INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 14, 1–12 (1992)

A HELMHOLTZ PRESSURE EQUATION METHOD FOR THE CALCULATION OF UNSTEADY INCOMPRESSIBLE VISCOUS FLOWS

MORGAN WILLIAMS

Rockwell International, Rocketdyne Division, 6633 Canoga Ave, Mail Stop IB39, Canoga Park, CA 91304, U.S.A.

SUMMARY

A time-implicit numerical method for solving unsteady incompressible viscous flow problems is introduced. The method is based on introducing intermediate compressibility into a projection scheme to obtain a Helmholtz equation for a pressure-type variable. The intermediate compressibility increases the diagonal dominance of the discretized pressure equation so that the Helmholtz pressure equation is relatively easy to solve numerically. The Helmholtz pressure equation provides an iterative method for satisfying the continuity equation for time-implicit Navier–Stokes algorithms. An iterative scheme is used to simultaneously satisfy, within a given tolerance, the velocity divergence-free condition and momentum equations at each time step. Collocated primitive variables on a non-staggered finite difference mesh are used. The method is applied to an unsteady Taylor problem and unsteady laminar flow past a circular cylinder.

KEY WORDS Helmholtz pressure equation Unsteady incompressible flow Time-implicit algorithm

INTRODUCTION

The equations of motion for a two-dimensional, unsteady incompressible viscous fluid are

$$\partial_t \mathbf{u} + \partial_x (E - E_v) + \partial_y (F - F_v) + \nabla p = 0, \tag{1a}$$

$$\nabla \cdot \mathbf{u} = \partial_x u + \partial_y v = 0, \tag{1b}$$

where

$$\mathbf{u} = \{u, v\}^T, \quad E = \{u^2, uv\}^T, \quad F = \{vu, v^2\}^T, \quad \nabla p = \{\partial_x p, \partial_y p\}^T,$$

$$E_v = Re^{-1}\{2\partial_x u, (\partial_x v + \partial_y u)\}^T, \quad F_v = Re^{-1}\{(\partial_x v + \partial_y u), 2\partial_y v\}^T.$$

Here p is the pressure, u and v are the Cartesian velocity components and superscript T indicates transpose. The above equations are written in dimensionless form. The Cartesian co-ordinates x and y are non-dimensionalized by a reference length L_{ref}. The Cartesian velocities u and v are normalized by a reference velocity U_{ref}. The time t is non-dimensionalized with respect to L_{ref}/U_{ref} and the pressure is normalized by ρU_{ref}^2. The density ρ is simply a constant. The Reynolds number is defined as $Re = \rho U_{ref} L_{ref}/\mu$, where μ is the fluid viscosity. The pressure field must be computed so that equation (1b) is satisfied. For incompressible flow this can involve deriving an explicit relation for the pressure from equations (1a) and (1b). This is one of the fundamental problems of steady or unsteady incompressible flow calculations.

0271–2091/92/010001–12$06.00

Received 1 August 1990
Revised 15 February 1991

The goal of the current paper is to develop a time-accurate incompressible flow calculation methodology based on a Helmholtz pressure equation

$$\Omega(\delta p) - \nabla \cdot \alpha \nabla(\delta p) = -\nabla \cdot \mathbf{u}^{\text{provisional}}, \tag{2}$$

where $\alpha \nabla(\delta p) = \mathbf{u}^{\text{provisional}} - \mathbf{u}^{\text{new}}$ and Ω is the Helmholtz parameter. The function δp can be interpreted as a pressure increment that helps to drive the velocity field to a divergence-free condition.

Straightforward time advancement is not possible for the incompressible flow equations. This is due to a lack of a time-derivative term in the continuity equation and because the appropriate pressure field is linked to the velocity divergence-free condition. The most common methods that are used to solve the unsteady incompressible flow equations involve a Poisson equation for the pressure or a pressure-type variable. The marker-in-cell method (MAC) or the simplified MAC (SMAC)[1] and the projection method[2] are examples of such a method. Pressure correction schemes[3] also involve solving a Poisson-type equation. In the following sections these methods will be referred to as projection methods.

The projection-type schemes are fractional step methods in which the velocity field is decomposed into a field that has a specified divergence and a general field that carries vorticity. The general velocity field that carries vorticity is provisional in nature since it does not necessarily satisfy the divergence-free condition required by the continuity equation. The velocity field that has a specified divergence serves as a velocity correction which helps satisfy the continuity condition. A provisional velocity field is calculated from the momentum equations and then a pressure field is calculated so that the corrected velocity field satisfies the incompressibility condition. If the provisional velocity field is calculated using a time-explicit discretization, additional iterations are not required since the corrected velocity field and pressure field will satisfy the momentum equations and the continuity equation after the correction step.[4] These schemes are only conditionally stable because of their explicit nature. Small time steps are often necessary owing to stability considerations and consequently the application of these schemes to practical problems can require extensive computational effort to integrate the equations to time levels of interest.

Several variants of the projection method have been developed[3,5-7] These methods increase the numerical stability bounds by introducing implicitness into the discretized equations. The discretized governing equations then become a non-linear algebraic system. The simple two-step procedure used for an explicit projection method must usually be supplemented by additional iterations because of the implicit nature of the equations. These additional iterations are required to solve the implicit momentum equations[7] and in some methods additional iterations are required to simultaneously satisfy the momentum and continuity equations.[5] The increased stability bounds and larger time steps that can be used for time integration usually offset any increase in computational time required for the additional iterations. This is the main attraction of implicit-type schemes.

An informative discussion on the theoretical and practical aspects of projection schemes is presented by Gresho.[8]

Whether the discretization is implicit or explicit, a Poisson equation still has to be solved for projection methods. The accurate solution of the Poisson equation is not trivial and is the most time-consuming part of projection algorithms.[9] If the projection method is *completely* implicit in time, then it may be required to solve the pressure Poisson equation several times before advancing to the next time level.

Since multiple iterations are necessary to solve the non-linear algebraic system representing the governing equations, it seems logical to derive an algorithm that will enforce the velocity

divergence condition in an iterative fashion without the need to satisfy a Poisson-type expression. Such a method was developed by Chorin for *steady state* problems.[10] The method introduces artificial or pseudo-compressibility into the continuity equation. An iterative *time-accurate pseudo-compressibility method* is discussed by Peyret.[5] Chorin's method illustrates the technique of using physical principles to modify a governing equation to obtain an equation whose solution may be easier to obtain and yet has the same asymptotic solution as the unmodified equation.

One of the nice things about Poisson-type algorithms is that pressure perturbations are propagated at a speed similar to the incompressible speed of sound. However, the pressure Poisson equation is computationally expensive to solve and it seems inefficient to have to solve it accurately many times whenever a time-implicit scheme is used for the governing equations. On the other hand, the pseudo-compressible pressure equation is very easy to solve (the equation has strong diagonal dominance) but is hyperbolic in nature, so pressure perturbations are propagated at a speed that depends on the amount of pseudo-compressibility. With the introduction of pseudo-compressibility the governing equations become similar to the compressible flow equations (minus the thermal energy equation) and thus exhibit hyperbolic-type behaviour.

Williams[11,12] developed a Helmholtz equation for a pressure-type variable for the solution of *steady* incompressible flows. The principle of the method consists of introducing an amount of pseudo-compressibility into a projection (velocity decomposition) type scheme to obtain an expression for a pressure variable. The key feature of the Helmholtz pressure equation is that it is relatively easy to solve and propagates pressure perturbations at infinite speed.

A time-accurate algorithm suitable for the solution of unsteady incompressible flow problems is presented here. The theory of the Helmholtz pressure equation method is presented. An unsteady flow algorithm based on *implicit* time discretization, successive iteration[5] and the Helmholtz pressure equation is then developed. Key differences between the current method and explicit and semi-implicit projection schemes are discussed. The algorithm is applied to a Taylor problem for which an exact solution is available and to the unsteady laminar flow past a circular cylinder.

HELMHOLTZ PRESSURE EQUATION

The motivation for deriving the Helmholtz pressure equation[11] is to get an equation that is computationally easy to solve like the pseudo-compressible pressure equation yet propagates pressure perturbations at infinite speed like the Poisson-based schemes.

Velocity decomposition methods split the velocity field into two parts: a part that has a specified divergence or dilatation and a part that can carry vorticity. The part that has a specified divergence is a zero-curl vector field and therefore can be written as the gradient of some scalar (i.e. a potential) and can be used to calculate a corrected velocity field that has zero divergence. This decomposition technique will be used in deriving the Helmholtz pressure equation.

A vector field $\nabla(\delta p)$ can be defined such that

$$\mathbf{u}^{m+1} = \mathbf{u} - \alpha \nabla(\delta p), \tag{3}$$

where δp is some potential, α is a 2×2 diagonal matrix that depends on the form of the iterative equation used to solve the momentum equation, m indicates the iteration level and $\mathbf{u}$ is evaluated at a level before $\mathbf{u}^{m+1}$. For the unsteady algorithm presented here $\alpha = \Delta t[I]$, where $[I]$ is a 2×2 identity marix.

The function δp is calculated by taking the divergence of equation (3) and assuming that $\nabla \cdot \mathbf{u}^{m+1} = -b\nabla \cdot \mathbf{u} - \Omega(\delta p)$ rather than zero (traditional projection methods enforce $\nabla \cdot \mathbf{u}^{m+1} = 0$):

$$\Omega(\delta p) - \nabla \cdot \alpha \nabla(\delta p) = -(1+b)\nabla \cdot \mathbf{u}. \tag{4}$$

The amount of corrected compressibility $\nabla \cdot \mathbf{u}^{m+1}$ is referred to as pseudo- or intermediate compressibility. Ω is referred to as the pseudo-diffusion[11] or the Helmholtz parameter and b is the pseudo-compressibility parameter used in Chorin's pseudo-compressible pressure equation.[10] For time-accurate solutions it appears that the Helmholtz parameter should be chosen so that $\Omega \approx \Delta t$. This will help maintain order-of-magnitude consistency with the term $\nabla \cdot \alpha \nabla(\delta p)$ since $\alpha \approx \Delta t$.

In the limit Ω, $b \ll 1$ a Poisson-type pressure equation is obtained; in the limit Ω, $b \gg 1$ the pseudo-compressible pressure equation is obtained.

The expression $\nabla \cdot \mathbf{u}^{m+1} = -b\nabla \cdot \mathbf{u} - \Omega(\delta p)$ can be thought of as some sort of Taylor series expansion for the velocity divergence, but this expression is not unique. An alternative derivation by Williams[11] uses $\nabla \cdot \mathbf{u}^{m+1} = -\Omega(\delta p)$. The only reason for including $b\nabla \cdot \mathbf{u}$ is so that the limiting processes (e.g. Ω, $b \ll 1$) can be clearly seen. In the following sections this term will be neglected and the Helmholtz pressure equation is given as

$$\Omega(\delta p) - \nabla \cdot \alpha \nabla(\delta p) = -\nabla \cdot \mathbf{u}. \tag{5}$$

Equation (5) is still elliptic in character but resembles Helmholtz equations commonly used in acoustic theory. The boundary conditions for the Helmholtz equation are the same as the conditions used for Poisson-type equations,[8] i.e. Neumann boundary conditions.

The set of algebraic equations resulting from the discretization of equation (5) can be written as (assuming Ω is a constant)

$$(\Omega[I] + [A])\{\delta p\} = \{f\}. \tag{6}$$

The standard Poisson equation ($\nabla \cdot \alpha \nabla(\delta p) = \nabla \cdot \mathbf{u}$) is written as

$$[A]\{\delta p\} = \{f\}, \tag{7}$$

where the diagonal elements of the matrix $[A]$ are assumed to be greater than zero. The matrix *spectral radius* norms are ordered as $\|(\Omega[I] + [A])^{-1}\|_{\text{H-press}} < \|[A]^{-1}\|_{\text{Poisson}}$ so an iterative solution of the Helmholtz pressure equation will be easier to obtain than the corresponding Poisson solution. This will be true for $\Omega > 0$. This analysis suggests that an alternative to having Ω be a constant would be to select Ω to be a diagonal matrix whose elements consist of the diagonal of the matrix $[A]$, i.e. $[\Omega] = \text{diag}[A]$. The assumption that the diagonal elements of the matrix $[A]$ have to be positive would then be unnecessary.

The key feature of equation (5) is that only a few iterations are needed to solve it since Ω increases the diagonal dominance of the discretized equations. A pressure equation that is relatively easy to solve is obtained at the expense of having $\nabla \cdot \mathbf{u}^{m+1} \approx -\Omega(\delta p)$ instead of $\nabla \cdot \mathbf{u}^{m+1} = 0$.

The performance of the Helmholtz pressure equation for a steady state laminar flow is shown in Figure 1. This figure dramatically shows that as Ω gets smaller, more time is spent trying to get a solution to the pressure equation. Small Ω corresponds to a Poisson approach. This confirms the observation by Peric[9] that a large proportion of the computational budget is spent solving the pressure equation when a Poisson approach is used. When Ω becomes large, a pseudo-compressible formulation is approached. Very few iterations are required to solve the pressure equation since it becomes strongly diagonalized for large Ω. However, more global iterations are required to obtain a converged solution since the pressure wave magnitude and speed are smaller than

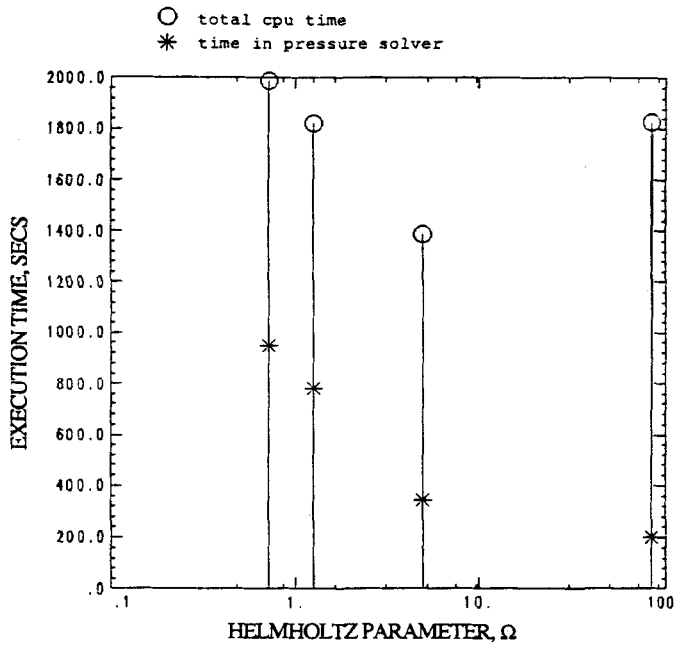

Figure 1. Performance of Helmholtz pressure equation

required. These calculation results are from a driven square cavity laminar flow ($Re=400$) computed with a 51×51 uniform mesh.[11] There is no special significance to the exact values used for Ω; odd-looking values appear in Figure 1 because Ω was scaled with other relaxation parameters used in the computations.

UNSTEADY FLOW ALGORITHM

The algorithm for solving equations (1a) and (1b) can now be formulated as an iterative scheme.[2,5,6,8] A Euler implicit scheme will be used for the current discussion but more elaborate and accurate schemes can be used if desired. The form of the momentum equation discretized in time is

$$(\mathbf{u}^{n,m}-\mathbf{u}^{n})/\Delta t+\partial_x(E-E_v)^{n,m}+\partial_y(F-F_v)^{n,m}+\nabla p^{n,m-1}=S^{n,m-1}, \tag{8}$$

where

$$E^{n,m}=\{u^n u^{n,m},\ u^n v^{n,m}\}^{\mathrm{T}}, \tag{8a}$$

$$u^{n,0}=u^n, \tag{8b}$$

with similar expressions for the other terms. The index n indicates the time level and $t=n\Delta t$; m is the inner iteration or fractional step level, where $m=1, 2, \ldots, M-1$; the counter M indicates the total number of fractional steps or inner iterations, e.g. $u^{n+1}=u^{n,M}$. Non-linear terms are linearized about the time level n. For example, the non-linear term $u^{n+1}u^{n+1}$ arising from the momentum convection terms is written as

$$u^{n+1}u^{n+1}\approx u^n u^{n+1}+(u^{n+1}-u^n)u^n. \tag{9}$$

In the iteration notation this term is written as

$$u^{n+1}u^{n+1} \approx u^n u^{n,m} + (u^{n,m-1} - u^n)u^n. \tag{10}$$

The source term S handles iterative changes in the non-linear convection terms, such as the second term on the right-hand side of equation (10), and can include any other source terms due to time linearization. For example,

$$-S^{n,m-1} = \partial_x\{(u^{n,m-1} - u^n)u^n, (u^{n,m-1} - u^n)v^n\}^{\mathrm{T}} + \partial_y\{(v^{n,m-1} - v^n)u^n, (v^{n,m-1} - v^n)v^n\}^{\mathrm{T}}. \tag{11}$$

The intermediate or provisional velocity $\mathbf{u}^{n,m}$ carries the exact vorticity but does not necessarily satisfy the continuity equation.[6] To help satisfy the continuity equation, the Helmholtz pressure equation is solved

$$\Omega(\delta p) - \nabla \cdot \alpha \nabla(\delta p) = -\nabla \cdot \mathbf{u}^{n,m}, \tag{12}$$

with the corrected velocity field given as

$$\mathbf{u}^{n,m+1} = \mathbf{u}^{n,m} - \alpha \nabla(\delta p), \tag{13}$$

and the next iterate for the pressure is given by

$$\nabla p^{n,m} = \nabla p^{n,m-1} + \omega_{\mathrm{p}} \nabla(\delta p), \tag{14}$$

where ω_{p} is an underrelaxation factor. These equations are solved until norms of $\mathbf{u}^{n,m+1} - \mathbf{u}^{n,m}$ and $\nabla \cdot \mathbf{u}^{n,m}$ satisfy appropriate constraints; then the flow variables can be advanced to the next time level, e.g. $\mathbf{u}^{n+1} = \mathbf{u}^{n,m+1} = \mathbf{u}^{n,M}$. The algorithm is shown in Figure 2. Since $\|\nabla \cdot \mathbf{u}^{n,m+1}\| \approx \|\Omega(\delta p)\|$, a sufficient number of inner iterations must be taken to ensure the velocity divergence condition is satisfied to a specified tolerance.

These equations are solved on a structured finite difference mesh with the flow variables collocated at the node points. The set of conservation equations is formulated in terms of

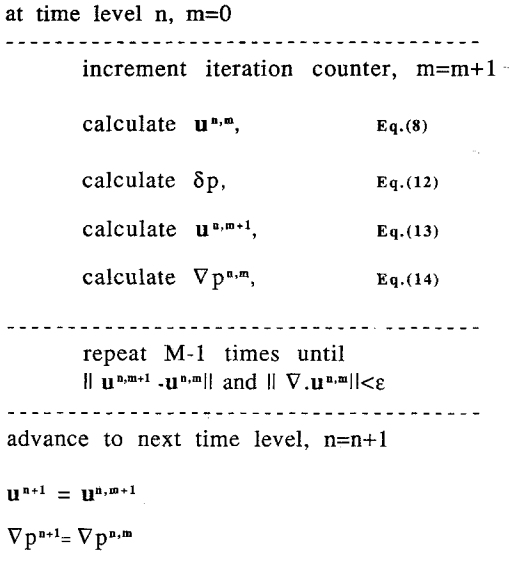

Figure 2. Unsteady flow algorithm

a non-orthogonal generalized co-ordinate system (ξ, η). Second-order centred differencing of the momentum convection terms was used for the following calculations. No artificial dissipation was needed because the Reynolds numbers of the test problems were relatively small.

Each discretized implicit equation is solved by using Stone's iterative elimination method.[13] Finite differences on a structured grid are used so the equation matrices are banded and sparse. Conjugate gradient methods (e.g. Kershaw[14]) were not considered in the current study but can be used as an alternative to Stone's method. If the algebraic equation to be solved for the unknown ψ is given as

$$[A]\{\psi\}=\{R\}, \tag{15}$$

where the vector R is known, then Stone's procedure consists of modifying the matrix $[A]$ by adding a 'small' (matrix spectral norm of N is less than norm of A) matrix $[N]$ so that $[A+N]$ is easily factorized into the lower–upper product $[L_s][U_s]$ and the components of L_s and U_s are calculated in a simple manner[15] and involve less arithmetic than the standard LU decomposition of A. The iterative procedure for solving equation (15) is given as

$$[A+N]\{d\}^r=[L_s][U_s]\{d\}^r=R-[A]\{\psi\}^r, \tag{16}$$

where r is an iteration counter and $d^r=\psi^{r+1}-\psi^r$. The detailed components of L_s and U_s can be found in Reference 13 or 15. Equation (16) is solved until successive iterates satisfy a constraint.

DISCUSSION

Equation (12) assumes that $\nabla\cdot\mathbf{u}^{n,\,m+1}=-\Omega(\delta p)$, so as δp iteratively approaches zero (or a small number relative to the problem at hand) for $m=1,\ldots,M-1$, then the divergence at the inner iteration $(n, m+1)$ also approaches zero; Ω can remain a constant for this process. This is analogous to the time-accurate pseudo-compressibility approach given by Peyret,[5] where Ω plays the role of the pseudo-compressibility parameter which is a constant in Reference 5.

The number of inner iterations, M, can be chosen so that the desired accuracy is achieved. For example, M can be selected to ensure that a norm of the velocity divergence satisfies a constraint as shown in Figure 2.

Explicit projection-type methods require the solution of the momentum equations for a provisional velocity field $\mathbf{u}^{n+1/2}=\mathbf{u}^n+\ldots$ and a Poisson equation for δp, $\nabla\cdot\alpha\nabla(\delta p)=\nabla\cdot\mathbf{u}^{n+1/2}$. The solution of the explicit equation is trivial. Iteration or matrix inversion is not required to obtain the provisional velocities. The computer work required for this segment of calculations is written as W_m. The computer work required to solve the Poisson equation is written as W_p. A matrix inversion must be performed to solve this Poisson equation and so $W_p \gg W_m$. Semi-implicit methods require the solution of a matrix equation for the provisional velocity field. The amount of work required is written as a multiple of the explicit solution work, $c_{si}W_m$, where c_{si} is some number greater than one (the banded matrix corresponding to the semi-implicit momentum equations requires more arithmetic to invert than the diagonal matrix corresponding to the explicit momentum equations). The work required to invert the Poisson equation is again written as W_p. The work for the implicit Helmholtz pressure algorithm is written as $(M-1)(c_{hi}W_m+c_{hp}W_m)$, where $c_{hi}W_m$ is the work required to perform one or two Stone iterations on the algebraic system representing the implicit momentum equations and $c_{hp}W_m$ is the work required for 'several' Stone iterations on the algebraic system representing the Helmholtz pressure equation. 'Several' depends on the size of Ω. Five to 10 iterations will usually be more than sufficient for most flow calculations. A *complete matrix inversion is not performed on these equations*, only several iterations via Stone's method.[13] The factor $M-1$

arises because this procedure must be repeated several times. Multiple inner iterations per time step are required in the current unsteady flow algorithm because of the *implicit* nature of the scheme. The Helmholtz pressure equation and the momentum equations must be solved several times *so that the velocity divergence condition and the momentum equations are simultaneously satisfied at a time level* $n+1$. This is the same idea behind Peyret's[5] implicit scheme.

In general, semi-implicit projection schemes require more work per time step than explicit schemes. However, larger time steps and more robust performance are obtained by using a semi-implicit method. The advantage of a fully implicit scheme is that (in principle) larger time steps can be used and more robust performace is achieved compared to explicit or semi-implicit schemes. This is an important advantage when solving complicated, three-dimensional or turbulent problems. If $(M-1)(c_{hi}W_m+c_{hp}W_m)>(c_{si}W_m+W_p)$, then the Helmholtz implicit algorithm will be more expensive per time level than the semi-implicit scheme. If, however, $(M-1)(c_{hi}W_m+c_{hp}W_m)<(c_{si}W_m+W_p)$, then the Helmholtz implicit algorithm will be less expensive. More specific statements about the speed or efficiency of the unsteady algorithm based on the Helmholtz pressure equation and a time-implicit discretization, relative to other algorithms, can only be made after more experience is gained in the use of the algorithm and careful comparisons with different schemes are made.

It should be noted that a time-implicit scheme based on traditional Poisson projection methods would require $(M-1)(c_{si}W_m+W_p)$ units of work to be performed before advancing to the next time level. Clearly, such a scheme would be expensive because of the number of matrix inversions involved. This is probably why fully time-implicit projection methods based on the solution of a Poisson equation have not been used.

UNSTEADY TEST PROBLEM

An exact solution to the incompressible Navier–Stokes equations for a unit Reynolds number is given by Pearson[16] as

$$u=-\cos(x)\sin(y)\,e^{-2t}, \qquad v=\sin(x)\cos(y)\,e^{-2t},$$
$$\partial_x p=0{\cdot}5\sin(2x)\,e^{-2t}, \qquad \partial_y p=0{\cdot}5\sin(2y)\,e^{-2t}. \tag{17}$$

The performance of the present method is evaluated by solving this test problem (sometimes called a Taylor problem[7]).

The Navier–Stokes equations are numerically solved by the present method in a square domain of dimensions $(\pi/2, \pi/2)$. The computational grid is uniform and has 21×21 node points. Calculations were performed for time steps (Δt) of 0·002, 0·004, 0·01 and 0·02. Ω was set equal to $5\Delta t$ and the pressure underrelaxation factor ω_p was set to 0·9. The evolution of the numerical error as a function of the time step Δt is given in Figure 3. The number of inner iterations per time step used for these calculations was five ($m=1, 2, 3, 4$; $M=5$). The number of inner iterations used was manually (and somewhat arbitrarily) chosen so that the velocity divergence error after the first time step was less than 0·01. The inner iteration process for a time step equal to 0·01 is listed in Table I. This table shows how the numerical error evolves as the inner iteration index cycles from $m=1$ to 4. The numerical error is defined as $\sum|f_{jk}-f''_{jk}|/\sum|f''_{jk}|$, where f_{jk} is the analytical solution defined by equation (17), f''_{jk} is the calculated numerical value and the summation is over all interior nodes. The subscripts jk are node counters. The numerical errors are comparable to the results of Kim and Moin[7] and Braza *et al.*[6] The current algorithm spends about 15% of its computer-processing time solving the pressure equation.

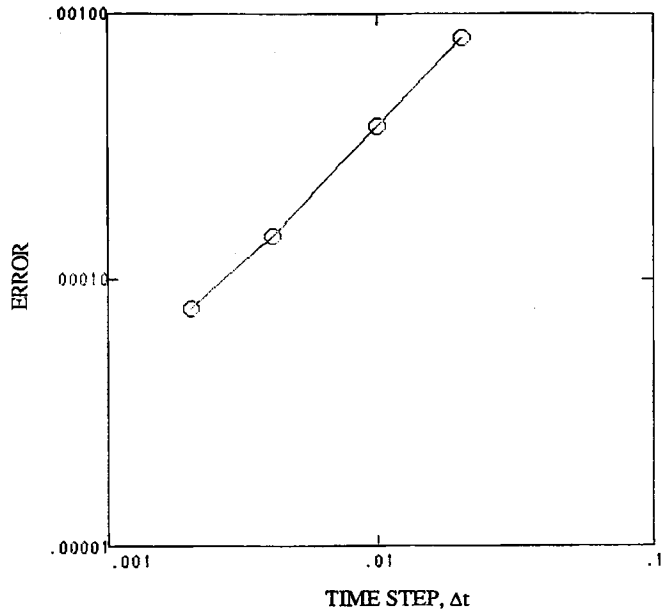

Figure 3. Solution error at $t=0{\cdot}2$ for unsteady test problem

Table I. Listing of iterative process for a time step of 0·01 and $M=5$

n, time level	m, inner iteration	t, time	Error u	Error v	Error $\partial_x p$	Error $\partial_y p$	Error $\nabla\cdot\mathbf{u}$
1	1	0·01000	0·000	0·000	0·000	0·000	0·000
	2		$0{\cdot}120\times10^{-1}$	$0{\cdot}120\times10^{-1}$	$0{\cdot}377\times10^{-2}$	$0{\cdot}383\times10^{-2}$	$0{\cdot}248\times10^{-1}$
	3		$0{\cdot}943\times10^{-2}$	$0{\cdot}950\times10^{-2}$	$0{\cdot}691\times10^{-2}$	$0{\cdot}705\times10^{-2}$	$0{\cdot}192\times10^{-1}$
	4		$0{\cdot}793\times10^{-2}$	$0{\cdot}785\times10^{-2}$	$0{\cdot}922\times10^{-2}$	$0{\cdot}935\times10^{-2}$	$0{\cdot}137\times10^{-1}$
	5		$0{\cdot}697\times10^{-2}$	$0{\cdot}692\times10^{-2}$	$0{\cdot}109\times10^{-1}$	$0{\cdot}110\times10^{-1}$	$0{\cdot}979\times10^{-2}$
2	1	0·02000	$0{\cdot}166\times10^{-1}$	$0{\cdot}166\times10^{-1}$	$0{\cdot}128\times10^{-1}$	$0{\cdot}124\times10^{-1}$	$0{\cdot}703\times10^{-2}$
	2		$0{\cdot}710\times10^{-2}$	$0{\cdot}691\times10^{-2}$	$0{\cdot}106\times10^{-1}$	$0{\cdot}102\times10^{-1}$	$0{\cdot}178\times10^{-1}$
	3		$0{\cdot}537\times10^{-2}$	$0{\cdot}521\times10^{-2}$	$0{\cdot}857\times10^{-2}$	$0{\cdot}826\times10^{-2}$	$0{\cdot}116\times10^{-1}$
	4		$0{\cdot}453\times10^{-2}$	$0{\cdot}446\times10^{-2}$	$0{\cdot}705\times10^{-2}$	$0{\cdot}680\times10^{-2}$	$0{\cdot}846\times10^{-2}$
	5		$0{\cdot}417\times10^{-2}$	$0{\cdot}414\times10^{-2}$	$0{\cdot}599\times10^{-2}$	$0{\cdot}576\times10^{-2}$	$0{\cdot}603\times10^{-2}$
⋮							
19	1	0·19000	$0{\cdot}202\times10^{-1}$	$0{\cdot}202\times10^{-1}$	$0{\cdot}206\times10^{-3}$	$0{\cdot}614\times10^{-3}$	$0{\cdot}598\times10^{-4}$
19	2		$0{\cdot}217\times10^{-3}$	$0{\cdot}123\times10^{-3}$	$0{\cdot}149\times10^{-3}$	$0{\cdot}559\times10^{-3}$	$0{\cdot}365\times10^{-3}$
19	3		$0{\cdot}162\times10^{-3}$	$0{\cdot}162\times10^{-3}$	$0{\cdot}138\times10^{-3}$	$0{\cdot}449\times10^{-3}$	$0{\cdot}392\times10^{-3}$
19	4		$0{\cdot}117\times10^{-3}$	$0{\cdot}992\times10^{-4}$	$0{\cdot}155\times10^{-3}$	$0{\cdot}401\times10^{-3}$	$0{\cdot}117\times10^{-3}$
19	5		$0{\cdot}115\times10^{-3}$	$0{\cdot}102\times10^{-3}$	$0{\cdot}163\times10^{-3}$	$0{\cdot}374\times10^{-3}$	$0{\cdot}747\times10^{-4}$
20	1	0·20000	$0{\cdot}202\times10^{-1}$	$0{\cdot}202\times10^{-1}$	$0{\cdot}207\times10^{-3}$	$0{\cdot}619\times10^{-3}$	$0{\cdot}589\times10^{-4}$
20	2		$0{\cdot}231\times10^{-3}$	$0{\cdot}151\times10^{-3}$	$0{\cdot}152\times10^{-3}$	$0{\cdot}566\times10^{-3}$	$0{\cdot}352\times10^{-3}$
20	3		$0{\cdot}180\times10^{-3}$	$0{\cdot}167\times10^{-3}$	$0{\cdot}145\times10^{-3}$	$0{\cdot}455\times10^{-3}$	$0{\cdot}377\times10^{-3}$
20	4		$0{\cdot}145\times10^{-3}$	$0{\cdot}131\times10^{-3}$	$0{\cdot}161\times10^{-3}$	$0{\cdot}407\times10^{-3}$	$0{\cdot}113\times10^{-3}$
20	5		$0{\cdot}143\times10^{-3}$	$0{\cdot}134\times10^{-3}$	$0{\cdot}169\times10^{-3}$	$0{\cdot}381\times10^{-3}$	$0{\cdot}728\times10^{-4}$

UNSTEADY FLOW PAST A CYLINDER

A practical application of the current method is the analysis of the laminar transient flow past a two-dimensional circular cylinder. A comprehensive experimental study has been carried out by Coutanceau and Bouard;[17] the current calculation results will be compared with their experimental data. The particular case examined here is for a Reynolds number (based on the cylinder diameter) of about 40.

The computational flow domain and boundary conditions are indicated in Figure 4. A non-uniform mesh of 51×41 grid points was used to discretize the flow domain. A time step equal to 0·1 was used. Twenty inner iterations were used for the initial time step. Ten inner iterations per time step were used for the subsequent time steps. This maintained the sum over the interior grid

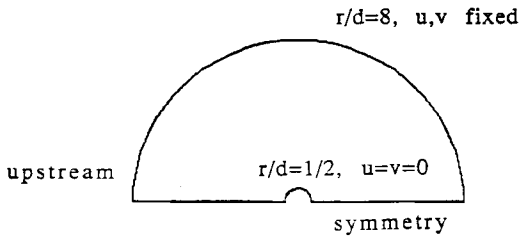

Figure 4. Computational domain for cylinder

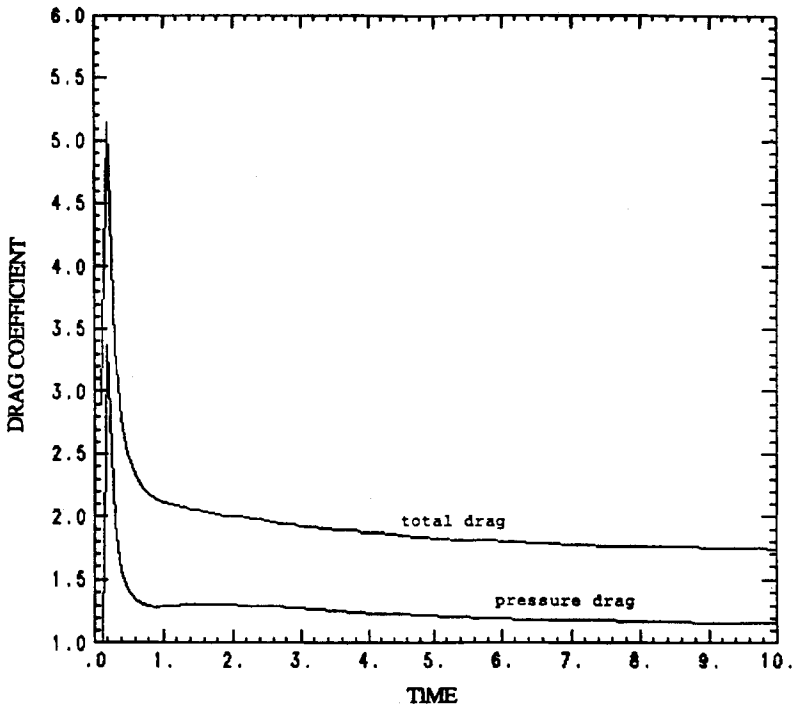

Figure 5. Drag coefficients for cylinder

points of the absolute value of velocity divergence to a level less than 0·05. Changing the number of inner iterations to five instead of 10 (with 20 for the initial time step) did not significantly change the results but the velocity divergence error was slightly higher than 0·05 at the completion of the second time cycle. General-purpose computer codes that use the current algorithm should probably include computer logic to monitor various residuals and automatically vary the number of inner iterations performed at each time cycle so that inner iterations beyond those required to meet some convergence criteria are not taken.

Ω was taken to be 0·1 and $\omega_p = 0{\cdot}95$. The initial flow at $t=0$ was assumed to be the potential flow past a cylinder. The far-field boundary velocities were specified as being equal to the potential flow velocities. In the experiment[17] the cylinder was impulsively started in a container of still fluid.

The pressure and total drag coefficient variation with time is indicated in Figure 5. The drag coefficient is defined as $2D/\rho U_{\text{ref}}^2 L_{\text{ref}}$, where D is the drag force on the cylinder and L_{ref} is the cylinder diameter. The asymptotic pressure and total drag components (1·16 and 1·74 respectively) are slightly higher than the values given by Dennis and Chang[18] (0·998, 1·522). The calculated separation angle at $t=10$ is about 3% higher than the experimental value.

The calculated evolution with time of the cylinder closed-wake length is shown in Figure 6. The calculated results are compared with the experimental data of Coutanceau and Bouard.[17] The agreement is good. A finer grid should be used if it is desired to more accurately resolve the detailed flow structure of the cylinder flow.

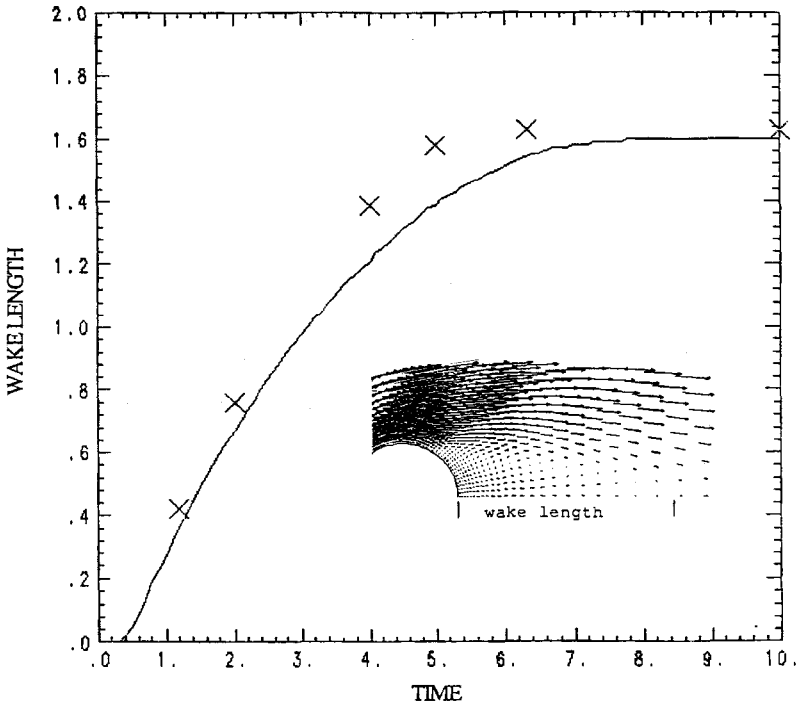

Figure 6. Evolution with time of the cylinder wake: ———, present results; ×, Coutanceau and Bouard[17] ($h=0{\cdot}12$; h is the ratio of the cylinder diameter to the outer boundary diameter)

CONCLUSIONS

A time-accurate, implicit, iterative scheme based on the Helmholtz pressure equation was developed and successfully applied to an unsteady Taylor problem and the unsteady laminar flow past a circular cylinder. The algorithm should be useful for a variety of problems including fluid–structure interaction and unsteady heat transfer, and can be adapted to spectral, finite-element and finite volume Navier–Stokes formulations.

ACKNOWLEDGEMENT

The author would like to thank Dr. P. M. Gresho for his helpful comments and suggestions.

REFERENCES

1. A. A. Amsden and F. H. Harlow, 'A simplified MAC technique for incompressible fluid flow calculations', *J. Comput. Phys.*, **6**, 322–325 (1970).
2. A. J. Chorin, 'Numerical solution of the Navier–Stokes equations', *Math. Comput.*, **22**, 745–762 (1968).
3. S. V. Patankar and D. B. Spalding, 'A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flow', *Int. J. Heat Mass Transfer*, **15**, 1787–1806 (1972).
4. R. Peyret and T. D. Taylor, *Computational Methods for Fluid Flow*, Springer, New York, 1983.
5. R. Peyret, 'Unsteady evolution of a horizontal jet in a stratified fluid', *J. Fluid Mech.*, **78**, 49–63 (1976).
6. M. Braza, P. Chassaing and H. Ha Minh, 'Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder', *J. Fluid Mech.*, **165**, 79–130 (1986).
7. J. Kim and P. Moin, 'Application of a fractional-step method to incompressible Navier–Stokes equations', *J. Comput. Phys.*, **59**, 308–323 (1985).
8. P. M. Gresho, 'On the theory of semi-implicit projection methods for viscous incompressible flow and its implementation via a finite element method that also introduces a nearly-consistent mass matrix. Part 1: Theory', *Int. j. numer. methods fluids*, **11**, 587–620 (1990).
9. M. Péric, 'Efficient semi-implicit solving algorithm for nine diagonal coefficient matrix', *Numer. Heat Transfer*, **11**, 251–279 (1987).
10. A. Chorin, 'A numerical method for solving incompressible viscous flow problems', *J. Comput. Phys.*, **2**, 12–26 (1967).
11. M. Williams, 'A method for the calculation of incompressible viscous flows', *Numer. Heat Transfer, Part B: Fundamentals*, in the press.
12. M. Williams, 'An incompressible flow solver with application to several three-dimensional flow problems', *AIAA Paper 1494*, 1990.
13. H. L. Stone, 'Iterative solution of implicit approximation of multi-dimensional partial differential equations', *SIAM J. Numer. Anal.*, **5**, 530–558 (1968).
14. D. S. Kershaw, 'The incomplete Cholesky–conjugate gradient method for the iterative solution of systems of linear equations', *J. Comput. Phys.*, **26**, 43–65 (1978).
15. G. D. Smith, *Numerical Solution of Partial Differential Equations*, 2nd edn, Oxford University Press, Oxford, 1978, pp. 268–274.
16. C. E. Pearson, 'A computer method for viscous flow problems', *J. Fluid Mech.*, **21**, 611–622 (1966).
17. M. Coutanceau and R. Bouard, 'Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation, Part 2, Unsteady flow', *J. Fluid Mech.*, **79**, 257–272 (1977).
18. S. C. R. Dennis and G. Z. Chang, 'Numerical solutions for steady flow past a circular cylinder at Reynolds numbers up to 100', *J. Fluid Mech.*, **42**, 471–489 (1973).