Originals

Computational Mechanics 20 (1997) 541–550 © Springer-Verlag 1997

A semi-implicit fractional step finite element method for viscous incompressible flows

P. Kjellgren

Abstract This paper describes a new semi-implicit finite element algorithm for time-dependent viscous incompressible flows. The algorithm is of a general type and can handle both low and high Reynolds number flows, although the emphasis is on convection dominated flows. An explicit three-step method is used for the convection term and an implicit trapezoid method for the diffusion term. The consistent mass matrix is only used in the momentum phase of the fractional step algorithm while the lumped mass matrix is used in the pressure phase and in the pressure Poisson equation. An accuracy and stability analysis of the algorithm is provided for the pure convection equation. Two different types of boundary conditions for the end-of-step velocity of the fractional step algorithm have been investigated.

Numerical tests for the lid-driven cavity at $Re = 1$ and $Re = 7500$ and flow past a circular cylinder at $Re = 100$ are presented to demonstrate the usefulness of the method.

1 Introduction

The finite element method is powerful and convenient to use for analyzing flows in complex geometries. However, just as in the case with finite differences, the presence of convective terms in the Navier-Stokes' equations causes serious numerical difficulties, appearing in the form of "wiggles" (oscillatory solutions). This could be solved somewhat heuristically by using upwinding. Unfortunately, traditional upwinding is often found to be too dissipative. More refined versions of the upwinding concept are the methods that utilize upwinding only or mainly in the streamline direction, e.g. the Streamline Upwind/Petrov Galerkin (SUPG) formulation of Hughes and Brooks (1979) and Brooks and Hughes (1982), the streamline balancing diffusion formulation of Kelly et al. (1980), the Balancing Tensor Diffusivity (BTD) formulation of Gresho et al. (1984) and the Taylor-Galerkin method of Donea (1984). The explicit and semi-implicit two-step methods of Hawken et al (1990) and the explicit three-step method of Jiang and Kawahara (1993) have also been used with success.

Another important issue when solving the fluid equations is computational efficiency. Since the momentum and continuity equations governing the fluid are coupled, the straight-forward implementation would require simultaneous solution of all these and since the continuity equation is of elliptic nature, it has to be solved in an implicit way. This approach, though correct and easy to implement, would lead to a rather time consuming formulation. In order to improve efficiency, Chorin (1968) introduced a time splitting or fractional step approach, in which the continuity equation was decoupled from the momentum equations. The momentum equations can usually also be decoupled from each other. The fractional step approach was later adopted in the finite element area by Donea et al. (1982). The fractional step method is an excellent basis for an efficient algorithm and is conveniently combined with one of the methods above.

In a semi-implicit method, the diffusive terms are solved implicitly and the convective terms explicitly. This is a very attractive formulation since if the diffusion terms are solved with an explicit numerical scheme there will usually be a severe restriction on the size of the timestep. This will not really cause any trouble for laminar high Reynolds number flows, where the contribution from the diffusion terms is small. However, if one really wants to analyze high Reynolds number flows, it is usually necessary to use some kind of turbulence model that gives rise to a turbulent viscosity, and it is the effective viscosity, i.e. the turbulent viscosity plus the molecular viscosity, that will restrict the size of the timestep in an explicit scheme. Since the effective viscosity may be rather large and since it is not known beforehand, it is better and more reliable to use an implicit scheme on the diffusion terms when analyzing turbulent flows, thus avoiding the diffusive timestep restriction altogether. Based on the fractional step approach, a three-step semi-implicit finite element method is proposed in the present study.

One of the main purposes of the developed computational code is to simulate three-dimensional, unsteady turbulent flows using a large eddy simulation (LES) turbulence model. This requires an accurate and efficient algorithm and when constructing that it might be interesting to use higher order elements. However, preliminary numerical tests with a biquadratic velocity/bilinear

Communicated by S. N. Atluri, 5 June 1997

P. Kjellgren
Department of Aeronautics, Royal Institute of Technology
S-100 44 Stockholm, Sweden

A significant part of this research was performed at the University of Tokyo, Dept. of Quantum Eng. and Systems Science, under the supervision of Prof. Genki Yagawa. This was made possible by a Monbusho (Japanese Ministry of Education) scholarship granted to the author. The author is very thankful to Prof. Yagawa and the Japanese Monbusho for their support. The author is also very grateful to Prof. Ulf Ringertz, Dept. of Aeronautics, The Royal Institute of Technology, for his support and help.

pressure mixed order interpolation element for unsteady two-dimensional channel flow showed poor results, see Kjellgren et al. (1994), and only the bilinear velocity/constant pressure has been used here. To improve accuracy, especially when dealing with flows in the high Reynolds number range, it is advantageous to use the consistent mass matrix, see Gresho et al. (1978). However, the fractional step approach requires the solution of a Poisson equation for the pressure terms and the pressure stiffness matrix includes the inverse of the mass matrix. Thus, the pressure stiffness matrix with a fully consistent mass matrix would be unnecessary expensive and complicated to construct. To overcome this, the present approach has used the "nearly consistent" mass matrix formulation of Gresho (1990) which uses the consistent mass matrix only in the momentum equation part of the fractional step algorithm.

2
Governing equations

The governing equations written in the Eulerian form for time-dependent, incompressible, viscous flow fields in a bounded domain Ω are

$$\frac{\partial u_i}{\partial x_i} = 0 \ , \tag{2.1}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = \frac{\partial \tau_{ij}}{\partial x_j} \ , \tag{2.2}$$

where u_i is the ith component of the velocity vector and τ_{ij} is the total stress per unit density of the fluid. In the case of an incompressible, Newtonian fluid, the constitutive equation is given by

$$\tau_{ij} = -p\delta_{ij} + 2\nu e_{ij} \ , \tag{2.3}$$

where

$$\mathrm{e}_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{2.4}$$

is the deformation rate tensor, p is the kinematic pressure (pressure divided by density), δ_{ij} is the Kronecker delta and ν is the kinematic viscosity. These equations are supplemented by the Dirichlet boundary conditions

$$u_i = \hat{u}_i \text{ on } \Gamma_1 \ , \tag{2.5}$$

and the Neumann boundary conditions

$$\tau_{ij} n_j = \hat{t}_i \text{ on } \Gamma_2 \ , \tag{2.6}$$

where $\Gamma_1 \cap \Gamma_2 = \varnothing$ (the empty set) and $\Gamma_1 \cup \Gamma_2 = \Gamma$, Γ is the boundary of Ω, $\hat{\mathbf{u}}$ represents the velocity vector prescribed on Γ_1, $\hat{\mathbf{t}}$ is the traction vector prescribed on Γ_2 and $\mathbf{n}$ is the outward unit normal vector to Γ. The initial condition consists of specifying the value of the velocity field $\mathbf{u}$ at the initial time,

$$\mathbf{u}(\mathrm{x}, 0) = \mathbf{u}_0(\mathrm{x}) \text{ in } \Omega \ , \tag{2.7}$$

such that the incompressibility condition at the initial time is satisfied,

$$\frac{\partial (u_0)_i}{\partial x_i} = 0 \ . \tag{2.8}$$

Finally, if $\Gamma_1 = \Gamma$ and $\Gamma_2 = \varnothing$, the global conservation of mass requires that the prescribed velocity $\hat{\mathbf{u}}$ satisfies

$$\oint_\Gamma \hat{u}_i n_i \, \mathrm{d}\Gamma = 0 \ . \tag{2.9}$$

3
Spatial discretization

The spatial discretization is performed with a finite element method based on the weak formulation of (2.1)–(2.4), giving

$$\begin{aligned} &\int_\Omega w^l \frac{\partial u_i}{\partial t} \, \mathrm{d}\Omega + \int_\Omega w^l u_j \frac{\partial u_i}{\partial x_j} \, \mathrm{d}\Omega \\ &\quad - \int_\Omega \frac{\partial w^l}{\partial x_i} p \, \mathrm{d}\Omega + \int_\Omega \nu \frac{\partial w^l}{\partial x_j}\frac{\partial u_i}{\partial x_j} \, \mathrm{d}\Omega \\ &\quad = \int_{\Gamma_2} w^l \left(\nu \frac{\partial u_i}{\partial x_j} - p\delta_{ij}\right) n_j \, \mathrm{d}\Gamma_2 \ , \end{aligned} \tag{3.1}$$

$$\int_\Omega w^c \frac{\partial u_i}{\partial x_i} \mathrm{d}\Omega = 0 \ , \tag{3.2}$$

where w^l is a bilinear weighting function and w^c is a constant weighting function and the terms for pressure and diffusion have been partially integrated. Using the ordinary Bubnov-Galerkin method results in the following matrix equations for (3.1)–(3.2),

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{Q}(\mathbf{u})\mathbf{u} + \mathbf{D}\mathbf{u} - \mathbf{H}p = \mathbf{f} \ , \tag{3.3}$$

$$\mathbf{H}^T\mathbf{u} = 0 \ , \tag{3.4}$$

where $\mathbf{M}$, $\mathbf{Q}$, $\mathbf{D}$, $\mathbf{H}$ and $\mathbf{H}^T$ represent the mass, convection, diffusion, gradient and divergence matrices and $\mathbf{f}$ is a vector that represents external forces and/or natural boundary conditions. The element matrices corresponding to these are defined as

$$\mathbf{M}^e_{\alpha\beta} = \int_{\Omega^e} N_\alpha N_\beta \, \mathrm{d}\Omega \ , \tag{3.5}$$

$$\mathbf{Q}^e_{\alpha\beta} = \int_{\Omega^e} N_\alpha N_\gamma \mathbf{u}_{j\gamma} \frac{\partial N_\beta}{\partial x_j} \, \mathrm{d}\Omega \ , \tag{3.6}$$

$$\mathbf{D}^e_{\alpha\beta} = \int_{\Omega^e} \frac{\partial N_\alpha}{\partial x_j}\frac{\partial N_\beta}{\partial x_j} \, \mathrm{d}\Omega \ , \tag{3.7}$$

$$\mathbf{H}^e_{\alpha i} = \int_{\Omega^e} \frac{\partial N_\alpha}{\partial x_i} \Phi \, \mathrm{d}\Omega \ , \tag{3.8}$$

where N_α is a piecewise bilinear interpolation function for velocity and Φ is a piecewise constant interpolation function for pressure.

4
The fractional step method

The purpose of using a fractional step method is to reduce the cost of a simulation by decoupling the velocities and the pressure. This can be done in several different ways and in the method used here the calculation is carried out in three steps. First, an intermediate velocity vector field $\mathbf{u}^*$ is derived from the momentum equation with the pressure terms omitted,

$$u_i^* = u_i^n + \text{convection} + \text{diffusion} \ , \tag{4.1}$$

where the exact form of the convection and diffusion terms will be treated later on.

By choosing different algorithms for the convection and diffusion terms, explicit and semi-implicit algorithms can be derived. Then in the second step the end-of-step velocity $\mathbf{u}^{n+1}$ is formed by adding the pressure terms to the intermediate velocity $\mathbf{u}^*$,

$$u_i^{n+1} = u_i^* - \Delta t \frac{\partial p^{n+1}}{\partial x_i} \ . \tag{4.2}$$

The pressure is in turn calculated from a Poisson equation which is derived by requiring that the end-of-step velocity $\mathbf{u}^{n+1}$ satisfies the incompressibility condition

$$\frac{\partial u_i^{n+1}}{\partial x_i} = \frac{\partial u_i^*}{\partial x_i} - \frac{\partial^2 p^{n+1}}{\partial x_i \partial x_i} = 0 \ . \tag{4.3}$$

The velocity boundary condition (2.5) has been used for the intermediate velocity $\mathbf{u}^*$ at the first step, while two different types of boundary conditions have been investigated for the end-of-step velocity. In the first one, here called "type 1" boundary condition, the end-of-step velocity $\mathbf{u}^{n+1}$ is only required to satisfy the velocity boundary condition normal to Γ_1, which implies a Neumann boundary condition on the Poisson equation (4.3), giving

$$\frac{\partial p^{n+1}}{\partial n} = \frac{u_i^{n+1} - u_i^*}{\Delta t} \cdot \hat{n}_i = 0 \text{ on } \Gamma_1 \ . \tag{4.4}$$

The discrete form of (4.2) using the gradient matrix defined by (3.8) and also the inverse of the row-sum lumped mass matrix $\mathbf{M}_d^{-1}$ can be written as

$$\mathbf{u}^{n+1} = \mathbf{u}^* + \Delta t \mathbf{M}_d^{-1}(\mathbf{I} - \eta\eta^T)\mathbf{H}p^{n+1} + \Delta t \mathbf{M}_d^{-1}\eta\eta^T\mathbf{H}p^{n+1} \ , \tag{4.5}$$

where η is the matrix with components of the normal vectors corresponding to nodes on Γ_1. The pressure gradient term on the right hand side of (4.5) has been split up into two terms; one pressure gradient term tangential to Γ_1 and one pressure gradient term normal to Γ_1. The pressure gradient term normal to Γ_1, $\mathbf{M}_d^{-1}\eta\eta^T\mathbf{H}p^{n+1}$, is set to zero which automatically satisfies the boundary condition for the end-of-step velocity since no contribution can then be given to u_i^{n+1} on Γ_1 in the normal direction and thus, $u_i^{n+1}n_i = u_i^* n_i = \hat{u}_i n_i$ on Γ_1. Setting the divergence matrix to $\mathbf{C}$, where $\mathbf{C} = (\mathbf{I} - \eta\eta^T)\mathbf{H}$, and applying it to (4.5) leads to the discrete formulation of the pressure equation and the end-of-step velocity,

$$\mathbf{C}^T\mathbf{M}_d^{-1}\mathbf{C}_p = -\frac{\mathbf{H}^T\mathbf{u}^*}{\Delta t} \ , \tag{4.6}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* + \Delta t \mathbf{M}_d^{-1}\mathbf{C}p^{n+1} \ . \tag{4.7}$$

The type 1 boundary condition can be said to be a generalization of the procedure given by Donea et al. (1982), i.e. the degrees of freedom corresponding to the prescribed normal components of the velocity are excluded from the pressure gradient matrix. It will also result in the same pressure Poisson equation matrix and pressure gradient terms if the normal directions of Γ_1 coincide with the global coordinate system. One advantage with this formulation is that it avoids the spurious spatial oscillations of pressure known as "checkerboarding" and it produces pressure equations of the "right form", see Donea et al. (1981). Since the implementation used here is based on the normal vectors to Γ_1, it can easily be used in three dimensions, which would not have been the case had it been based on the tangential vectors to Γ_1.

In an even simpler version of the boundary condition to the pressure Poisson equation, here called "type 2" boundary condition, the degrees of freedom corresponding to all prescribed components of the velocity are excluded from the pressure gradient terms, which means that the full velocity boundary conditions on Γ_1 are used both for the intermediate velocity and the end-of-step velocity.

In both cases, the pressure matrix is symmetric and can be solved efficiently by an ordinary linear equation solver for symmetric matrices. In all examples shown here, the diagonally scaled preconditioned conjugate gradient method has been used to solve the pressure equation.

It should be mentioned that the time splitting errors arising from the fractional step approach have not been analyzed here. For an analysis of these, and also other boundary conditions for the intermediate velocity, end-of-step velocity and the Poisson equation, see Gresho (1990), Gresho and Sani (1987), Shen (1993) and Karniadakis et al. (1991).

5
The convection and diffusion phase

The convection term has been discretized with a method similar to the three-step method of Jiang and Kawahara (1993), that is

$$u^{n+\frac{1}{3}} = u^n + \tfrac{1}{3}\Delta t f(u^n) \ , \tag{5.1}$$

$$u^{n+\frac{1}{2}} = u^n + \tfrac{1}{2}\Delta t f(u^{n+\frac{1}{3}}) \ , \tag{5.2}$$

$$u^{n+1} = u^n + \Delta t f(u^{n+\frac{1}{2}}) \ , \tag{5.3}$$

where $f(u^n)$ is the convection terms at time $t = n\Delta t$. To see how this method works, it is helpful to look at the pure convection equation in one dimension

$$\frac{\partial u(x,t)}{\partial t} + a\frac{\partial u(x,t)}{\partial x} = 0 \ , \tag{5.4}$$

where a is a constant transport velocity. Using (5.1)–(5.3) with

$$f(u^n(x)) = -a\frac{\mathrm{d}u^n(x)}{\mathrm{d}x} \ , \tag{5.5}$$

gives

$$\begin{aligned} u^{n+1} &= u^n - a\Delta t \frac{\mathrm{d}}{\mathrm{d}x}\left(u^n - \frac{1}{2}a\Delta t\frac{\mathrm{d}}{\mathrm{d}x}\left(u^n - \frac{1}{3}a\Delta t\frac{\mathrm{d}u^n}{\mathrm{d}x}\right)\right) \\ &= u^n - a\Delta t\frac{\mathrm{d}u^n}{\mathrm{d}x} + \frac{1}{2}(a\Delta t)^2\frac{\mathrm{d}^2u^n}{\mathrm{d}x^2} \\ &\quad - \frac{1}{6}(a\Delta t)^3\frac{\mathrm{d}^3u^n}{\mathrm{d}x^3} + O[(\Delta t)^4] \ . \end{aligned} \tag{5.6}$$

Since the solutions to (5.4) at two successive time levels n and $n+1$ satisfies the relation

$$u^{n+1}(x) = u^n(x - a\Delta t) \ , \tag{5.7}$$

a Taylor expansion of (5.7) around the point x yields

$$u^{n+1} = u^n - a\Delta t\frac{\mathrm{d}u^n}{\mathrm{d}x} + \frac{1}{2}(a\Delta t)^2\frac{\mathrm{d}^2u^n}{\mathrm{d}x^2} - \frac{1}{6}(a\Delta t)^3\frac{\mathrm{d}^3u^n}{\mathrm{d}x^3} + O[(\Delta t)^4] \ . \tag{5.8}$$

Thus, the numerical procedure in (5.1)–(5.3) gives a third order approximation to the linear convection equation. Applying the three-step method on the non-linear convection term and including the contribution from the diffusion term into the calculation of the intermediate velocity field gives the complete calculation of the intermediate velocity field as follows

$$u_i^{n+\frac{1}{3}} = u_i^n - \frac{1}{3}\Delta t u_j^n\frac{\partial u_i^n}{\partial x_j} \ , \tag{5.9}$$

$$u_i^{n+\frac{1}{2}} = u_i^n - \frac{1}{2}\Delta t u_j^{n+\frac{1}{3}}\frac{\partial u_i^{n+\frac{1}{3}}}{\partial x_j} \ , \tag{5.10}$$

$$u_i^* = u_i^n - \Delta t u_j^{n+\frac{1}{2}}\frac{\partial u_i^{n+\frac{1}{2}}}{\partial x_j} + \nu\Delta t\frac{\partial^2}{\partial x_j x_j}(\beta u_i^* + (1-\beta)u_i^n) \ . \tag{5.11}$$

The implicitness in the numerical scheme is introduced through the choice of β. $\beta = 0$ gives the first order accurate (for the diffusion term) explicit forward Euler method, $\beta = 1$ gives the first order accurate implicit backward Euler method and $\beta = 0.5$ gives the second order accurate implicit trapezoid method for the diffusion term. The main advantage with $\beta = 0$ is that it allows a fully explicit (and thus fast) solution of (5.11) if used together with a lumped mass matrix. The disadvantage is the restriction on the timestep Δt due to the viscous limit. However, for $\beta \geq 0.5$ there are no restrictions on Δt due to the viscous limit. The discrete forms of (5.9)–(5.11) become

$$\mathbf{M}(\mathbf{u}^{n+\frac{1}{3}} - \mathbf{u}^n) = \tfrac{1}{3}\Delta t\mathbf{Q}(\mathbf{u}^n)\mathbf{u}^n \tag{5.12}$$

$$\mathbf{M}(\mathbf{u}^{n+\frac{1}{2}} - \mathbf{u}^n) = \tfrac{1}{2}\Delta t\mathbf{Q}(\mathbf{u}^{n+\frac{1}{3}})\mathbf{u}^{n+\frac{1}{3}} \tag{5.13}$$

$$(\mathbf{M} + \beta\Delta t\nu\mathbf{D})(\mathbf{u}^* - \mathbf{u}^n) = \Delta t(\mathbf{Q}(\mathbf{u}^{n+\frac{1}{2}})\mathbf{u}^{n+\frac{1}{2}} + \mathbf{D}\mathbf{u}^n + \mathbf{f}^n) \ . \tag{5.14}$$

At each of these steps, a linear system of the form $\mathbf{A}\mathrm{x} = \mathbf{b}$ has to be solved. This is done efficiently by the procedure used by Donea (1984) and later refined by Zienkiewicz (1985) and Ding et al. (1992). In this procedure, the linear system is solved iteratively by a generalized Jacobi solver

$$\mathrm{x}^{r+1} = (\mathbf{I} - \omega\mathbf{A}_d^{-1}\mathbf{A})\mathrm{x}^r + \omega\mathbf{A}_d^{-1}\mathbf{b} \ , \tag{5.15}$$

where $\mathbf{A}_d^{-1}$ is a diagonalized version of $\mathbf{A}$, usually the lumped mass matrix $\mathbf{M}_d^{-1}$ or the main diagonal of $\mathbf{A}$ and ω is a positive relaxation factor. If $\mathbf{M}_d^{-1}$ is the inverse of the lumped mass matrix and $\omega = 1$, then one iteration of (5.15) leads to the ordinary explicit lumped mass approach. Usually as few as two or three iterations are sufficient, which means that the extra time needed to do the iterations is small compared to the total solution time.

In practice, it makes sense to have two different kinds of algorithms: a fully explicit algorithms with $\beta = 0$ and lumped mass matrix and a semi-implicit algorithm with $\beta = 0.5$ and consistent mass matrix. Both of these have been investigated and results are given in the numerical examples section.

The convection matrix $\mathbf{Q}(\mathbf{u})$ is non-linear and must be updated at every timestep. This would take considerable time if done with the ordinary 2×2 ($2 \times 2 \times 2$ in 3 dimensions) Gauss quadrature and therefore an one-point quadrature method similar to the one used by Gresho et al. (1984) has been used for $\mathbf{Q}(\mathbf{u})$.

6
Accuracy and stability

The main purpose with this work was to develop a general, efficient, accurate and reliable algorithm with emphasis on high Reynolds number flows that still maintains a simple algorithmic structure and which is not hampered by the diffusive timestep limit. Thus, the analysis given here is limited to the linear convection Eq. (5.4).

The exact solution to the linear convection equation with the initial condition that one Fourier mode with wavenumber k and amplitude ρ is excited, i.e. $u(x,0) = \rho\mathrm{e}^{\mathrm{i}kx}$, is given by $u(x,t) = \rho\mathrm{e}^{\mathrm{i}k(x-at)}$, which gives the exact solution at node j on an evenly spaced mesh as $u_j(t) = \rho\mathrm{e}^{\mathrm{i}k(x_j-at)} = \rho\mathrm{e}^{\mathrm{i}k((j-1)\Delta x-at)}$. The solution at node j at time $t = t_1$ can be viewed as an amplification factor g_{ex} times the solution at node j at a previous time $t = t_0$, that is, $u_j(t_1) = g_{\mathrm{ex}}u_j(t_0)$. Thus, the *exact* amplification factor is given by

$$g_{\mathrm{ex}} = \frac{u_j(t_1)}{u_j(t_0)} = \mathrm{e}^{-\mathrm{i}ka\Delta t} = \mathrm{e}^{-\mathrm{i}\Theta C_r} \ , \tag{6.1}$$

where $\Delta t = t_1 - t_0$, $\Theta = k\Delta x$ and $C_r = a\Delta t/\Delta x$ is the Courant number. The numerical method gives a numerical amplification factor g_{num}, and generally, $g_{\mathrm{num}} \neq g_{\mathrm{ex}}$. The relation between the numerical and exact amplification factors give important information about amplitude accuracy $|g_{\mathrm{num}}|/|g_{\mathrm{ex}}|$ and phase accuracy $\arg(g_{\mathrm{num}})/\arg(g_{\mathrm{ex}})$.

The total amplification factor $g(\Theta, C_r)$ for the three substeps (5.12)–(5.14) applied to the pure convection equation becomes

$$g_{\mathrm{num}}(\Theta, C_r) = 1 + (g_{\mathrm{num}}(\Theta, C_r)^{n+1} - 1) \times (1 + (g_{\mathrm{num}}(\Theta, C_r)^{n+\frac{1}{2}} - 1) \times g_{\mathrm{num}}(\Theta, C_r)^{n+\frac{1}{3}}) \ , \tag{6.2}$$

where $g_{\mathrm{num}}(\Theta, C_r)^{n+(\cdot)}$ is the numerical amplification factor at substep $n + (\cdot)$. The discrete equation for the first substep using linear finite elements and lumped mass matrix becomes

$$\frac{u_j^{n+\frac{1}{3}} - u_j^n}{1/3\Delta t}\Delta x + a\tfrac{1}{2}(u_{j+1}^n - u_{j-1}^n) = 0 \ . \tag{6.3}$$

Using the relations $u_{j+1}^n = u_j^n e^{i\Theta}$ and $u_{j-1}^n = u_j^n e^{-i\Theta}$ gives the numerical amplification factor for the first substep

$$g_{\text{num}}(\Theta, C_r)^{n+\frac{1}{3}} = 1 - i\tfrac{1}{3}C_r \sin(\Theta) \quad . \tag{6.4}$$

Similarly, the numerical amplification factor at substep $n + (\cdot)$ is

$$g_{\text{num}}(\Theta, C_r)^{n+(\cdot)} = 1 - i(\cdot)C_r \sin(\Theta) \tag{6.5}$$

in the case of a (row-sum) lumped mass matrix, and

$$g_{\text{num}}(\Theta, C_r)^{n+(\cdot)} = 1 - i(\cdot)C_r \frac{3\sin(\Theta)}{\cos(\Theta) + 2} \tag{6.6}$$

in the case of a consistent mass matrix. This results in the total numerical amplification factor

$$g_{\text{lump}}(\Theta, C_r) = 1 - \tfrac{1}{2}C_r^2 \sin^2(\Theta) + i(-C_r \sin(\Theta) + \tfrac{1}{6}C_r^3 \sin^3(\Theta)) \tag{6.7}$$

with the lumped mass matrix, and

$$g_{\text{cons}}(\Theta, C_r) = 1 - \frac{1}{2}C_r^2 \left(\frac{3\sin(\Theta)}{\cos(\Theta) + 2}\right)^2 + i\left(-C_r \frac{3\sin(\Theta)}{\cos(\Theta) + 2} + \frac{1}{6}C_r^3 \left(\frac{3\sin(\Theta)}{\cos(\Theta) + 2}\right)^3\right) \tag{6.8}$$

with the consistent mass matrix. Amplitude and phase accuracy of these at various Courant numbers are shown in Fig. 1 and Fig. 2. The values on the vertical axes correspond to the amplitude and phase accuracies and the values on the horizontal axes correspond to Θ/π. The amplitude accuracy is fairly uniform over the range of wavenumbers but the phase accuracy decreases rapidly for high wavenumbers and becomes zero at $\Theta = k\Delta x = \pi$,

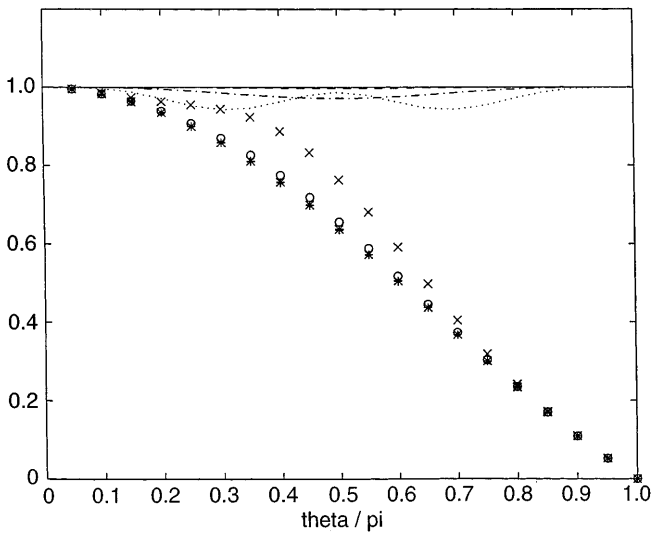

Fig. 1. Amplitude and phase accuracies with lumped mass matrix. Amplitude accuracy: (—)$C_r = 0.1$, (---)$C_r = 0.5$, $(- \cdot -)C_r = 1.0$, $(\cdots)C_r = 1.7$. Phase accuracy: (+) $C_r = 0.1$, (*)$C_r = 0.5$, (o)$C_r = 1.0$, (x)$C_r = 1.7$

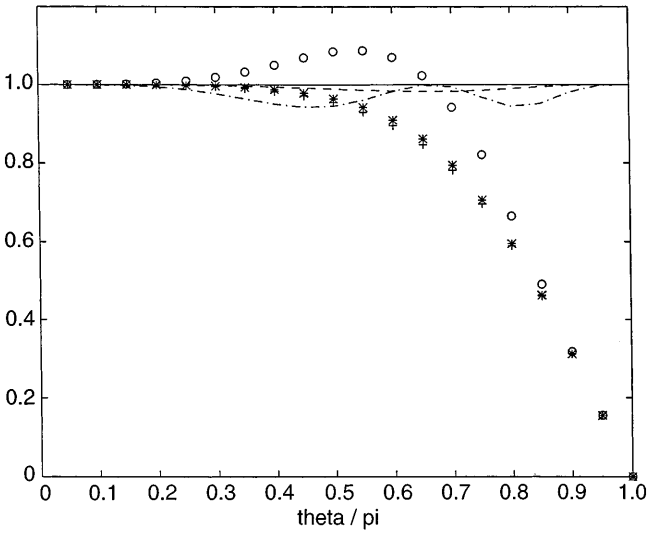

Fig. 2. Amplitude and phase accuracies with consistent mass matrix. Amplitude accuracy: (—)$C_r = 0.1$, (---)$C_r = 0.5$, $(- \cdot -)$ $C_r = 1.0$. Phase accuracy: (+) $C_r = 0.1$, (*) $C_r = 0.5$, (o)$C_r = 1.0$

which corresponds to the wavelength $\lambda = 2\Delta x$, i.e. the shortest wavelength that can be resolved on the mesh. Notice that usage of the consistent mass matrix significantly improves the phase accuracy at high wavenumbers and thus, it is advantageous to use the consistent mass matrix in flows which contains small scales (small wavelengths), i.e. high Reynolds number flows. This improved accuracy does not come for free. In order for the numerical solution not to grow out of control, $|g| \leq 1$ must hold. An analysis of $|g_{\text{lump}}|$ gives the stability requirement $C_r \leq \sqrt{3}$, while an analysis of $|g_{\text{cons}}|$ gives the stability requirement $C_r \leq 1$. Thus, stability requires a smaller timestep Δt when the consistent mass matrix is used. Numerical tests suggest that these limits on the Courant number are valid when solving the Navier-Stokes' equations with the explicit method. However, much larger Courant numbers, up to $C_r \sim 5$, can usually be used with the semi-implicit algorithm without causing instability, which has also been reported by Gresho (1990).

It is interesting to note that the numerical amplification factors for the three substeps are all unconditionally unstable for the linear convection equation when used alone, since $|g_{\text{num}}(\Theta, C_r)| > 1$ for all Θ except the trivial cases $\Theta = 0$ and $\Theta = \pi$ but that these substeps form a conditionally stable method when used together.

The time accuracy for long wavelengths (large scales), i.e. in the asymptotic limit $\Theta \to 0$, can be assessed from the numerical amplification factors (6.7), (6.8), which are reduced to

$$g_{\text{num}}(\Theta, C_r) \sim 1 - iC_r\Theta - \tfrac{1}{2}C_r^2\Theta^2 + \tfrac{1}{6}iC_r^3\Theta^3 + O(C_r^4\Theta^4) \quad . \tag{6.9}$$

Comparing this with the Taylor series of the exact amplification factor $e^{-iC_r\Theta}$ gives that the three-step method of above is third-order accurate in time ($C_r\Theta = a\Delta tk$) for small wavenumbers k both when lumped and consistent mass matrices are used.

7
Numerical examples

In this section two examples which demonstrate the generality and accuracy of the methods described in this paper are demonstrated. The first example, the "classic" lid-driven cavity, demonstrates that flows with a wide range of Reynolds number can be solved and the second example, vortex shedding behind a cylinder, demonstrates the excellent transient ability of the algorithm. All figures are from simulations where the type 1 velocity boundary condition has been used.

7.1
Lid-driven cavity

The (non-leaky) lid-driven cavity benchmark problem was solved with a 50×50 symmetrically graded mesh for both high and low Reynolds number flows in order to test the generality of the algorithm. In the lid-driven cavity problem, the boundary conditions are $u = 1$ on the top lid and $u = v = 0$ on all other sides and on the corners of the top lid. At the first node in from each corner of the top lid, $u = \Delta x_1/(\Delta x_1 + \Delta x_2)$, where Δx_1 is the length of the element on the top lid that is closest to the corner and Δx_2 is the length of the element on the top lid that is second closest to the corner. This is done in order to satisfy the "checkerboarding" constraint, see Sani et al. (1981). Type 1 boundary conditions eliminates the checkerboarding mode and works actually well even if u is set to $u = 1$ instead of $u = \Delta x_1/(\Delta x_1 + \Delta x_2)$. However, with the type 2 boundary condition, this leads to a non-convergent solution.

High Reynolds number. The Reynolds number $\mathrm{Re} = 7500$ was chosen for the high Re case, a case that has been well documented in Gresho et al. (1984), who used a 50×50 graded mesh and in Ghia et al. (1982), who used a 256×256 uniform mesh for a numerical simulation that is considered being a very accurate one.

The calculations were carried out such that the timestep was changed at each tenth timestep by an automatic timestepper that keep the Courant number constant at a user specified level and the calculations were carried on until nearly steady state. Figures 3 and 4 shows the traces

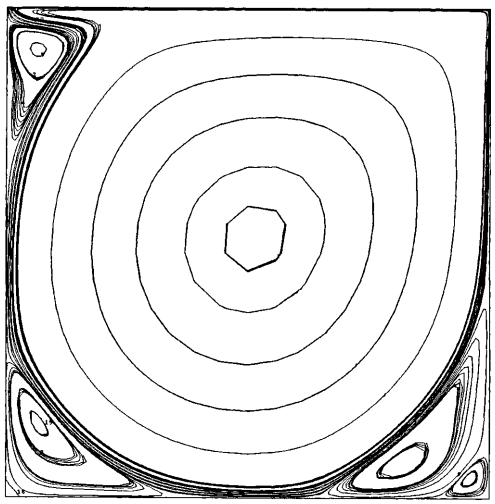

Fig. 3. Particle traces, $\mathrm{Re} = 7500$

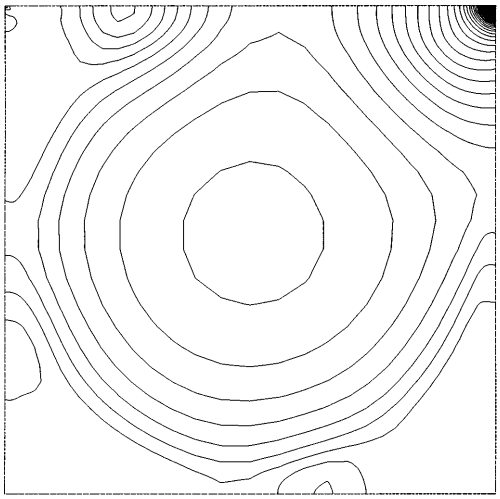

Fig. 4. Pressure contours, $\mathrm{Re} = 7500$

of massless particles inserted at a certain time and pressure contours at a Courant number of 0.8 using the semi-implicit consistent mass matrix three-step method with type 1 boundary conditions. Note that no wiggles at all are apparent in the velocity vector field, in spite of a rather coarse mesh for such a high Re-number.

Since the acceleration terms disappear at steady state, the lumped mass matrix yields more or less the same results as the consistent mass matrix. The results for the explicit lumped mass matrix and semi-implicit consistent mass matrix calculations in the form of minimum and maximum vertical velocity components $v_{\min}$ and $v_{\max}$ along the horizontal center line $(x, 0.5)$ and the minimum horizontal velocity component $v_{\min}$ along the vertical center line $(0.5, y)$ are shown in Table 1.

Low Reynolds number. The low Re-range ability of the code was tested by running the lid-driven cavity flow at Reynolds number $\mathrm{Re} = 1$. The results for $\mathrm{Re} = 1$ are presented in Fig. 5, Fig. 6 and Table 2. The calculations was carried out with the same mesh and boundary conditions as in the high Re-case. It should be noted that the boundary condition (4.4) is a high-Reynolds number condition and works well at high Re, but can not be expected to give accurate results in the very low Re-range. However, the main point with this test was to investigate how the semi-implicit method at low Reynolds number reacts to the highly simplified (and fast) linear equation solution process of (5.15), i.e. would it still be unconditionally stable? Several tests were done, and no diffusive limit on the timestep was found, not even in the case of pure diffusion. When comparing the results in Table 2, it should be noticed that the velocity distribution on the top lid used here are slightly different than in Hawken et al. (1990) or Sivaloganathan and Shaw (1988).

7.2
Vortex shedding behind a circular cylinder

Flow past a circular cylinder is a common test problem and a good way to evaluate the transient characteristics of an algorithm. It is a rather difficult test problem, since it requires an algorithm that is accurate in predicting the separation on the cylinder to ensure that the vortices are

Table 1. Values of velocity exterma, Re = 7500

	$u_{\min}(x=0.5,y)$	$v_{\min}(x,y=0.5)$	$v_{\max}(x,y=0.5)$
Explicit, lumped mass, $C_r = 0.8$, type 1 b.c.	−0.436	−0.561	0.423
Semi-implicit, cons. mass, $C_r = 0.8$, type 1 b.c.	−0.436	−0.563	0.423
Explicit, lumped mass, $C_r = 0.8$, type 2 b.c.	−0.437	−0.565	0.425
Semi-implicit, cons. mass, $C_r = 0.8$, type 2 b.c.	−0.439	−0.567	0.426
Ghia et al. [8]	−0.436	−0.552	0.440
Gresho et al. [12]	−0.430	−0.567	0.424

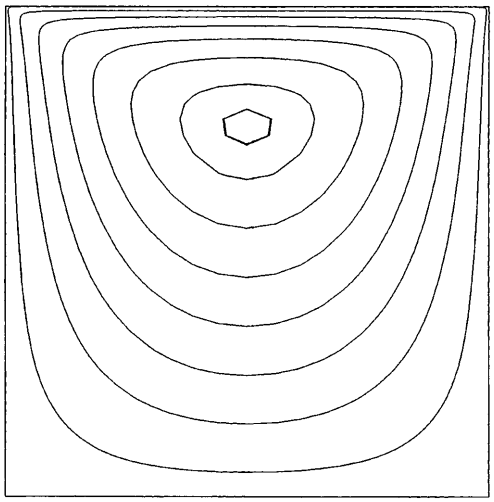

Fig. 5. Particle traces, Re = 1

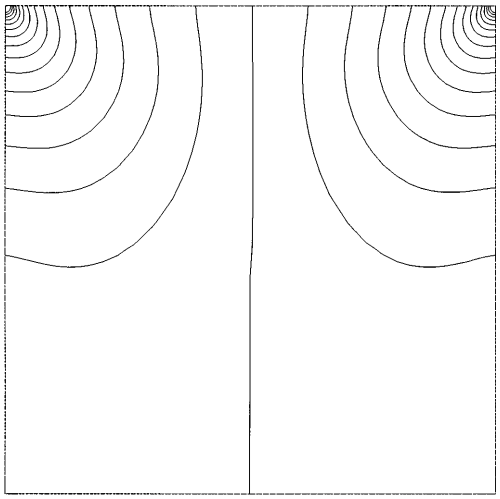

Fig. 6. Pressure contours, Re = 1

generated properly and which also has good phase accuracy in order to propagate the generated vortices through the domain without too much distortion. Thus, the excellent phase characteristics of the consistent mass matrix and the three-step method come to its full use.

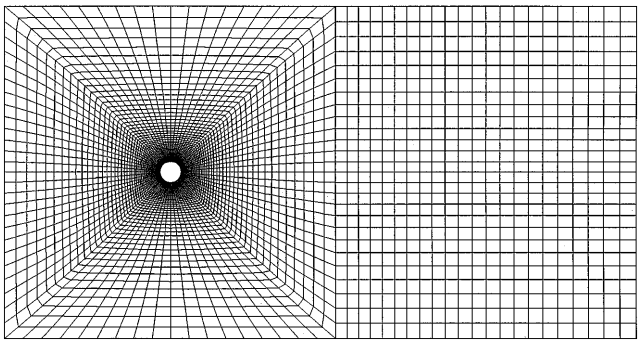

Fig. 7. Computational mesh for cylinder flow analysis (5346 velocity nodes)

The computations was carried out at a Reynolds number Re = 100 using the same size of the computational domain as in Tezduyar et al. (1992), i.e. the computational domain was 61 units long and 32 units high, where one unit is the radius r of the cylinder. Two different meshes were analyzed; one with 5236 elements and 5346 nodes and one with 2660 elements and 2742 nodes. The 5346 nodes mesh is shown in Fig. 7. The velocity boundary conditions are $u = 1$, $v = 0$ at the inlet and $u = v = 0$ on the cylinder. The traction free boundary condition was applied at the upper and lower wall, which have been found to give better results than the usual slip wall conditions, see Kato and Ikegawa (1991). The traction free boundary condition was also applied at the outlet. The initial velocity was set to $u = 1$, $v = 0$ in the whole domain and the Courant number was kept at 0.8 throughout the calculation by an automatic timestepper. Computations were done both for the type 1 and type 2 end-of-step velocity boundary conditions. Particle traces of massless particles inserted at the non-dimensional time $T = u_\infty t/r = 330$ are shown in Fig. 8. Typical instantaneous velocity vectors and pressure contours in the near wake region are shown in Figs. 9 and 10. Time histories of the drag coefficient $C_D = F_D/(1/2\rho u_{\text{in}}^2 S)$ and the lift coefficient $C_L = F_L/(1/2\rho u_{\text{in}}^2 S)$ are shown in Figs. 11–14. Here, S is the diameter of the cylinder and F_D, F_L are the forces per unit span in the drag and lift directions, which have been calculated from the distribution of pressure and shear stresses on the cylinder. The density ρ

Table 2. Values of velocity extrema, Re = 1

	$u_{\min}(x=0.5,y)$	$v_{\min}(x,y=0.5)$	$v_{\max}(x,y=0.5)$
Implicit cons. mass, $C_r = 0.1$, type 1 b.c.	−0.182	−0.185	0.185
Implicit cons. mass, $C_r = 0.1$, type 2 b.c.	−0.189	−0.184	0.184
Sivaloganathan and Shaw [25]	−0.20	−0.19	0.19
Hawken	−0.199	−0.177	0.177

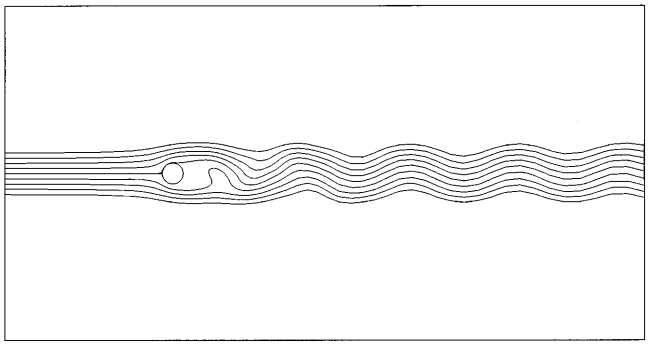

Fig. 8. Particle traces during vortex shedding at time $t = 330$ (5346 velocity nodes)

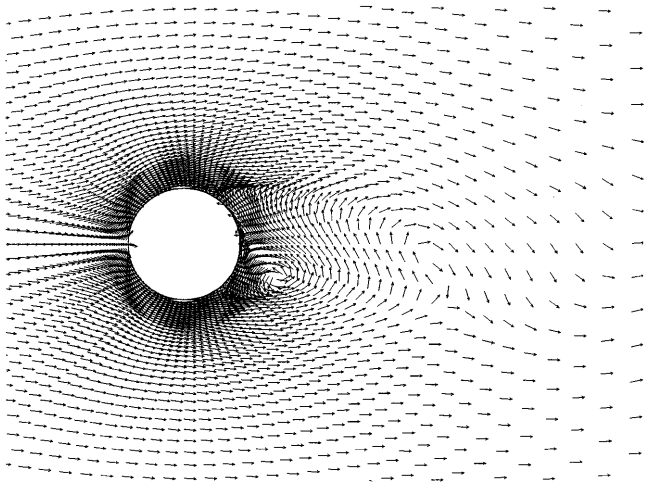

Fig. 9. Velocity vectors in the near wake region at time $t = 330$ (5346 velocity nodes)

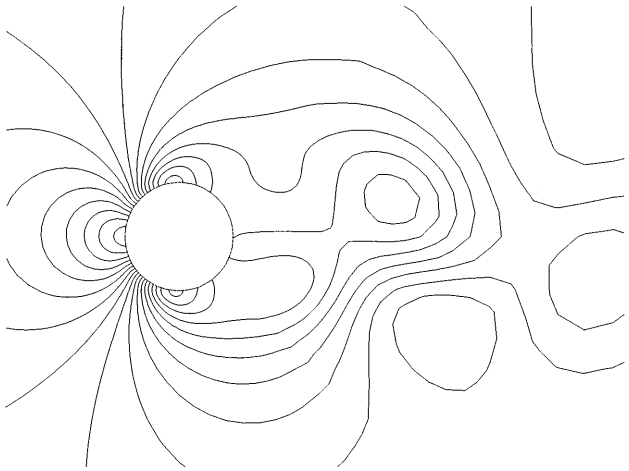

Fig. 10. Pressure contours in the near wake region at time $t = 330$ (5346 velocity nodes)

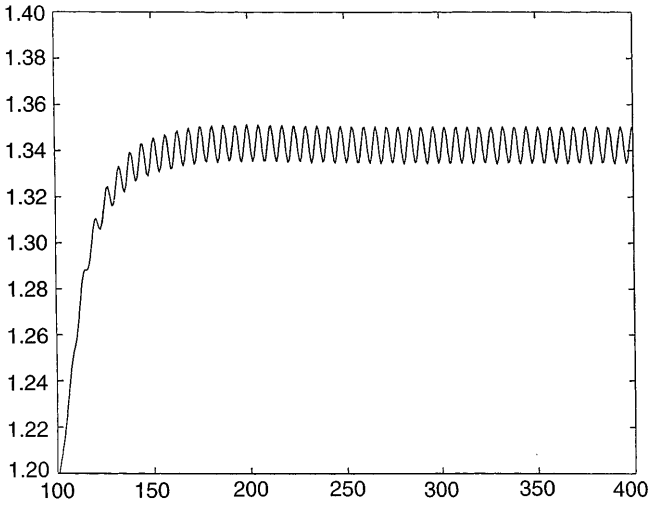

Fig. 11. Time history of the drag coefficient for the 2-dimensional cylinder flow with type 1 b.c. and 2742 nodes

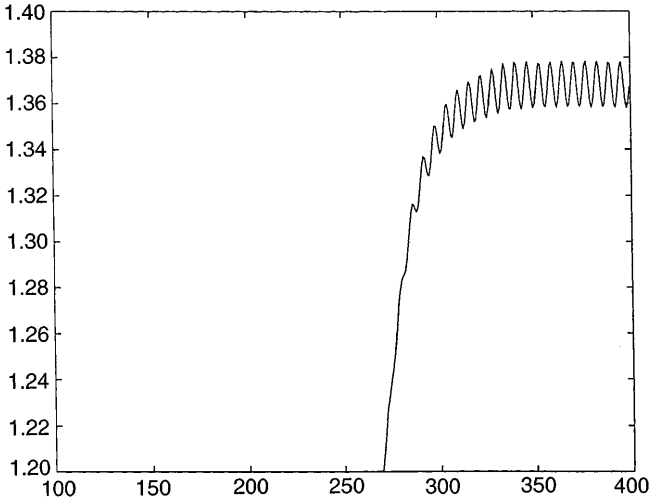

Fig. 12. Time history of the drag coefficient for the 2-dimensional cylinder flow with type 2 b.c. and 2742 nodes

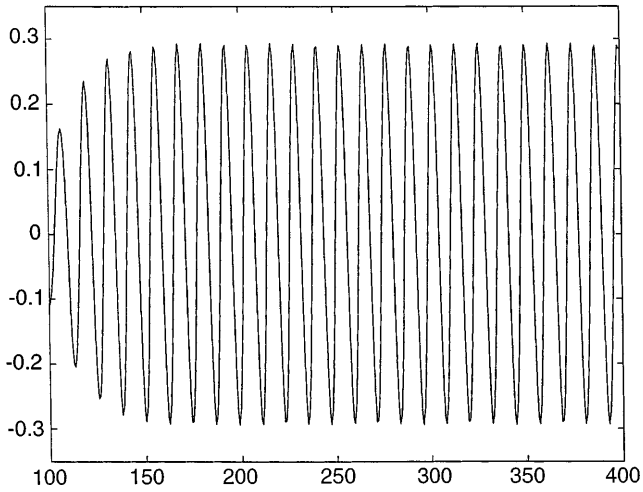

Fig. 13. Time history of the lift coefficient for the 2-dimensional cylinder flow with type 1 b.c. and 2742 nodes

Table 3. Summary of vortex shedding results, Re = 100

Scheme	St	C_{Dmean}	C_{Dpp}	C_{Lpp}
5346 nodes, type 1 b.c., $C_r = 0.8$	0.17	1.34	0.013	0.50
2742 nodes, type 1 b.c., $C_r = 0.8$	0.16	1.34	0.016	0.59
5346 nodes, type 2 b.c., $C_r = 0.8$	0.17	1.37	0.017	0.58
2742 nodes, type 2 b.c., $C_r = 0.8$	0.16	1.37	0.020	0.66
Gresho et al. [12], 1852 nodes	0.18	1.76	0.07	1.48
Tezduyar et al. [26], 5350 nodes	0.17	1.38–1.40	0.02	0.70–0.75
Jordan and Fromm (experiments) [18]	0.16–0.17	1.3		
Berger and Wille (experiments) [1]	0.16–0.17			

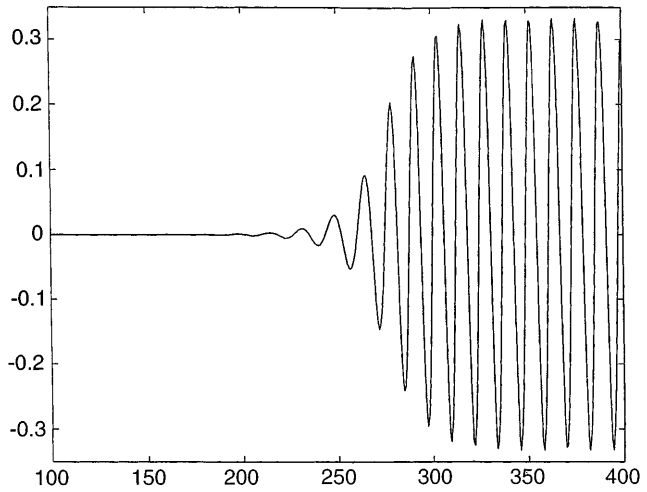

Fig. 14. Time history of the lift coefficient for the 2-dimensional cylinder flow with type 2 b.c. and 2742 nodes

was set to $\rho = 1$ and the velocity component in the x-direction at the inlet, u_{in}, was set to $u_{\text{in}} = 1$.

Values of time-averaged drag-and lift coefficients, $C_{D\text{mean}}$ and $C_{L\text{mean}}$ and Strouhal's number $St = fS/u_0$, where f is the vortex shedding frequency, are shown in Table 3. When comparing the results in Table 3, notice that Gresho and Sani (1984) have used a smaller computational domain and considerably fewer nodes in their analysis.

The type 1 end-of-velocity boundary condition gave slightly better results for the drag coefficient than the type2 boundary condition, while the resolution of the computational mesh had little influence. Furthermore, comparing Fig. 11 and Fig. 13 with Fig. 12 and Fig. 14 shows that the initiation of the vortex shedding started much earlier with the type 1 b.c.

The total computational time per node per sub-timestep for the semi-implicit algorithm with the consistent mass matrix was 0.0007 s on a 1 MFLOPS computer. This includes also the time spent in the automatic timestepper, calculation of lift and drag coefficients and time spent on writing out results and convergence checking parameters at regular intervals. This indicates that in the case of cylinder flow at Re = 100, the proposed method is not only accurate, but it is also rather efficient.

8 Conclusions

A finite element method for predicting time-dependent viscous incompressible flows over a wide range of inertial conditions has been presented. The method is mainly aimed at solving convection dominated flows and employs an explicit three-step algorithm for the convection terms, which gives not only high accuracy but also high efficiency since it allows large Courant numbers. To further improve accuracy for this kind of flows, the consistent mass matrix has also been included. Two variants of the method have been used; one fully explicit scheme with lumped mass matrix and one semi-implicit scheme with the consistent mass matrix in the momentum phase of the fractional step algorithm but with the lumped mass matrix in the pressure phase and in the pressure Poisson equation. The latter of these variants requires an extra system of linear equations to be solved at every time step. This was done in a simple and efficient way by using just a few Jacobi iterations and it was shown that this worked well even for very low Reynold's number flows. Two different kinds of velocity boundary conditions for the end-of-step velocity of the fractional step algorithm have been investigated, one which excludes checkerboarding (type 1 b.c.) and one simpler version which does not exclude the checkerboard mode (type 2 b.c.). The type 1 b.c. was found to be slightly more accurate and it was also found to initiate the vortex shedding behind the circular cylinder much earlier than the type 2 b.c.

One of the advantages with the algorithm described here is the simple algorithmic structure and that no extra terms or new higher-order derivatives are needed. In spite of the simplicity, the method is of a general nature and can easily handle complex geometries.

Numerical tests show good agreement with other numerical solutions and experimental data and suggest that the proposed method is competitive in terms of both accuracy and efficiency.

References

Berger E, Wille R (1972): Periodic flow phenomena. Ann. Rev. Fluid Mech. 4, 313

Brooks A, Hughes TJR (1982): Streamline upwind/Petrov-Galerkin formulation for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Comp. Meth. Appl. Mech. Eng. 32, 199–259

Chorin A (1968): Numerical solution of the Navier-Stokes equations. Math. Comput. 22, 745–762

Ding D, Townsend P, Webster M (1992): The iterative solution of Taylor-Galerkin augmented mass matrix equations. Int. J. Num. Meth. Eng. 35, 241–253

Donea J (1984): A Taylor-Galerkin method for convective transport problems. Int. J. Num. Meth. Eng. 20, 101–119
Donea J, Giuliani S, Laval H (1982): Finite element solution of the unsteady Navier-Stokes equations by a fractional step method. Comput. Meth. Appl. Mech. Eng. 30, 53–73
Donea J, Giuliani S, Morgan K, Quartapelle L (1981): The significance of chequerboarding in a Galerkin finite element solution of the Navier-Stokes equations. Int. J. Num. Meth. Eng. 17, 790–795
Ghia U, Ghia K, Shin C (1982): High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. J. Computat. Phys. 48, 387–411
Gresho P (1990): 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. Num. Meth. Fluids. 11, 587–620
Gresho P (1990): 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 2: implementation. Int. J. Num. Meth. Fluids. 11, 621–659
Gresho P, Chan S, Lee R, Upson C (1984) A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations. part 1: theory. Int. J. Num. Meth. Fluids. 4, 557–598
Gresho P, Chan S, Lee R, Upson C (1984): A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations. part 2: Applications. Int. J. Num. Meth. Fluids. 4, 619–640
Gresho P, Lee R, Sani R (1978): Advection-dominated flows, with emphasis on the consequences of mass lumping. Fin. Ele. Fluids 3, 335–350
Gresho PM, Sani RL (1987): On pressure boundary conditions for the incompressible Navier-Stokes equations. Fin. Ele. Fluids 7, 123–157
Hawken D, Tamaddon-Jahromi H, Townsend P, Webster M (1990): A Taylor-Galerkin-based algorithm for viscous incompressible flow. Int. J. Num. Meth. Fluids 10, 327–351
Hughes TJR, Brooks A (1979): A multi-dimensional upwind scheme with no crosswind diffusion, in Finite element methods for convection dominated flows. In: Hughes T (ed.), ASME 19–35, New York
Jiang O, Kawahara M (1993): A three-step finite element method for unsteady incompressible flows. Comput. Mech. 11, 355–370
Jordan S, Fromm J (1972): Oscillatory drag, lift, and torque on a circular cylinder in a uniform flow. Phys. Fluids 15, 371
Karniadakis GE, Israeli M, Orszag SA (1991): High-order splitting methods for the incompressible Navier-Stokes equations. J. Comput. Phys. 97, 414–443
Kato C, Ikegawa M (1991): Large eddy simulation of unsteady turbulent wake of a circular cylinder using the finite element method. ASME FED 117, 49–56
Kelly D, Nakazawa S, Zienkiewicz O, Heinrich J (1980): A note on anisotropic balancing dissipation in finite element method approximation to convection diffusion problems. Int. J. Num. Meth. Eng 15, 1705–1711
Kjellgren P, Yagawa G, Okuda H (1994): Accuracy and efficiency of finite element procedures for analysing turbulent flows using large eddy simulation. JSSC-94
Sani R, Gresho P, Lee R, Griffiths D (1981): The cause and cure? of the spurious pressures generated by certain FEM solutions of the incompressible Navier-Stokes equations. Int. J. Num. Meth. Fluids. 1, 17–43
Shen J (1993): A remark on the projection-3 method. Int. J. Num. Meth. Fluids. 16, 249–253
Sivaloganathan S, Shaw G (1988): A multigrid method for recirculating flows. Int. J. Num. Meth. Fluids. 8, 417–440
Tezduyar TE, Mittal S, Ray SE, Shih R (1992): Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements. Comp. Meth. Appl. Mech. Eng. 95, 221–242
Zienkiewicz O (1985): High-speed compressible flow and other advection-dominated problems of fluid dynamics. Fin. El. Fluids 6, John Wiley & Sons, Ltd., Chichester.