t\frac{\partial R_y}{\partial \mathcal{P}}\right) \tag{9}$$

$$C = \left(\frac{\partial w}{\partial \mathcal{P}} + \Delta t\frac{\partial R_x}{\partial \mathcal{P}}\right)\frac{\partial w}{\partial \mathcal{P}}^{-1}\left(\frac{\partial w}{\partial \mathcal{P}} + \Delta t\frac{\partial R_y}{\partial \mathcal{P}}\right) \tag{10}$$

and

$$b = -\Delta t(\boldsymbol{R}_x + \boldsymbol{R}_y + \boldsymbol{R}_z) \tag{11}$$

followed by the solution of a block pentadiagonal system for the updates

$$\left(\frac{\partial w}{\partial \mathcal{P}} + \Delta t\frac{\partial R_z}{\partial \mathcal{P}}\right)\delta\mathcal{P} = x \tag{12}$$

The two-factor method has substantially reduced memory requirements compared with the fully unfactored method. For the third-order spatial discretisation there are 13 nonzero 5×5 blocks for the rows in the unfactored matrix associated with any one grid cell. This means that the number of floating point numbers that must be stored for the coefficient matrix for a mesh with $\mathcal{N}$ cells is $325\mathcal{N}$. Since $\mathcal{N}$ can be on the order of 1×10^6 for flows around basic wings, this implies that even if we can solve the linear system efficiently, storage requirements will be a limiting factor. For the two-factor method only the matrix for one spanwise slice or one line in the spanwise direction need be stored at any one time. This has the effect of reducing the matrix storage requirements at any one time in the calculation to $\max(225\mathcal{N}_{\text{slice}}, 125\mathcal{N}_{\text{line}})$ where $\mathcal{N}_{\text{slice}}$ is the number of grid points in a spanwise slice and $\mathcal{N}_{\text{line}}$ is the number of grid points in the spanwise direction. Since $\mathcal{N}_{\text{line}}\mathcal{N}_{\text{slice}} = \mathcal{N}$, it can be seen that the storage requirements have been reduced substantially (by around two orders of magnitude for the test case examined in this paper).

As a test case, we consider flow over the ONERA M6 wing at transonic conditions. Experimental data for this wing are available in Ref. 19, and there are several previously published computational studies including those in Ref. 20. The flow problem we consider here has a freestream Mach number of 0.84, an incidence of 3.06 deg, and a Reynolds number of 11×10^6. The convergence history is shown in Fig. 4 for a grid with $121 \times 33 \times 33$ points. For this case 300 explicit steps were required before FUN was used with a CFL number of 10. A plateau in the convergence is reached when the residual has been reduced by around three orders of magnitude from the initial value. This was also observed for flows over aerofoils in Ref. 11 and was found to be due to small oscillations in the pressure at the far field when single precision storage was used. This plateau was not encountered when double precision storage was used. No attempt has been made in the present case to verify if the plateau is due to the single precision storage used.

The convergence history for three-factor ADI is also shown. The FUN method requires 25% less CPU time compared with ADI to reduce the residual by two orders of magnitude. ADI was unstable for CFL numbers larger than 5. As discussed for the two-dimensional case, there is a balance for the AF-CGS method (which is used to solve each spanwise factor) between increasing the CFL number to reduce the number of implicit steps and decreasing the CFL number to reduce the number of CGS iterations at each time step. For the present case the optimal CFL number turns out to be 10, although the method produces comparable convergence rates for CFL numbers of up to 20. It should be noted that the spanwise factors could be solved by other methods and alternative preconditioners that do not deteriorate with increasing time steps are under consideration with potentially large benefits in terms of allowing larger CFL numbers to be used.

Multigrid together with an approximately factored lower-upper (LU) implicit scheme was used in Ref. 21 to study inviscid flow on stretched grids for the same test case. A reduction of three orders in the residual was achieved in 500 implicit steps on a single grid and multigrid improved the convergence rate by a factor of 6.5. To allow a rough comparison with present results, one step of the implicit method described in this paper takes around 8 work units. The LU steps of Ref. 21 should be cheaper than the FUN steps, but this estimate suggests that the convergence achieved here is comparable with the single grid rate of Ref. 21.

In Ref. 20, a comparable level of convergence has been achieved on a finer grid by using multigrid together with explicit time stepping in around 2000 work units. This is an impressive result given the size of the grid used and is beyond the basic method presented here. However, a study for two-dimensional problems has suggested that mesh sequencing can yield a fivefold reduction in the time to convergence by providing a good fine grid starting solution. This technique has not been used here and would improve the comparative performance. Also, the results of Ref. 21 suggest that multigrid coupled with the present method would yield a very competitive method for steady-state flows.

The comparison of the computed pressure distribution with the experimental results of Ref. 19 at six spanwise slices is shown in Fig. 3, and again good agreement is observed. The results on two meshes, the first with $121 \times 33 \times 33$ points and the second with

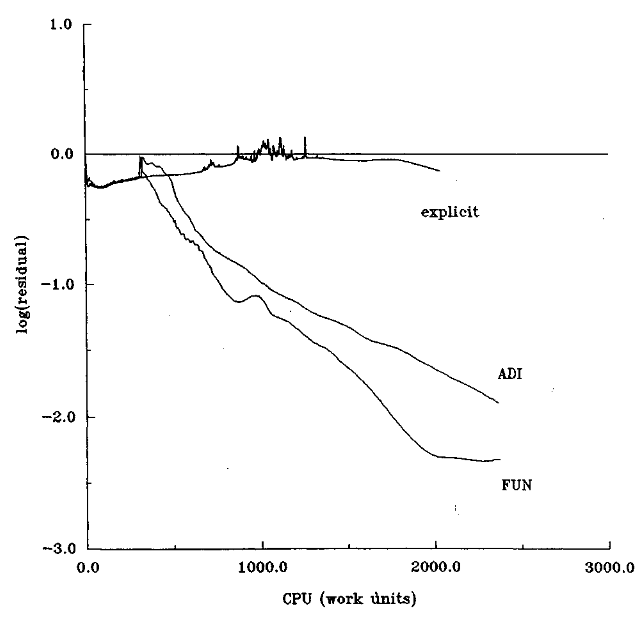

Fig. 3 Convergence history for the factored-unfactored (FUN) methods for ONERA M6 wing test case.

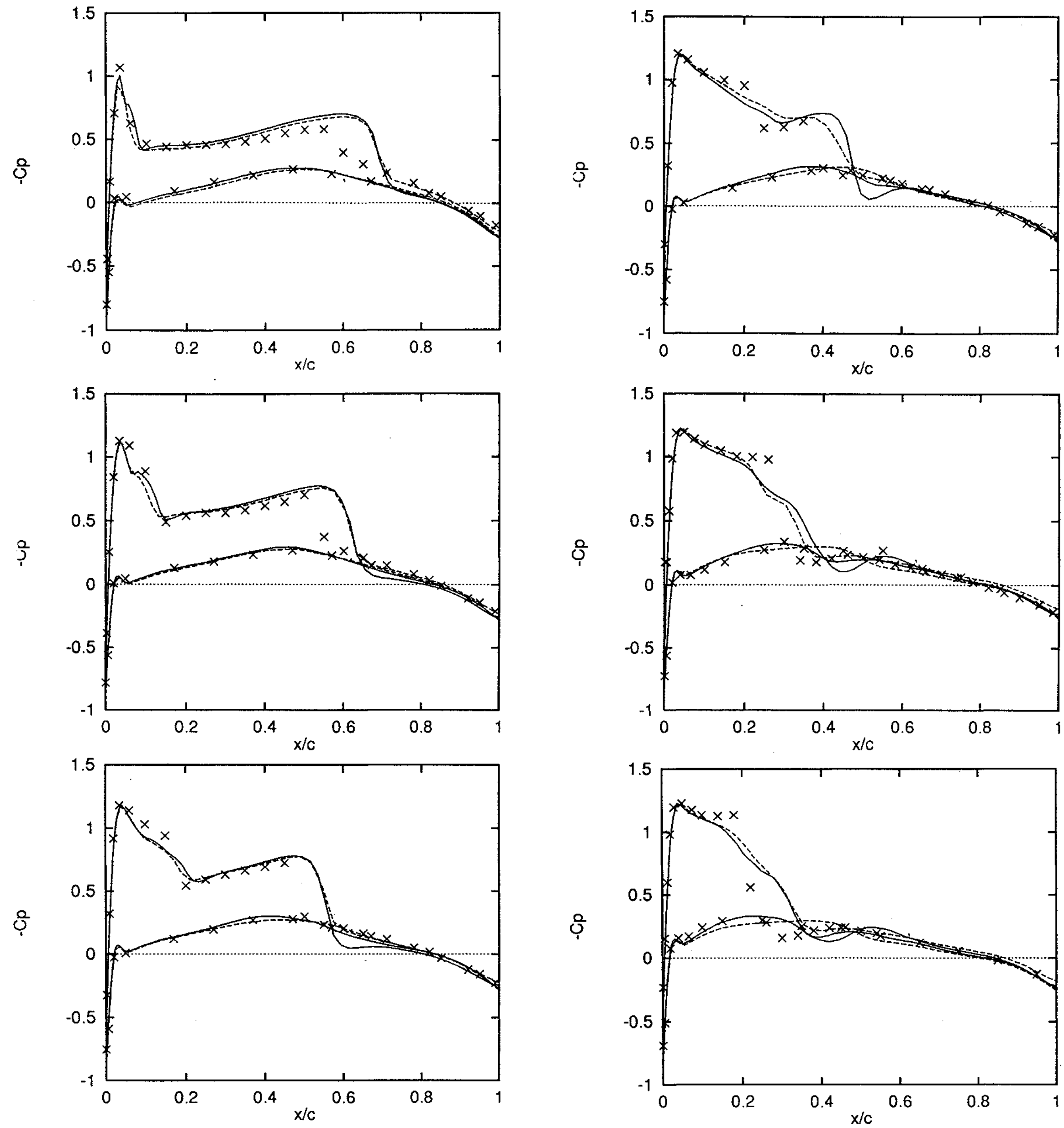

Fig. 4 Comparison of computed pressure distribution with experiment for ONERA M6 wing: solid line—191 × 33 × 49 grid, and dashed line—121 × 33 × 33 grid.

191 × 33 × 49, are shown and the close agreement suggests that a good solution can be obtained on the coarser mesh. The results are in line with those of Ref. 20. It should be noted that the mesh used here is of C-H type that results is skewed cells in the region of the wing tip. It has been observed[22] that C-O type grids give better quality solutions for this type of flow.

IV. Conclusions

The development of an implicit method for simulating three-dimensional compressible and viscous flow has been discussed. This method is based on a two-dimensional method that consists of an iterative solution of the linear system by the conjugate gradient squared algorithm with preconditioning by the alternating direction implicit approximate factorization. This method has been tested for steady and unsteady flows and an improvement in efficiency (i.e., a reduction in execution time) has been noted when compared with the standard ADI method.

A novel two-factor method has been presented for three-dimensional flows that builds on the two-dimensional method by factoring the linear system into a factor arising from spanwise slices in the mesh and a block pentadiagonal factor arising from strips in the spanwise direction. The more complicated factors arising from the spanwise slices are solved by the two-dimensional method. This approach yields a method that has similar stability properties to the two-dimensional ADI method, a situation that is substantially better than for the three-dimensional version of ADI. Results have been presented for a transonic flow over the ONERA M6 wing that demonstrate reasonable accuracy and convergence characteristics compared with multigrid methods employing explicit[20] and implicit[21] time stepping. A reduction of 25% in the CPU time to reduce the residual by two orders of magnitude was noted when compared with ADI. Further improvements are likely with the use of preconditioners for the solution of the spanwise factors that do not degrade with the time step.

However, as with the two-factor ADI method in two dimensions, the real potential of the method lies with the simulation of unsteady flows. Future extensions of this work will include the investigation of the performance for unsteady and aeroelastic three-dimensional flows. It is anticipated that the reduction in the factorization error will allow larger time steps to be taken when compared with the three-factor method.

Acknowledgments

This work was funded under SERC/MoD Contract GR H 47371 and EPSRC/MoD Contract GR H 47371 and by British Aerospace and the MoD. Computer time on the EPSRC Cray T3D located at the Edinburgh Parallel Computing Centre was obtained under the Computation of Complex Aerodynamic Flows initiative. The mesh generation subroutines were supplied by A. L. Gaitonde of Bristol University, for which the authors are grateful. The authors are also grateful to Ian Glover for his contribution to the early part of this work.

References

[1]Guruswamy, G. P., "Navier–Stokes Computations on Swept-Tapered Wings, Including Flexibility," *Journal of Aircraft*, Vol. 29, 1992, pp. 588–597.

[2]Chaderjian, N. M., and Guruswamy, G. P., "Transonic Navier–Stokes Computations for an Oscillating Wing Using Zonal Grids," *AIAA Journal*, Vol. 29, 1992, pp. 326–335.

[3]Guruswamy, G. P., "Unsteady Aerodynamic and Aeroelastic Calculations for Wings Using Euler Equations," *AIAA Journal*, Vol. 28, 1990, pp. 461–469.

[4]Ekaterinaris, J. A., and Schiff, L. B., "Numerical Prediction of Vortical Flow over Slender Delta Wings," *Journal of Aircraft*, Vol. 30, 1993, pp. 935–942.

[5]Fujii, K., Gavali, S., and Holst, T. L., "Evaluation of Navier–Stokes and Euler Solutions for Leading-Edge-Separation Vortices," *International Journal of Numerical Methods in Fluids*, Vol. 8, 1988, pp. 1319–1329.

[6]Robinson, B. A., Batina, J. T., and Yang, H. T. Y., "Aeroelastic Analysis of Wings Using the Euler Equations with a Deforming Mesh," *AIAA Journal*, Vol. 28, 1991, pp. 781–788.

[7]Nakamichi, J., "Calculations of Unsteady Aerodynamics over Oscillating Wings," *European Forum on Aeroelasticity and Structural Dynamics*, Deutsche Gesellschaft fur Luftund Raumfahrt e.V., 1989, pp. 123–131.

[8]Simpson, L. B., and Whitfield, D. L., "Flux-Difference Split Algorithm for Unsteady Thin-Layer Navier–Stokes Solutions," *AIAA Journal*, Vol. 30, 1992, pp. 914–922.

[9]Belov, A. A., Martinelli, L., and Jameson, A., "A Novel Fully Implicit Multigrid Driven Algorithm for Unsteady Incompressible Flow Calculation," *Second European Conference on CFD*, ECCOMAS, 1994, pp. 663–670.

[10]Thomas, J. L., Van Leer, B., and Walters, R. W., "Implicit Flux-Split Schemes for the Euler Equations," *AIAA Journal*, Vol. 28, 1990, pp. 973, 974.

[11]Badcock, K. J., Glover, I. C., and Richards, B. E., "A Preconditioner for Steady Two-Dimensional Turbulent Flow Simulation" (submitted for publication).

[12]Badcock, K. J., Glover, I. C., and Richards, B. E., "Convergence Acceleration for Viscous Aerofoil Flows Using an Unfactored Method," *Second European Conference on CFD*, ECCOMAS, 1994, pp. 333–341.

[13]Badcock, K. J., and Gaitonde, A. L., "An Unfactored Method with Moving Meshes for Solution of the Navier–Stokes Equations for Flows about Aerofoils" (submitted for publication).

[14]Badcock, K. J., "A Partially-Implicit Method for Simulating Viscous Aerofoil Flows," *International Journal of Numerical Methods in Fluids*, Vol. 19, 1994, pp. 259–268.

[15]Badcock, K. J., "The Implementation of Two and Three Dimensional Flow Codes on Various Parallel Platforms, Glasgow Univ., Aerospace Engineering Rept. 15, Glasgow, Scotland, UK, 1994.

[16]Badcock, K. J., "Newton's Method for Laminar Aerofoil Flows," Glasgow Univ., Aerospace Engineering Rept. 30, Glasgow, Scotland, UK, 1993.

[17]Sonneveld, P., "CGS: A Fast Lanczos-Type Solver for Nonsymmetric Linear Systems," *SIAM Journal Statistics and Computing*, Vol. 10, 1989, pp. 36–52.

[18]Vitaletti, M., "Solver for Unfactored Schemes," *AIAA Journal*, Vol. 29, 1991, pp. 1003–1005.

[19]Schmitt, V., and Charpin, F., "Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers," AGARD, TR AR-138, 1979.

[20]Radespiel, R., Rossow, C., and Swanson, R. C., "Efficient Cell-Vertex Multigrid Scheme for the Three-Dimensional Navier–Stokes Equations," *AIAA Journal*, Vol. 28, 1990, pp. 1464–1472.

[21]Yoon, S., and Kwak, D., "Multigrid Convergence of an Implicit Symmetric Relaxation Scheme," *AIAA Journal*, Vol. 32, 1994, pp. 950–955.

[22]Crumpton, P. I., "A Cell Vertex Method for 3D Navier–Stokes Solutions," Oxford Univ. Computing Lab., Numerical Analysis Rept. 9, Oxford, England, UK, 1993.