University of Glasgow

DEPARTMENT OF

AEROSPACE ENGINEERING

Time Stepping Approaches for Three-Dimensional Transonic Viscous Flows

K.J. Badcock and I.C. Glover

Glasgow University Aero report 9412

Time Stepping Approaches for Three-Dimensional Transonic Viscous Flows

K.J. Badcock and I.C. Glover

Glasgow University Aero report 9412

Time Stepping Approaches for Three-Dimensional Transonic Viscous Flows

K.J. Badcock and I.C.Glover
Aerospace Engineering Department
University of Glasgow,
Glasgow, G12 8QQ, U.K.

July 27, 1994

Abstract

A novel time-stepping approach is presented for the three-dimensional thin-layer Navier-Stokes equations. The method involves a two factor approximate factorisation and uses a preconditioned conjugate gradient solution for one of the factors. The method has potential to provide most of the advantages of a fully unfactored method without the huge memory requirements. A partially implicit method is also considered. Results and a stability analysis are presented to evaluate the approaches.

1 Introduction

There is considerable potential for saving valuable computer time by careful examination of solution methods for the equations describing three-dimensional flows. Simulations based on the compressible Navier-Stokes equations are important in aerospace engineering because of their generality which in theory should lead to the accurate prediction of shock waves, boundary layers and flow separation. However, due to the large amount of computation required to solve these equations, coarse grids and poorly converged solutions often mean that the advantages are not fully realised due to the poor quality of the approximation to the flow field.

To obtain steady flow solutions quickly it is crucial to select appropriate ways of accelerating the convergence to steady state. One way of doing this is to use an implicit treatment of some or all terms to allow the use of larger time steps. Two particular approaches were examined for aerofoil flows in [1] and [2]. The first method used an implicit treatment for terms normal to an aerofoil and yielded high efficiency in parallel. The second method, called AF-CGS, which proved preferable overall, used an implicit treatment of all spatial terms together with an unfactored solution of the resulting linear system by the conjugate gradient squared (CGS) iterative method together with preconditioning based on the alternating direction implicit (ADI) factorisation.

The generalisation of these methods to solve for three-dimensional flows is complicated by several issues. First, computer storage becomes a limiting factor because of the need to store large Jacobian matrices. This leads to a giga-byte requirement on relatively small meshes. Secondly, the ADI factorisation is significantly worse in three-dimensions than in two. Thirdly, the relative grid spacings required to resolve scales in the three directions is different compared with the two dimensional problem and hence opens up new possibilities for efficient methods, eg, an explicit treatment in the spanwise direction of a wing along with an implicit treatment in the streamwise and normal directions.

In the current report we discuss various alternatives for the efficient solution of the three-dimensional Navier-Stokes equations. The discussion emphasises the likely application of each method. Analysis is presented to demonstrate the stability properties of two of the methods proposed. Finally, results are presented for a transonic laminar test case which illustrate points raised in the discussion and analysis.

2 Equations

The three dimensional thin-layer Navier-Stokes equations are given in Cartesian co-ordinates by

$$\frac{\partial \mathbf{w}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} + \frac{\partial \mathbf{g}}{\partial y} + \frac{\partial \mathbf{h}}{\partial z} = \frac{\partial \mathbf{s}}{\partial y} \qquad (1)$$

where

$$\mathbf{w} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \mathbf{f} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ u(e+p) \end{bmatrix}, \mathbf{g} = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ v(e+p) \end{bmatrix}, \mathbf{h} = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ w(e+p) \end{bmatrix}$$

$$\mathbf{s} = \begin{bmatrix} 0 \\ \sigma_{xy} \\ \sigma_{yy} \\ \sigma_{zy} \\ u\sigma_{xy} + v\sigma_{yy} + w\sigma_{zy} - q_y \end{bmatrix}.$$

Here,

$$\sigma_{yy} = 2\mu v_y - \frac{2}{3}\mu(u_x + v_y + w_z), \quad \sigma_{xy} = \sigma_{yx} = \mu(u_y + v_x), \quad \sigma_{zy} = \sigma_{yz} = \mu(v_z + w_y)$$

$$q_y = -\kappa\frac{\partial T}{\partial y}, \quad p = (\gamma - 1)(e - \frac{1}{2}\rho(u^2 + v^2 + w^2)),$$

$$T = c_v(\frac{e}{\rho} - \frac{1}{2}(u^2 + v^2 + w^2)).$$

The symbols ρ, u, v, w, e, p, μ, κ, T represent the fluid density, the three components of velocity, energy, pressure, viscosity, heat conductivity and temperature respectively. The constants γ and c_v stand for the ratio of the specific heats and the specific heat at constant volume respectively. The fluid viscosity is assumed to vary with temperature by Sutherland's law. The spatial discretisation used is Osher's method for the inviscid terms with MUSCL interpolation and a flux limiter used to provide third order spatial accuracy. Central differencing is used for the viscous fluxes.

3 Time-Stepping Methods

The efficiency of the underlying numerical method is particularly important for three-dimensional flow simulations due to the high computer costs involved. Large CPU time and memory requirements are limiting factors which must be overcome. In this section we discuss some time-stepping methods for steady and unsteady flows over wings. The following notation is used in describing the numerical method:

$$\frac{\partial \mathbf{f}}{\partial x} \approx R_x$$

$$\frac{\partial(\mathbf{g} - \mathbf{s})}{\partial y} \approx R_y$$

$$\frac{\partial \mathbf{h}}{\partial z} \approx R_z.$$

method	formulation
explicit	$\delta w = -\Delta t(R_x + R_y + R_z) = \delta^{exp}$
implicit	$(I + \Delta t\frac{\partial R_x}{\partial w} + \Delta t\frac{\partial R_y}{\partial w} + \Delta t\frac{\partial R_z}{\partial w})\delta w = \delta^{exp}$
partially implicit	$(I + \Delta t\frac{\partial R_x}{\partial w} + \Delta t\frac{\partial R_y}{\partial w})\delta w = \delta^{exp}$
F/UNF	$(\frac{\partial w}{\partial p} + \Delta t\frac{\partial R_x}{\partial p} + \Delta t\frac{\partial R_y}{\partial p})\frac{\partial w}{\partial p}^{-1}(\frac{\partial w}{\partial p} + \Delta t\frac{\partial R_z}{\partial p})\delta p = \delta^{exp}$

Table 1: *Time stepping methods for 3D case,*

The time stepping methods which we consider are shown in table 1. Here $\delta \mathbf{w} = \mathbf{w}^{n+1} - \mathbf{w}^n$ and $\delta \mathbf{p} = \mathbf{p}^{n+1} - \mathbf{p}^n$ where $\mathbf{w}$ and $\mathbf{p}$ are the vectors of cell conserved and primitive variables respectively and the superscript denotes the time level of the approximation.

The potential for using the fully implicit method is limited because of the very high memory requirements. Also, the size of the linear system means that a direct or iterative solution is significantly harder than in two dimensions. The generalisation of the AF-CGS method [2] to three dimensions is unpromising because of these two problems and also because of the poor quality of the three factor approximate factorisation.

The explicit method has the advantage of low memory requirements relative to the fully implicit method. However, from experience in two-dimensions, the long time to convergence makes this method inefficient and this problem is likely to be worse for the more complex flows encountered in three dimensions.

The problems associated with the explicit and fully unfactored methods motivate the investigation of the partially implicit and the factored-unfactored (F/UNF) methods. Both of these approaches overcome the memory limitation because the matrix of the linear system only needs to be stored for one spanwise factor at a time and hence the major memory requirements are similar to that of an unfactored method for each two-dimensional spanwise slice. Two-dimensional problems on meshes with over 15000 grid points can be solved on workstations with 64 Mb of memory by the fully implicit method and so large three-dimensional problems can be solved on workstations using these two methods.

The partially implicit method is likely to yield the most efficient use of a parallel computer like the intel Hypercube with each spanwise slice being solved on a processor without communication after an initial setup stage with the communication requirements of an explicit method. However, for grids which are strongly clustered at the wing tip it is likely that stability limits arising from the spanwise explicit treatment of the partially implicit method will be too restrictive and for these cases the fully implicit or the factored-unfactored (F/UNF) methods should have an advantage.

The fully factored method in three dimensions is theoretically unconditionally unstable. However, it is shown in the next section that the two factor (or factored-unfactored) method has similar stability properties to the two factor method in two dimensions. A loss in efficiency of up to fifty per-cent has been noted for the factored method in two-dimensions when applied to aerofoil flows but, as discussed above, the generalisation of unfactored methods from two dimensions to three is likely to bring reduced performance. The problem with the factored-unfactored method is that a linear system must be solved which is more complicated in its sparsity pattern than the simple banded matrices normally encountered in approximately factored methods. However, the more complex factors can be solved by an extension of the AF-CGS method i.e.

$$\left(\frac{\partial w}{\partial p} + \Delta t\frac{\partial R_x}{\partial p} + \Delta t\frac{\partial R_y}{\partial p}\right)\mathbf{x} = \mathbf{y}$$

is solved by the CGS method with a preconditioner given by inverse of the approximate factorisation

$$\left(\frac{\partial w}{\partial p} + \Delta t\frac{\partial R_x}{\partial p}\right)\frac{\partial w}{\partial p}^{-1}\left(\frac{\partial w}{\partial p} + \Delta t\frac{\partial R_y}{\partial p}\right)$$

The factored-unfactored method is formulated in primitive variables to overcome a problem with the AF-CGS method which was first encountered in two dimensions The difficulty with the conserved variable formulation is that large relative errors in the CGS solution become small absolute errors when scaled by the cell volume when the conserved variable formulation is used.

A crucial part of the three methods involving some form of implicit treatment is the calculation of the Jacobian matrix. This is achieved in the present work by a fully analytical derivation using the symbolic manipulation package *REDUCE*. The formulation follows the method used for the Jacobians in two-dimensions as given in [3].

4 Newton scalar analysis

In order to give an indication of the stability of the factored-unfactored method, a Newton scalar analysis is carried out along the lines of [4]. A scalar model of the problem is written as

$$u_t + r(u) = 0 \tag{2}$$

where $r(u) = f_1(u) + f_2(u)$ in two dimensions, and $r(u) = f_1(u) + f_2(u) + f_3(u)$ in three dimensions. The three dimensional factored algorithm is written as

$$\{1+\Delta t f_1^{|}(u^n)\}\{1+\Delta t f_2^{|}(u^n)\}\{1+\Delta t f_3^{|}(u^n)\}(u^{n+1} - u^n) = -\Delta t r(u^n) \tag{3}$$

where the f_i represent the differencing signature which is determined by the choice of spatial differencing. When the f_i are complex, it can be shown that the algorithm is at best conditionally stable. However, in the 'worst case' of the f_i being purely imaginary, it can be shown that for any Δt there exist modes for which the modulus of the amplification factor is greater than one, and so the method is unconditionally unstable.

We shall consider the two dimensional factored algorithm, and the three-dimensional factored-unfactored algorithm for the 'worst case'.

For the two dimensional factored algorithm we have

$$\{1 + \Delta t f_1^{|}(u^n)\}\{1 + \Delta t f_2^{|}(u^n)\}(u^{n+1} - u^n) = -\Delta t r(u^n) \tag{4}$$

Following the analysis of [4] leads to the following equation for the global error $\epsilon^n = u^n - v$, where v is the solution of $r(u) = 0$,

$$\epsilon^{n+1} = \sigma(\Delta t)\epsilon^n + \frac{g^{||}(\xi)}{2}(\epsilon^n)^2. \tag{5}$$

Here, the function g is obtained by writing (4) in the form $u^{n+1} = g(u^n)$, and $\sigma(\Delta t)$ is given by

$$\sigma(\Delta t) = \frac{1 + (\Delta t)^2 f_1^{|}(v) f_2^{|}(v)}{[1 + \Delta t f_1^{|}(v)][1 + \Delta t f_2^{|}(v)]}.$$

Writing $f_1^{|}(v) = \alpha i$ and $f_2^{|}(v) = \beta i$, where $i = \sqrt{-1}$, we find that

$$|\sigma(\Delta t)| = \frac{(1 - (\Delta t)^2 \alpha\beta)}{[(1 - (\Delta t)^2 \alpha\beta)^2 + (\Delta t)^2(\alpha + \beta)^2]^{\frac{1}{2}}}.$$

Hence $|\sigma(\Delta t)|^2 \leq 1$ for all $\alpha, \beta \in \Re$. We also have that $\sigma(0) = 1$, and that $\sigma \to 1$ as $\Delta t \to \infty$. If we now consider $\frac{d|\sigma|}{d(\Delta t)}$ when Δt is small, neglecting powers of Δt of 2 or more leads to

$$\frac{d|\sigma|}{d(\Delta t)} \approx -\Delta t(\alpha + \beta)^2.$$

Hence $\sigma(\Delta t) < 1$ when Δt is small, which implies there exists a finite Δt which gives the optimal convergence rate.

Repeating this analysis for the three dimensional factored-unfactored method, which is given by

$$\{1 + \Delta t f_1^{|}(u^n) + \Delta t f_2^{|}(u^n)\}\{1 + \Delta t f_3^{|}(u^n)\}(u^{n+1} - u^n) = -\Delta t r(u^n) \quad (6)$$

we find that

$$\sigma(\Delta t) = \frac{1 + (\Delta t)^2 f_1^{|}(v) f_3^{|}(v) + (\Delta t)^2 f_2^{|}(v) f_3^{|}(v)}{[1 + \Delta t f_1^{|}(v) + \Delta t f_2^{|}(v)][1 + \Delta t f_3^{|}(v)]}.$$

Writing $f_1^{|}(v) = \alpha i$, $f_2^{|}(v) = \beta i$ and $f_3^{|}(v) = \gamma i$, we find that

$$|\sigma(\Delta t)|^2 = \frac{[1 - (\Delta t)^2(\alpha\gamma + \beta\gamma)]^2}{[1 - (\Delta t)^2(\alpha\gamma + \beta\gamma)]^2 + (\Delta t)^2(\alpha + \beta)^2}.$$

Hence $|\sigma(\Delta t)| \leq 1$ for all α, β and $\gamma \in \Re$. Again we have $\sigma(0) = 1$, and $\sigma \to 1$ as $\Delta t \to \infty$. Considering Δt small we have

$$\frac{d|\sigma|}{d(\Delta t)} \approx -\Delta t(\alpha + \beta)^2.$$

Hence $\sigma(\Delta t) < 1$ when Δt is small, which implies there exists a finite Δt which gives the optimal convergence rate.

The analysis suggests that the three dimensional factored-unfactored method behaves in a similar way to the two dimensional factored algorithm, rather than exhibiting the stability problems associated with the three dimensional factored algorithm in the absence of artificial dissipation.

5 Results

The test case considered is laminar flow over the IEPG delta wing on a $49 \times 17 \times 25$ mesh, with the following flow conditions:

$$M_\infty = 0.8, \quad \alpha = 0^\circ, \quad \text{Re} = 9.0 \times 10^6.$$

A comparison of the convergence histories for the explicit, partially implicit and factored-unfactored methods is shown in figure 1. A work unit is defined as the CPU time required for one iteration of the explicit method. The partially implicit and factored-unfactored results were both obtained with a global CFL number of 10. This CFL number corresponds to the optimal convergence rate for the factored-unfactored method. For the partially implicit method the convergence rate does not increase when the CFL number is increased beyond 5 because the time step throughout the mesh is bounded by the explicit stability limit in the spanwise direction. Hence, increasing the CFL number has a decreasing effect as the explicit limit bounds the time step in an increasing number of cells. Figure 2 shows that the approximately factored treatment in the spanwise direction overcomes this difficulty with the partially implicit method with faster convergence arising from the larger time steps. The factored-unfactored method achieves convergence significantly faster than the explicit method.

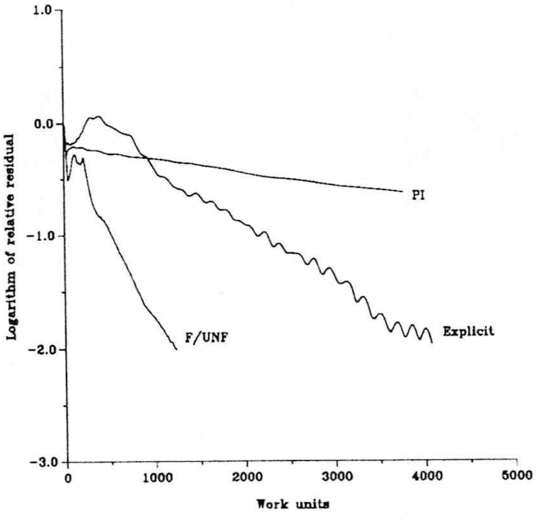

Figure 1: *Comparison of convergence histories for the explicit, partially implicit (PI) and factored-unfactored (F/UNF) methods.*

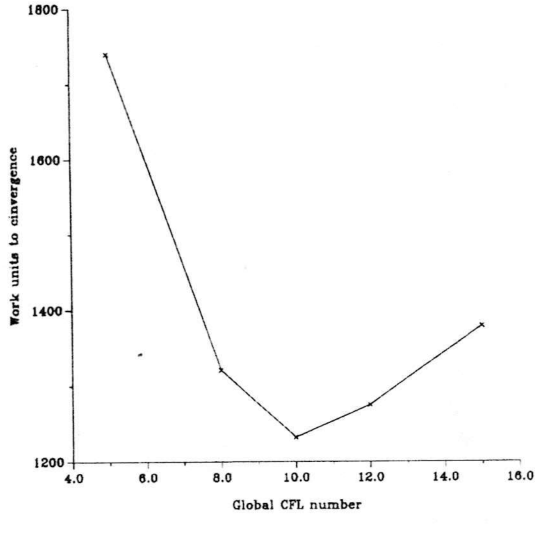

Figure 2: *Comparison of the number of work units to convergence for the factored-unfactored methods with varying values of the CFL number.*

6 Future Work

The results of this report have shown that the partially implicit and factored-unfactored time stepping methods both have the potential to efficiently solve for three-dimensional flows under certain circumstances. These two methods only require the storage at any one time of blocks in the Jacobian matrix arising from one slice in the mesh and hence the memory needs are reduced to a size which can easily fit onto a workstation.

The penalty for the partially implicit method is the time step restriction arising from the explicit treatment of terms in the spanwise direction of the wing. For the IEPG case used as a test problem above this restriction proves too severe and the convergence is degraded to the extent that the method is not competitive. However, this might not be the case for higher aspect ratio wings. Also, this method should yield very efficient use of a parallel computer since the communications are the same as for an explicit method and the computation on each node is increased by a factor of five. A speed-up of around 95 per-cent was observed for the explicit method on 16 nodes of an intel i860a Hypercube and hence the partially implicit method should be able to attain almost 100 per-cent of the theoretically possible speed up on this machine.

The factored-unfactored method shows a significant speed up over the explicit method. For problems on larger meshes and with more severe flow features this improvement is likely to be greater.

The fully unfactored method was not investigated because of the lack of a suitable machine with sufficient memory. In two-dimensions the unfactored method is around thirty per-cent quicker than the factored method. In three-dimensions this improvement is likely to be less due to the poorer quality of the factorisation involving three factors than that involving two in two-dimensions. Therefore, the degradation of convergence by using the factored-unfactored method is likely to be small.

Future work includes

- evaluation of factored-unfactored method for turbulent test cases
- testing on larger meshes using a parallel computer
- evaluate the three implicit methods for unsteady flows.

References

[1] K.J.Badcock. A parallelisable partially implicit method for unsteady viscous aerofoil flows. Technical report, G.U. Aero report 9312, 1993.

[2] K.J.Badcock, I.C.Glover, and B.E.Richards. Fast and accurate two-dimensional turbulent flow simulation. Technical report, G.U. Aero report 9326, 1993.

[3] K.J.Badcock and A.L.Gaitonde. An unfactored method with mesh regeneration for turbulent moving aerofoil simulations. *submitted for publication*, 1994.

[4] T.H. Pulliam. Implicit methods in CFD. In K.W. Morton and M.J. Baines, editors, *Numerical for Fluid Dynamics 3*, pages 117–136. Clarendon Press, 1988.