INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 26, 17–37 (1998)

PARALLEL FINITE ELEMENT COMPUTATION OF UNSTEADY INCOMPRESSIBLE FLOWS

TAMER WASFY,[1] ALAN C. WEST[2] AND VIJAY MODI[1]*

[1] *Department of Mechanical Engineering, Columbia University, New York, NY 10027, U.S.A.*
[2] *Department of Chemical Engineering, Columbia University, New York, NY 10027, U.S.A.*

SUMMARY

A parallel semi-explicit iterative finite element computational procedure for modelling unsteady incompressible fluid flows is presented. During the procedure, element flux vectors are calculated in parallel and then assembled into global flux vectors. Equilibrium iterations which introduce some 'local implicitness' are performed at each time step. The number of equilibrium iterations is governed by an implicitness parameter. The present technique retains the advantages of purely explicit schemes, namely (i) the parallel speed-up is equal to the number of parallel processors if the small communication overhead associated with purely explicit schemes is ignored and (ii) the computation time as well as the core memory required is linearly proportional to the number of elements. The incompressibility condition is imposed by using the artificial compressibility technique. A pressure-averaging technique which allows the use of equal-order interpolations for both velocity and pressure, this simplifying the formulation, is employed. Using a standard Galerkin approximation, three benchmark steady and unsteady problems are solved to demonstrate the accuracy of the procedure. In all calculations the Reynolds number is less than 500. At these Reynolds numbers it was found that the physical dissipation is sufficient to stabilize the convective term with no need for additional upwind-type dissipation. © 1998 John Wiley & Sons, Ltd.

Int. J. Numer. Meth. Fluids, **26**: 17–37 (1998)

KEY WORDS: incompressible Navier–Stokes; parallel finite element method; Galerkin approximation

1. INTRODUCTION

In this paper a semi-explicit finite element procedure for solving the time-dependent incompressible laminar Navier–Stokes equations is presented. As with typical explicit schemes, the present procedure can be easily implemented on a parallel computer architecture with a speed-up (over a single processor) that is nearly equal to the number of processors. This feature is of importance in fluid flow problems, where a large number of elements can make the computational time prohibitive for the majority of single processor serial computers.

There are a variety of finite element solution methods in use today for the time dependent incompressible Navier–Stokes equations. At two extremes among these are (i) direct methods to solve the fully implicit non-linear systems leading from the 'coupled' approach and (ii) explicit methods leading from the 'decoupled' approach. A discussion of these methods from the particular perspective of parallel computing and incompressible flows is provided in a recent review article by

Correspondence to: V. Modi, Department of Mechanical Engineering, Columbia University, 500 West 120th Street, Room 220, New York, NY 10027, U.S.A. Email: modi@columbia.edu

Contract grant sponsor: National Science Foundation; Contract grant number: CTS-93-15991

CCC 0271–2091/98/010017–21 $17.50

Received 10 July 1995
Revised 26 August 1996

Fischer and Patera.[1] Direct implicit methods involve solving the large number of finite element equations using some form of Gauss elimination, among which banded matrix LU methods are commonly used owing to their speed. The parallel efficiency of Gauss elimination algorithms is generally low, however, because most of the steps in the algorithm depend strongly on previous steps. Unlike in linear problems, the use of implicit methods for the non-linear Navier–Stokes equations also require equilibrium or 'inner' iterations. An advantage of these methods is the lack of a time step restriction, even though in practice, to ensure time accuracy of the solution, the time step cannot be arbitrarily large.

A purely explicit method, on the other hand, is limited by a severe time step restriction arising out of stability considerations. The performance of explicit methods, however, can be improved if some type of 'local implicitness' is introduced with equilibrium iterations. Iterative semi-explicit methods[2] are well suited for dynamic non-linear problems because they are local (in the sense that within a small time step the effect of a nodal flux decreases as the distance from the node increases) and the critical time step is somewhat larger than for purely explicit methods. In addition, an explicit or an iterative time-explicit method has an ideal computational cost per time step which is linearly proportional to the number of finite elements, E. This is an important advantage over implicit methods that need to perform Gauss elimination and hence have a computational cost proportional to $n \times b \times b$, where n is the total number of degrees of freedom and b is the bandwidth. Here n is equal to the total number of nodes (which is roughly equal to the number of elements, E, for a rectangular four-node element) multiplied by D, the number of degrees of freedom per node (for two-dimensional problems, $D=3$, since there are two velocity degrees of freedom and one pressure degree of freedom), and b is at least proportional to $\sqrt{n}$.

Another significant advantage of explicit methods is that the core memory storage requirement is significantly less than for implicit methods. In the latter methods the core memory required is proportional to $n \times b$, while in explicit methods the core memory required is roughly of the order of $E \times \max(n^e)^2 \times D^2$, where $\max(n^e)$ is the maximum number of nodes of an element. For large problems this can translate into huge core memory savings.

The new generation of MIMD (multiple-instruction–multiple-data)/GSM (global shared memory) parallel machines are ideal for explicit methods, since the theoretical 'parallel speed-up' is equal to the number of parallel processors. In practice, however, the parallel speed-up degrades with the number of processors owing to intensive memory access by the processors, which leads to congestion on the memory bus and memory access bottlenecks. The semi-explicit iterative finite element solution procedure presented in this paper achieved a speed-up of 14 on a 16-processor MIMD/GSM SGI Power Challenge. In the procedure, at each iteration within a time step, the calculation of the element flux vectors, which is the most CPU-intensive step, is performed in parallel with a computational time proportional to E/P, where P is the total number of processors. For an iteration the element flux vectors are mutually independent (non-commutative) and thus can be distributed on a parallel computer system. The term mutual independence means that the output of one vector is not used to evaluate another vector and that two processors do not perform a memory write operation to the same location at the same time. If this occurs, then the result of that operation is unpredictable and in general wrong. Fortunately, simple algorithmic 'tricks' can prevent this form from occurring (see e.g. References 3 and 4). A read operation to the same memory location at the same time is permissible. The element fluxes are then assembled (summed) in parallel and used in the global finite element equations to solve for the nodal values also in parallel.

The present procedure can also be used on an MIMD parallel computer cluster with distributed memory.[5] In this case, at the end of each iteration, a processor transfers the assembled flux vectors from its own memory to the memory of another processor as well as receiving the updated vector of degrees of freedom needed for the next iteration. This obviously adds some communication overhead.

The advantage of distributed memory MIMD clusters is that it is easier to add more processors and hence increase the computational capacity.

After calculating and assembling the global flux vectors, the global acceleration vector is evaluated in parallel using a lumped mass matrix approach. Then, using the Newmark[6] numerical integration technique, the global acceleration vector is used to find the nodal velocities, which in turn are used to calculate better estimates of the element flux vectors. This iterative procedure is repeated a prescribed number of times. These iterations introduce some local implicitness which permits a moderate increase in the effective critical time step size. In addition, an implicitness parameter is introduced to reduce the change between iterations. The number of equilibrium iterations increases as the implicitness parameter increases. However, as the number of equilibrium iterations increases, the critical time step size also increases. Depending on the nature of the problem, an optimum implicitness parameter value, time step size and number of iterations exist which minimize the computing time while maintaining a high degree of time accuracy.

The performance of an algorithm is also tied to the choice of the governing equations and their discretization. There are two particular difficulties associated with the solution of the incompressible Navier–Stokes equations. The first difficulty is that the Navier–Stokes equations are actually a set of two partial differential equations, namely the momentum conservation equation and the mass conservation equation, which must be satisfied simultaneously. A related issue is that in the incompressible limit there is no explicit equation for pressure. The second difficulty arises from the presence of the convective terms which is non-linear and non-symmetric.

The coupling between the momentum and continuity equations imposes the so-called Babuska–Brezzi condition,[7,8] namely that the degree of the polynomial interpolation function for pressure must be at least one order less than that for velocity. For elements that satisfy this condition, the velocity and pressure interpolation functions are either quadratic and linear respectively or linear and constant respectively. Unfortunately, the use of lower-order pressure interpolation frequently introduces an indesirable pressure-checkerboarding effect,[9] even though the velocity field remains acceptable at least for steady flows. In unsteady flows, however, which are sensitive to pressure changes, the pressure-checkerboarding effect might lead to inaccurate time evolution of the solution. To alleviate this problem, pressure-smoothing techniques have been used, of which a least-squares-type smoothing is the most common.[9,10]

It would be more advantageous to use equal-order interpolations for velocities and pressure, since only one set of interpolation functions needs to be determined and the algorithm used to calculate the finite element fluxes is simpler. The techniques used to allow equal-order interpolations for velocities and pressure involve the use of additional stabilizing diffusive terms in both the momentum and continuity equations. Among these techniques are the Galerkin least squares (GLS) finite element approximation[11–16] and the Taylor–Galerkin formulation.[17–19] In the GLS formulation the stabilizing terms are obtained by minimizing the sum of the squared residuals of the momentum and continuity equations integrated over a finite element. In the Taylor–Galerkin formulation, alternative time-stepping methods based on a forward time Taylor series expansion are used, thus introducing stabilizing diffusive terms, similar to those introduced in GLS, in the governing equations. The complexity of these stabilizing terms, however, offsets the advantages of the simplicity introduced by equal-order interpolation. In the present study the Babuska–Brezzi conditions is circumvented by a pressure-'averaging' procedure at each equilibrium iteration. This permits the use of equal-order interpolations for velocity and pressure with the standard Galerkin formulation.

A related complexity in dealing with the coupling between the momentum and continuity equations is that in the incompressible limit there is no explicit equation for the pressure. To allow the use of an explicit solution procedure that lends itself to parallelization, the artificial compressibility technique[20] is employed herein. In this technique a time derivative of pressure is introduced into the

continuity equation. The coefficient of this term is chosen to reduce the stiffness of the new system of equations while ensuring that the flow is nearly incompressible.

The second difficulty in dealing with the Navier–Stokes equations is the presence of the convective term. The standard Galerkin approximation in which the test functions are chosen to be the same as the trial functions reduces to a central-difference-type approximation. This differencing, owing to the non-symmetric nature of the convective operator, leads to spurious oscillations in a convection-dominated problem. Upwind FEM techniques solve this problem by using a biased or upwind difference approximation for the convective term.[9,11,12,21–25] In the range of Reynolds numbers of interest (limited to an upper value of 500) in this study, use of the standard Galerkin formulation was found to be acceptable. Apparently the physical diffusion present in the problem is sufficient to damp the spurious oscillations.

This paper is divided into eight sections. The integral conservative form of the governing equations along with our artificial compressibility hypothesis is presented in Section 2. Notation and preliminary steps are introduced in Section 3. In Section 4 the finite element discretization of the momentum equation is presented. In Section 5 the finite element discretization of the continuity equation along with the pressure-averaging procedure is presented. The solution procedure is detailed in Section 6. Examples which show the features of the present method are given in Section 7. Finally, in Section 8, conclusions are offered.

2. GOVERNING EQUATIONS

The dynamic response of incompressible fluids is described by the equations of conservation of momentum and mass. The former can be written in integral form as

$$\int_V \rho \frac{\partial u_i}{\partial t}\, \mathrm{d}V = \int_V \frac{\partial(-\rho u_i u_j + \sigma_{ij})}{\partial x_j}\, \mathrm{d}V + \int_V \rho F_i\, \mathrm{d}V, \tag{1}$$

where V is a fixed control volume, t is the running time, ρ is the density of the fluid, $\vec{u}$ is the fluid velocity vector, $\vec{x}$ is the position vector, $\tilde{\sigma}$ is the stress tensor and $\vec{F}$ is the vector of body forces. In order to introduce the artificial compressibility approach, it is more natural to start with the compressible form of the continuity equation:

$$\int_V \frac{\partial \rho}{\partial t}\, \mathrm{d}V + \int_V \frac{\partial(\rho u_i)}{\partial x_i}\, \mathrm{d}V = 0. \tag{2}$$

Notes

1. A conservative form of the governing equations is used.
2. The summation convention over repeated indices is used throughout the paper unless otherwise specified.
3. In all subsequent deviations all motions are assumed to be spatial. Therefore the range of the indices i and j in (1) and (2) is from one to three.

Artificial Compressibility

In general for compressible fluids the absolute pressure P^* is related to the density using an equation of state of the form

$$\rho = f(P^*, T), \tag{3}$$

where T is the temperature. For a slightly compressible fluid the variations in P^* are small even though its absolute value is large, so it is more convenient to split P^* as

$$P^* = P + P_\infty, \tag{4}$$

where P_∞ is the ambient reference pressure and P is a small pressure change. In the absence of temperature variations, if we assume the density to be linearly related to the pressure, we may write:

$$\rho = cP^* = c(P + P_\infty), \tag{5}$$

where the coefficient c is a property of the fluid. In the artificial compressibility approach[20] the pressure change P in (5) is replaced by P/a for the evaluation of the time derivative of density in the continuity equation (2), hence

$$\rho = c\left(\frac{P}{a} + P_\infty\right), \tag{6}$$

where a is the artificial compressibility parameter. In all other terms the density can be assumed to be constant, i.e. $\rho = cP_\infty$, since pressure changes P are much smaller than P_∞.

In (1) the stress tensor can be written as

$$\sigma_{ij} = -P^*\delta_{ij} + \tau_{ij}, \tag{7}$$

where τ_{ij} is the deviatoric stress evaluated using an appropriate strain rate measure and constitutive material model. For Newtonian fluids,

$$\tau_{ij} = \lambda D_{kk}\delta_{ij} + 2\mu D_{ij}, \tag{8}$$

where λ and μ are viscosity coefficients and $\tilde{D}$ is the rate-of-deformation tensor given by

$$D_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right). \tag{9}$$

For incompressible fluids, $\partial u_i/\partial x_i = 0$, so we can write (9) as

$$\tau_{ij} = 2\mu D_{ij}. \tag{10}$$

Substituting (4), (7) and (10) in (1) gives the momentum equation for incompressible fluids:

$$\rho\frac{\partial u_i}{\partial t} = -\rho\frac{\partial(u_i u_j)}{\partial x_j} - \frac{\partial P}{\partial x_i} + \mu\frac{\partial^2 u_i}{\partial x_j \partial x_j} + \rho F_i. \tag{11}$$

Substituting (6) in (2) gives the artificial compressibility continuity equation:

$$\frac{\partial P}{\partial t} = -aP_\infty\frac{\partial u_i}{\partial x_i}. \tag{12}$$

Theoretically for incompressible flows the speed of propagation of a disturbance, s (speed of sound), is infinite. However, using our artificial compressibility momentum and continuity equations (11) and (12), it can be easily shown that s is finite and given by

$$s = \sqrt{(a/c)}. \tag{13}$$

In our semi-explicit iterative solution procedure the time step is inversely proportional to s. The key advantage of the artificial compressibility approach is that s can be 'artificially' reduced using the artificial compressibility parameter a, thus allowing larger time steps. To ensure that the flow solution does not deviate significantly from the incompressible flow we are attempting to model, the reference

Mach number Ma is not allowed to exceed 0·1, where $Ma = U/s$ and U is a characteristic flow speed.

3. NOTATION AND PRELIMINARIES

We define a global inertial Cartesian reference frame. Let x_{Ij} and u_{Ij} by the position co-ordinate and fluid velocity respectively in direction j of that frame at global node I. Uppercase indices will be used to denote node numbers. The nodal pressures can be written as P_I, where I is the global node number. Next we define a local-node-numbering scheme for each element. Let C_I^e be the global node number of local node I of element e and n^e be the total number of nodes of element e. Also let x_{Ij}^e and u_{Ij}^e be the position co-ordinates and fluid velocity respectively in direction j of the inertial reference frame at local node I of element e. Similarly, we define the vectors of elemental nodal pressures P_I^e. Accordingly, the relation between the local element inertial co-ordinates and the global inertial co-ordinates can be written as $x_{Ij}^e = x_{C_I^e j}$.

For each finite element of volume V^e the unknown continuous variables of (1) and (2) are expressed in terms of their nodal values using the matrix of interpolation functions:

$$u_i = N_j^e u_{Ji}^e, \quad \text{no summation over } e \text{ and summation over } j \ (j = 1 \to n^e), \tag{14}$$

$$P = N_j^e P_j^e, \tag{15}$$

where N_j^e is the interpolation function associated with local node number J of element e.

4. FINITE ELEMENT DISCRETIZATION FOR MOMENTUM EQUATION

Using the Galerkin finite element approximation, the infinitesimal forces in (1) are distributed to the finite element nodes using the interpolation functions N_K^e:

$$\int_{V^e} N_K^e \rho \frac{\partial u_i}{\partial t} \, \mathrm{d}V = \int_{V^e} N_K^e \frac{\partial(-\rho u_i u_j + \sigma_{ij})}{\partial x_j} \, \mathrm{d}V + \int_{V^e} N_k^e \rho F_i \, \mathrm{d}V. \tag{16}$$

Note that since the fluid is incompressible, the density is constant and uniform over the problem domain. Integrating by parts and introducing S^e, the surface of the finite element, and n_i, the normal to the surface, (16) can be written as

$$[M^e]\{\dot{u}^e\} = \{F_{\mathrm{c}}^e\} + \{F_{\mathrm{m}}^e\} + \{F_{\mathrm{f}}^e\} + \{F_{\mathrm{r}}^e\} + \{F_{\mathrm{a}}^e\}, \tag{17}$$

where

$$M_{KJ}^e = \int_{V^e} \rho N_K^e N_J^e \, \mathrm{d}V, \tag{18}$$

$$F_{c_{Ki}}^e = \int_{V^e} \frac{\partial N_K^e}{\partial x_j} \rho u_i u_j \, \mathrm{d}V, \tag{19}$$

$$F_{\mathrm{m}_{Ki}}^e = \int_{V^e} \frac{\partial N_K^e}{\partial x_j} \sigma_{ij} \, \mathrm{d}V, \tag{20}$$

$$F_{\mathrm{f}_{Ki}}^e = -\int_{S_o^e} (N_K^e \rho u_i u_j + N_K^e \sigma_{ij}) n_j \, \mathrm{d}S, \tag{21}$$

$$F_{\mathrm{r}_{Ki}}^e = \int_{S_i^e} (-N_K^e \rho u_i u_j + N_K^e \sigma_{ij}) n_j \, \mathrm{d}S, \tag{22}$$

$$F_{\mathrm{a}_{Ki}}^e = \int_{V^e} N_K^e \rho F_i \, \mathrm{d}V. \tag{23}$$

Here $[M^e]$ is the element mass matrix, $\{\dot{u}^e\}$ is the vector of nodal accelerations, $\{F^e_c\}$ is the vector of convective forces, $\{F^e_m\}$ is the vector of internal material forces, $\{F^e_f\}$ is the vector of free surface forces, $\{F^e_r\}$ is the vector of inter-element reaction forces and $\{F^e_a\}$ is the vector of externally applied forces. In (21), S_o is the outer surface of the element, and in (22), S_i is the inner surface of the element. The outer surface is the free surface of the domain and the inner surface is the surface with which the element is interfaced with other surrounding elements. The inter-element reaction term $\{F^e_r\}$ which is applied over S_i cancels out between elements after assembly of the element equations. Substituting (14) in (19), we can write

$$F^e_{c_{Ki}} = C^e_{KLMj} u^e_{Li} u^e_{Mj}, \tag{24}$$

where

$$C^e_{KLMj} = \int_{V^e} \frac{\partial N^e_K}{\partial x_j} \rho N^e_L N^e_M \, \mathrm{d}V. \tag{25}$$

Also, substituting (14), (15), (7) and (10) in (20) gives

$$F^e_{m_{Ki}} = Q_{KLi} P^e_L + K^e_{KL} u^e_{Li}, \tag{26}$$

where

$$Q^e_{KLi} = \int_{V^e} \frac{\partial N^e_K}{\partial x_i} N^e_L \mathrm{d}V, \tag{27}$$

$$K^e_{KL} = \int_{V^e} \mu \frac{\partial N^e_K}{\partial x_j} \frac{\partial N^e_L}{\partial x_j} \, \mathrm{d}V. \tag{28}$$

Thus for each element e we calculate a four-dimensional array C^e_{KLMj}, a three-dimensional array Q^e_{KLi} and a two-dimensional array K^e_{KL}. All these arrays are constant with time. The calculation of these arrays takes the most computer time; however, since they are constant, they can be calculated only once for each element and then during the simulation we multiply these arrays by the element nodal values (see (24) and (26)) to find the fluid forces. Similarly, substituting (14), (15), (7) and (10) in (21) gives

$$F^e_{f_{Ki}} = -\bar{C}^e_{KLMj} u^e_{Li} u^e_{Mj} + \bar{Q}^e_{KLi} P^e_L - \bar{K}^e_{KL} u^e_{Li}, \tag{29}$$

where

$$\bar{C}^e_{KLMj} = -\int_{S^e_o} \rho N^e_K N^e_L N^e_M n_j \, \mathrm{d}S, \tag{30}$$

$$\bar{Q}^e_{KLj} = \int_{S^e_o} N^e_K N^e_L n_i \, \mathrm{d}S, \tag{31}$$

$$\bar{K}^e_{KL} = -\int_{S^e_o} \mu N^e_K \frac{\partial N^e_L}{\partial x_j} n_j \, \mathrm{d}S. \tag{32}$$

Similarly, $\bar{C}^e_{KLMj}$, $\bar{Q}^e_{KLi}$ and $\bar{K}^e_{KL}$ are all constant arrays which can be calculated once for each element.

5. FINITE ELEMENT DISCRETIZATION FOR CONTINUITY EQUATION

Using the Galerkin finite element method, the continuity equation (12) can be written as

$$\int_{V^e} N^e_K \dot{P} \mathrm{d}V = -aP_\infty \int_{V^e} N^e_K \frac{\partial u_i}{\partial x_i} \, \mathrm{d}V. \tag{33}$$

Integrating the RHS of (33) by parts, we can rewrite it in the form

$$[C^e]\{\dot{P}^e\} = \{L^e_{\mathrm{b}}\} + \{L^e_{\mathrm{f}}\} + \{L^e_{\mathrm{r}}\}, \tag{34}$$

where

$$[C^e] = \int_{V^e} N^e_K N^e_J \, \mathrm{d}V, \tag{35}$$

$$L^e_{\mathrm{b}_K} = aP_\infty \int_{V^e} \frac{\partial N^e_K}{\partial x_i} u_i \, \mathrm{d}V, \tag{36}$$

$$L^e_{\mathrm{f}_K} = -aP_\infty \int_{S^e_{\mathrm{o}}} N^e_K u_i n_i \, \mathrm{d}S, \tag{37}$$

$$L^e_{\mathrm{r}_K} = -aP_\infty \int_{S^e_i} N^e_K u_i n_i \, \mathrm{d}S. \tag{38}$$

Here $[C^e]$ is the compressibility matrix, $\{L^e_{\mathrm{b}}\}$ is the vector of mass flux, $\{L^e_{\mathrm{f}}\}$ is the vector of free surface mass flux and $\{L^e_{\mathrm{r}}\}$ is the inter-element mass flux vector which cancels out between elements after assembly of the element equations. Substituting (14) in (36) gives

$$L^e_{\mathrm{b}_K} = R^e_{KLi} u^e_{Li}, \tag{39}$$

where

$$R^e_{KLi} = aP_\infty \int_{V^e} \frac{\partial N^e_K}{\partial x_i} N^e_L \, \mathrm{d}V. \tag{40}$$

Similarly, substituting (14) in (37) gives

$$L^e_{\mathrm{f}_K} = \bar{R}^e_{KLi} u^e_{Li}, \tag{41}$$

where

$$\bar{R}^e_{KLi} = -aP_\infty \int_{S^e_{\mathrm{o}}} N^e_K N^e_L n_i \mathrm{d}S. \tag{42}$$

R^e_{KLi} and $\bar{R}^e_{KLi}$ are three-dimensional constant arrays which are evaluated once at the beginning of the solution procedure.

Pressure-Averaging Technique

Substituting (8) in (7), we see that the pressure is added to the rate-of-deformation tensor $\tilde{D}$. If equal-order interpolations are used for velocity and pressure, then the degree of the polynomial of $\tilde{D}$ is lower than that for the pressure. This leads to an unstable solution. This difficulty is circumvented here by the use of a pressure-'averaging' procedure that effectively removes the instability associated with equal-order interpolations. In addition, pressure averaging propagates pressure information from the exit, where the pressure boundary condition is set, to the interior of the domain. The pressure-averaging technique consists of performing the following operation for all the nodes at each iteration:

$$P_I = w_0 P_1 + \sum_{K=1}^{nn(I)} w_K P_{n(I,K)}, \tag{43}$$

$$\sum_{K=0}^{nn(I)} w_K = 1, \tag{44}$$

where P_I is the pressure at node I, $nn(I)$ is a one-dimensional array containing the total number of nodes immediately adjacent to node I, $n(I, K)$ is a two-dimensional array containing the Kth node number of the node adjacent to node I and w_K is the weight assigned to the Kth node. Note that $n(I, 0) = I$. In addition, the same averaging procedure is performed on the assembled mass flux vector $\{L_b\}$:

$$L_{b_I} = w_0 L_{b_I} + \sum_{K=1}^{nn(I)} w_K L_{b_{n(I,K)}}. \tag{45}$$

For a two-dimensional mesh of rectangular four-node Lagrangian isoparametric elements the pressure-averaging routine can be performed using the following two steps. The first step is to find the average pressure for each element (P^e):

$$P^e = \frac{1}{n^e} \sum_{I=1}^{n^e} P_{C_i^e}. \tag{46}$$

The second step is to find the pressure at a node using

$$P_I = (1 - \beta) P_I + \beta \frac{1}{e^I} \sum_{k=1}^{e^I} P^{B_k^I}, \tag{47}$$

where e^I is the number of elements which have node I as a common node, B_k^I is the global element number of element number k which is connected to global node I and β is a pressure-averaging parameter. In (47), $\beta = 0$ corresponds to zero pressure averaging and $\beta = 1$ to full pressure averaging. The two steps in (46) and (47) are performed on the mass flux vector. The above procedure is applied to boundary nodes as well, but now the elements e^I which have node I as a common node are fewer.

For a typical interior node, say node number I in Figure 1, the weights corresponding to full pressure averaging were found to give the most stable and accurate solution while allowing the largest time step:

$$w_0 = 0{\cdot}25, \qquad w_1 = w_2 = w_3 = w_4 = 0{\cdot}125, \qquad w_5 = w_6 = w_7 = w_8 = 0{\cdot}0625.$$

Note that the weights satisfy (44). Also, note that pressure averaging as small as

$$w_0 = 0{\cdot}9625, \qquad w_1 = w_2 = w_3 = w_4 = 0{\cdot}00625, \qquad w_5 = w_6 = w_7 = w_8 = 0{\cdot}003125$$

is enough to stabilize the solution, but the restriction on the time step size in that case is more severe.

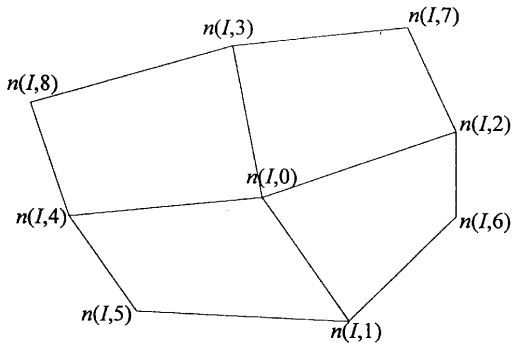

Figure 1. Two-dimensional mesh composed of rectangular four-node Lagrangian elements

6. PARALLEL NUMERICAL SOLUTION PROCEDURE

Assembly of Global Flux Vectors

Equations (16) and (33) are the conservation finite element equations for an element written for the element nodes. In general a node is in common between many elements. The discretized governing equations of the entire fluid domain consist of the conservation equations of all the nodes. These equations are assembled from the element equations by adding the flux contributions at a node from all the surrounding elements. In (17) and (34) the nodal flux vectors are $\{F_c^e\}$, $\{F_m^e\}$, $\{F_f^e\}$, $\{L_b^e\}$ and$\{L_f^e\}$. These vectors are evaluated using (24), (26), (29), (39) and (41) respectively. At an iteration within a time step these vectors are mutually independent and thus can be readily distributed on a parallel computer. The assembled discretized finite element equations can be written as

$$\{\dot{u}\} = [M]^{-1}(\{F_c\} + \{F_m\} + \{F_f\} + \{F_a\}), \tag{48}$$

$$\{\dot{P}\} = [C]^{-1}(\{L_b\} + \{L_f\}). \tag{49}$$

The global mass matrix $[M]$ and compressibility matrix $[C]$ are assembled from the elemental mass and compressibility matrices respectively. Since $[M]$ and $[C]$ do not change with time, they can be evaluated and inverted once at the start of the solution procedure. In order to reduce the computational burden as well as to preserve the full concurrency of the procedure, these matrices will be lumped such that only the diagonal terms have non-zero values. In this case the inversion of these matrices becomes trivial.

Newmark Time Integration

The simultaneous solution of (48) and (49) along with constraint equations gives the time history of the velocity and pressure fields. The constraint equations are generally algebraic equations which constrain the nodal values of some of the nodes. Equations (48) and (49) are first-order ordinary differential equations in time. We will use a semi-explicit iterative solution procedure based on Newmark numerical integration.[26] Using this procedure, (48) and (49) are first rewritten in the form

$$\{\dot{u}\}_t^i = [M]^{-1}(\{F_c\}_t^{i-1} + \{F_m\}_t^{i-1} + \{F_f\}_i^{i-1} + \{F_a\}_t^{i-1}), \tag{50}$$

$$\{\dot{P}\}_t^i = [C]^{-1}(\{L_b\}_t^{i-1} + \{L_f\}_t^{i-1}), \tag{51}$$

where subscript t indicates the time and superscript i indicates the iteration number. These equations are integrated numerically in time using the Newmark method:

$$\{u\}_t^i = [\{u\}_{t-\Delta t} + (1-\alpha)\Delta t\{\dot{u}\}_{t-\Delta t} + \alpha\Delta t\{\dot{u}\}_t^i](1-p) + \{u\}_i^{i-1}p, \tag{52}$$

$$\{P\}_t^i = [\{P\}_{t-\Delta t} + (1-\alpha)\Delta t\{\dot{P}\}_{t-\Delta t} + \alpha\Delta t\{\dot{P}\}_t^i](1-p) + \{P\}_t^{i-1}p, \tag{53}$$

where Δt is an appropriately chosen time step, α is the parameter of the Newmark integration and p is an implicitness parameter. For an unconditionally stable numerical solution and no numerical damping we choose $\alpha = 0{\cdot}5$, which corresponds to the classic Crank–Nicolson time stepping. Note that the initial conditions at the previous time step, namely $\{u\}_{t-\Delta t}$, $\{\dot{u}\}_{t-\Delta t}$, $\{P\}_{t-\Delta t}$ and $\{\dot{P}\}_{t-\Delta t}$, are known. Also, in (52) and (53) we will use the values at the previous time step as the initial guess in our iterative procedure. Thus

$$\{u\}_t^0 = \{u\}_{t-\Delta t}, \qquad \{\dot{u}\}_t^0 = \{\dot{u}\}_{t-\Delta t}, \qquad \{P\}_t^0 = \{P\}_{t-\Delta t}, \qquad \{\dot{P}\}_t^0 = \{\dot{P}\}_{t-\Delta t}. \tag{54}$$

The implicitness parameter p introduced in (52) and (53) has a value between zero and one, with the purpose of reducing the changes between iterations so as to increase the size of the critical time step.

The trade-off is that more iterations will be required per time step. Let R be the minimum number of iterations required per time step. As we increase the value of p, the number of iterations, R, increases and the maximum allowable time step (critical time step) also increases. As a guideline to choosing R and p, we suggest the relationship

$$p^R < 0{\cdot}017. \tag{55}$$

To use the above relation, one first selects a value for p and then finds the value of R that satisfies the relation. For example, for $p = 0{\cdot}6$ the value of R is $R = 8$. The meaning of (55) is very easily demonstrated if we expand equation (52) or (53) at the Rth iteration:

$$\begin{aligned}\{u\}_t^R = {} & [\{u\}_{t-\Delta t} + (1-\alpha)\Delta t\{\dot{u}\}_{t-\Delta t} + \alpha\Delta t\{\dot{u}\}_t^R](1-p) \\ & + [\{u\}_{t-\Delta t} + (1-\alpha)\Delta t\{\dot{u}\}_{t-\Delta t} + \alpha\Delta t\{\dot{u}\}_t^{R-1}](1-p)p \\ & + \cdots \\ & + [\{u\}_{t-\Delta t} + (1-\alpha)\Delta t\{\dot{u}\}_{t-\Delta t} + \alpha\Delta t\{\dot{u}\}_t^{1}](1-p)p^{R-1} \\ & + \{u\}_t^0 p^R. \end{aligned} \tag{56}$$

Therefore (55) guarantees that at the end of R iterations the weight of our initial guess (given in (54)) will be 1·7%.

Solution Algorithm

An outline of the semi-explicit iterative numerical solution procedure is given subsequently. The following two steps are carried out once at the beginning of the solution procedure.

(a) $[M]$ and $[C]$ are evaluated and inverted. We have used lumped matrices $[M]$ and $[C]$; thus these matrices are diagonal. In that case the core memory storage required is very small compared with the consistent matrices. Also, the inversion process is trivial.
(b) The arrays C^e_{klmj}, Q^e_{kli}, K^e_{kl}, $\bar{C}^e_{klmj}$, $\bar{Q}^e_{kli}$, $\bar{K}^e_{kl}$, R^e_{kli} and $\bar{R}^e_{kli}$ are evaluated.

Then at each iteration the following steps are performed.

1. Element flux vectors $\{F^e_{\mathrm{c}}\}$, $\{F^e_{\mathrm{m}}\}$, $\{F^e_{\mathrm{f}}\}$, $\{L^e_{\mathrm{b}}\}$ and $\{L^e_{\mathrm{f}}\}$ are evaluated in parallel on an element-by-element basis using (24), (26), (29), (39) and (41) respectively (see Note 1 below).
2. Global flux vectors are assembled in parallel on an element-by-element basis from the element vectors.
3. Equations (50) and (51) are directly used to find $\{\dot{u}\}$ and $\{\dot{P}\}$ respectively in parallel on a node-by-node basis.
4. Estimates of $\{u\}$ and $\{P\}$ at time step t are calculated in parallel on a node-by-node basis using (52) and (53) respectively.
5. The pressure-averaging procedure is executed in parallel on a node-by-node basis.
6. Constraint equations are executed in parallel (see Note 2 below).
7. Steps 1–6 are repeated using the new estimates of the nodal values. The above iterative process can be terminated and control passed to step 8 if one of two stopping criteria are satisfied. The first criterion is met when the maximum difference between two consecutive nodal values is less than a certain tolerance. The second criterion is met when a certain fixed number of iterations (which give a stable solution and at the same time satisfy (55) are completed. Even though the second criterion is slower, it was found to be more robust and was adopted for the present study.

8. The initial nodal values are taken to be the new calculated values, time is incremented by Δt and steps 1–7 are repeated.

Notes

1. In modelling problems involving deforming fluid domains, such as fluid–structure interaction problems and free surface flow problems, equations (19), (20), (21), (36) and (37) must be used to evaluate $\{F_c^e\}$, $\{F_m^e\}$, $\{F_f^e\}$, $\{L_b^e\}$ and $\{L_f^e\}$ respectively. This is due to the fact that in that case the arrays C_{klmj}^e, Q_{kli}^e, K_{kli}^e, $\bar{C}_{klmj}^e$, $\bar{Q}_{kli}^e$, $\bar{K}_{kli}^e$, R_{kli}^e and $\bar{R}_{kli}^e$, which are functions of the element geometry, are no longer constant.
2. Constraints can be set on the nodal velocities and/or pressures. Typical velocity and pressure constraint equations have the forms $u_{Ij} = f(t)$ and $P_l = g(t)$ respectively, where t is time.

For a purely explicit scheme the critical time step is given by the CFL condition $s\Delta t_c/\Delta x \leqslant 1$. Note that the CFL condition is based on the sound speed and not the convective speed of the flow. This is due to the fact that the characteristic speed of flow using the artificial compressibility technique is the sound speed. The improvement in the critical time step over a purely explicit scheme using the present semi-explicit solution procedure is demonstrated in Figure 2. In order to compare the purely explicit and semi-explicit time steps, an effective time step for the semi-explicit procedure is defined as $\Delta t_{\text{effective}} = \Delta t_{\text{critical}}/R$. The ratio of this effective time step to the time step for a purely explicit method is plotted against the number of iterations, R, for the example problem of Section 7.3. Figure 2 shows that the maximum increase in the effective critical time step is achieved using the semi-explicit solution procedure with $R = 4$ and $p = 0{\cdot}36$. The CFL numbers found permissible in the present study are about 10. We note here that if the CFL condition were defined based on the physical characteristic speed, the permissible values would be of order unity.

In order to determine the parallel speed-up of the algorithm, the computation time was obtained for one to 16 processors on a Silicon Graphics Power Challenge (which is an MIMD/GSM machine) for the example problem of Section 7.3 with 2500 and 10,000 elements. The number of time steps and the number of equilibrium iterations within a time step were kept fixed for these tests. The speed-up versus the number of processors is plotted in Figure 3 for 2500 and 10,000 elements along with the theoretical parallel speed-up which is equal to the number of processors. For 10,000 elements the

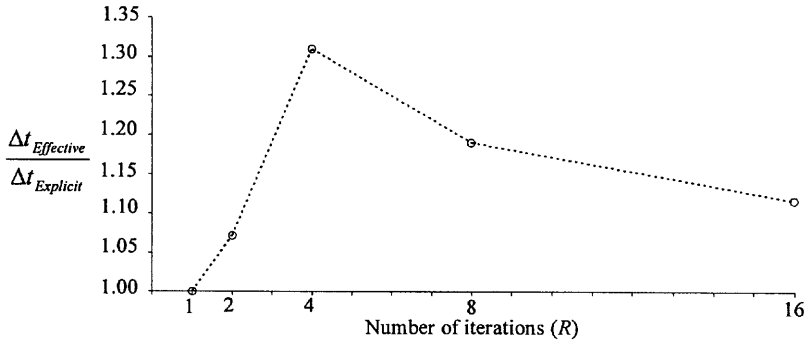

Figure 2. Improvement in critical time step over purely explicit critical time step $\Delta t_{\text{explicit}}$ by using sem-explicit solution procedure. $\Delta t_{\text{effective}} = \Delta t_{\text{critical}}/R$ and for all simulations $p^R = 0{\cdot}017$

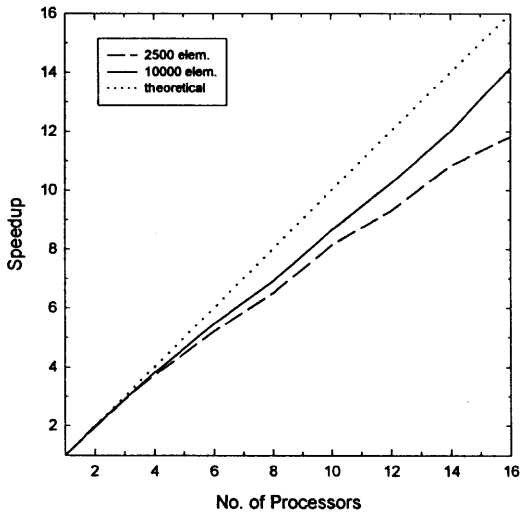

Figure 3. Parallel speed-up versus number of processors for 2500 and 10,000 elements

speed-up is nearly linear and is almost 14 for a 16-processor calculation. Also note the degradation of the speed-up for 2500 elements.[27]

7. EXAMPLES

The results for three benchmark fluid flow problems are given subsequently. The three problems are (i) steady state fluid flow over a backward-facing step, (ii) unsteady flow in an oscillating-lid-driven closed cavity and (iii) vortex shedding over a circular cylinder. The actual calculations are carried out in dimensional variables; however, they are reported using a Reynolds number defined as $Re = \rho UD/\mu$, where D and U are the lengths and velocity scales respectively.

An implementation of the formulation presented in this paper was written in ANSI C. The results in this section were computed on a six-processor SGI Onyx platform. Each processor was a 150 MHz MIPS R4400 processor with a peak capacity of 75 Mflops. All computational times reported in this section are for this configuration. The output of the programme consists of velocities, accelerations, pressures and rates of change of pressure at the nodes as a function of time. The input consists of the Cartesian nodal co-ordinates along with the elements connecting these nodes. Material properties of the elements are also supplied. Velocity and pressure constraints, initial conditions as well as body forces can also be specified.

Only isoparametric rectangular four-node Lagrangian elements are used; thus both pressure and velocity are interpolated using the conventional bilinear isoparametric interpolation.[28,29] The present formulation can accommodate most isoparametric element types. In a future paper we will investigate the use of higher-order elements (see e.g. References 24 and 30–33). An important advantage of using higher-order elements over linear elements is that generally fewer degrees of freedom are required for the same accuracy.[31]

In all the examples the following parameters were used: (i) artificial speed of sound such that the maximum artificial Mach number Ma was 0·1; (ii) number of equilibrium iterations was eight; (iii) implicitness parameter value was 0·6; (iv) time step such that the CFL number was about 10.

Backward-Facing Step

The flow in a planar channel with a backward-facing step is solved and compared with the experimentally determined flow of Armaly *et al.*[34] The distribution of velocity at the entrance is taken to be parabolic (fully developed channel flow). A schematic diagram of the channel ($l = 14h$) and the boundary conditions are shown in Figure 4 and the finite element mesh is shown in Figure 5. A characteristic of this flow is the recirculation region behind the step, after which the flow reattaches to the wall. A comparison of the computed and experimentally determined normalized distance x_r/h of the reattachment point from the edge of the step at steady state as a function of the Reynolds number ($Re = 50$–300) is shown in Figure 6. The Reynolds number is based on h and the average entrance velocity. The figure shows that the prediction from the present numerical technique matches closely the experimental results. Figure 7 shows the steady state flow streamlines for $Re = 150$.

Oscillating-Lid-driven Square Cavity

This example shows the accuracy of the present formulation in modelling dynamic flow problems in closed containers. The dynamic response of a viscous fluid inside a two-dimensional square container with a sinusoidally oscillating top lid was computed with the mesh shown in Figure 8. A no-slip condition is used at all the walls. The velocity of the top wall is given by $u = \sin(w't)$, where w' is the non-dimensional frequency of lid oscillation. Initially the fluid is motionless inside the entire domain. The results presented here are for a Reynolds number of 400 based on the container width, the peak lid velocity and $w' = 1{\cdot}0$. Figure 9 shows the history of the coefficient of friction at the upper wall, C_f, during the first six lid oscillations. Here C_f is given by

$$C_f = \int_0^1 [\partial u/\partial x]_{y=1} \mathrm{d}x.$$

At the sixth lid oscillation it is observed that C_f reaches a periodic steady state with an amplitude of 24. This flow was also solved by Iwatsu *et al.*[35] and the present result for C_f is in close agreement with their computed value of 25. Figures 10 and 11 show respectively the profiles of the horizontal velocity (at $x = 0{\cdot}5$) and vertical velocity (at $y = 0{\cdot}5$) during the sixth lid oscillation. Figure 12 shows snapshots of the streamlines also taken during the sixth lid oscillation. A total of 900 elements were used in this example. The time step used was $2\pi/592$ or 0·0106, corresponding to a CFL number of 10·6. The computation time per time step was about 0·15 s or 166 μs per time step node.

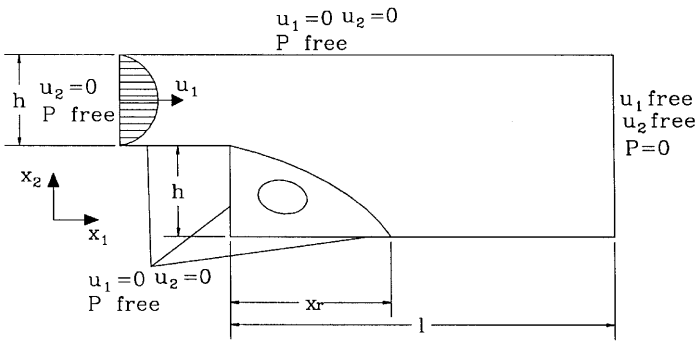

Figure 4. Schematic diagram of channel with backward-facing step

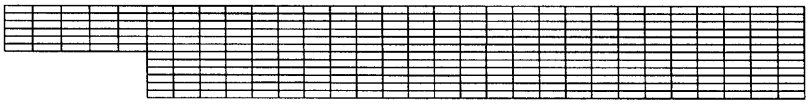

Figure 5. Finite element mesh of backward-facing step

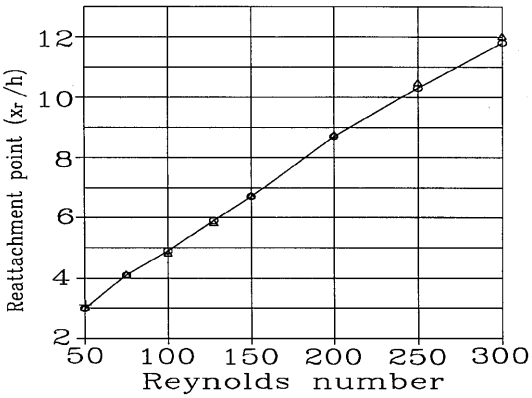

Figure 6. Variation of reattachment point with *Re* for backward-facing step. ○, experimental data; △, present computation

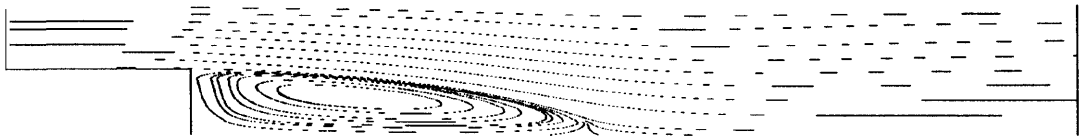

Figure 7. Streamlines for backward-facing step at $Re = 150$

Vortex Shedding over a Circular Cylinder

In the previous example the flow was driven by a time-dependent motion of one of the boundaries. In this example a time-dependent flow field is produced owing to instabilities in the flow over a circular cylinder, leading to vortex shedding. The Reynolds number is based on the cylinder diameter ($D = 0{\cdot}0016$ m), the upstream uniform velocity U, a density of 1 kg m^{-3} and a viscosity of 0·001 Pa s. The finite element mesh and the boundary conditions are shown in Figure 13. The vortex-shedding frequency f for the flow past a circular cylinder was measured experimentally by Rishko.[36] A relation between the Strouhal number and the Reynolds number for the flow past a cylinder for Reynolds number between 90 and 190 is given by $S = 0{\cdot}212(1 - 21{\cdot}2/Re)$, where the Strouhal number is defined as $S = fD/U$. Based on this relation, the variation of the shedding frequency with Reynolds number is plotted in Figure 14. The figure shows very good agreement between the experimentally measured vortex-shedding frequency and the one obtained using the present method. The results of Figure 14 were obtained using an artificial $Ma = 0{\cdot}1$ but were not altered when a lower value of $Ma = 0{\cdot}05$ is used. Figure 15 shows the periodic steady state history of the vertical component of velocity at a point behind the cylinder for $Re = 90$ and 130. The oscillation period is equal to the vortex-shedding period. Figure 16 shows snapshots of the stationary streamlines during the periodic steady state for $Re = 90$ and 130. A total of 2500 elements were used in this example. The time step

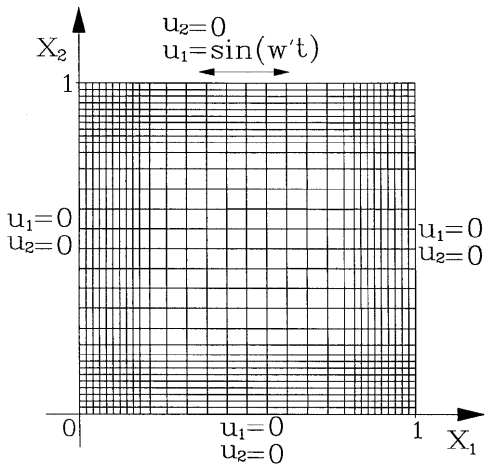

Figure 8. Geometry, finite element mesh and boundary conditions of square container

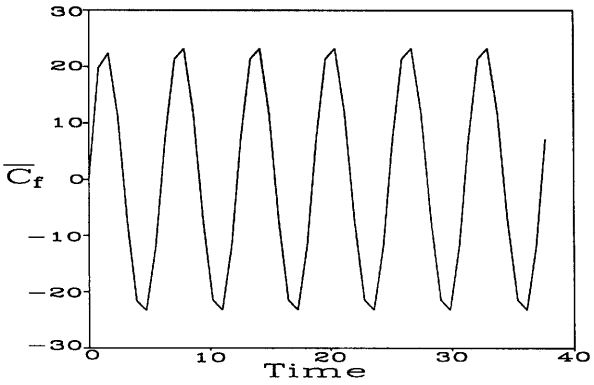

Figure 9. History of friction coefficient C_f

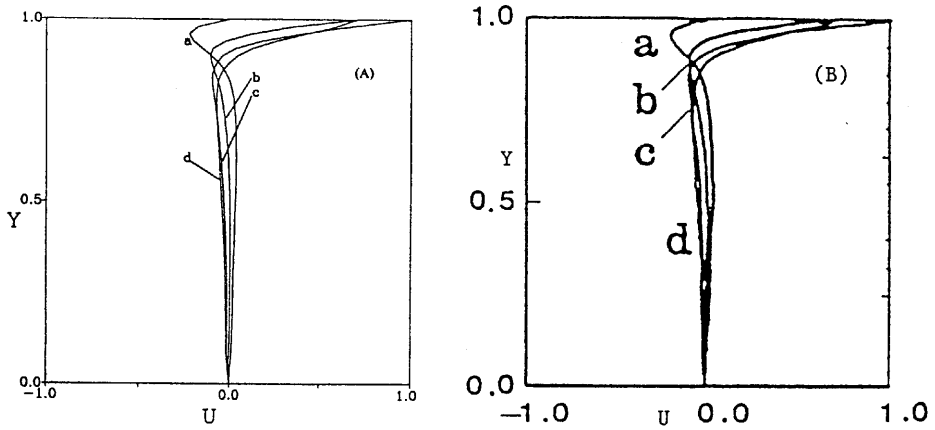

Figure 10. Profiles of horizontal velocity along vertical centreline, $x = 0{\cdot}5$, during one half-cycle of lid oscillation ($Re = 400$, $w' = 1{\cdot}0$): (A) present method; (B) Reference 35. Curves a–d are at the times 0, $T/8$, $T/4$ and $3T/8$ respectively, where $T = 2\pi/w'$ is the period of oscillation

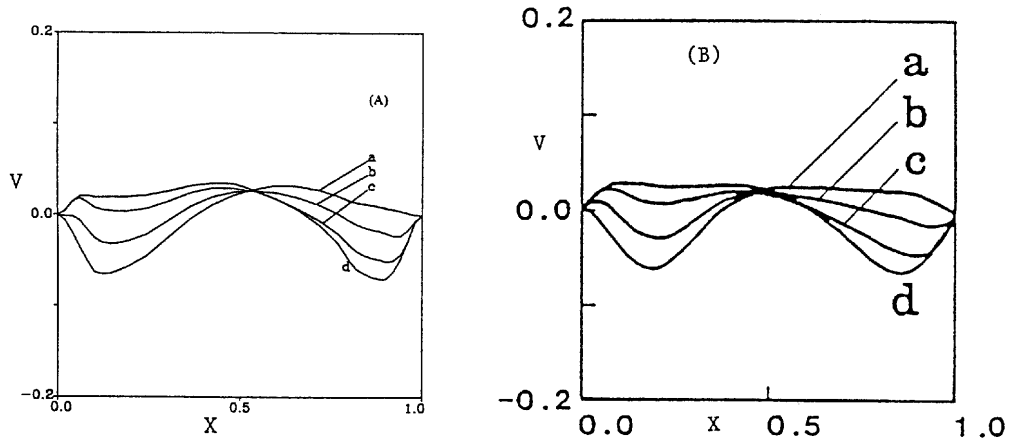

Figure 11. Profiles of vertical velocity along horizontal mid-height, $y = 0{\cdot}5$, during one half-cycle of lid oscillation ($Re = 400$, $w' = 1{\cdot}0$): (A) present method; (B) Reference 35. Curves a–d are at the times 0, $T/8$, $T/4$ and $3T/8$ respectively

was 0·0004 s, corresponding to a CFL number of 11·2. The inlet velocity was ramped up linearly from zero at time 0 s to the nominal flow speed at time 1 s. A small asymmetry in the flow velocity near the cylinder was introduced at time 0·5 s to promote vortex-shedding phenomena. A periodic steady state was obtained at time 3 s. The shedding period for $Re = 130$ was 0·112 s. The computation time was 0·4 s per time step or 160 μs per time step per node. Increasing the number of elements to 10,000 did not change the computation time per time step per node.

8. CONCLUSIONS

A finite element formulation for modelling unsteady incompressible laminar fluid flows was presented. Benchmark steady and unsteady fluid flow problems were solved to demonstrate the accuracy and features of the present method. The method is based on assembling global finite element vectors rather than matrices. As expected of such a procedure, it was found that the amount of computer core memory required was indeed proportional to the number of elements. A parallel speed-up that is nearly linear with the number of processors was achieved. Computation times were found to be nearly constant per time step per node regardless of the number of elements, verifying that computation times are indeed linearly proportional to the number of elements.

A parallel semi-explicit solution procedure with equilibrium iterations at each time step is used. The number of equilibrium iterations is related to an implicitness parameter. An increase in the number of equilibrium iterations permits a larger critical time step. Depending on the nature of the problem, an optimum time step and number of iterations exists which minimize the computing time. A new pressure-averaging technique is used which allows the use of equal-order interpolations for velocity and pressure. An artificial compressibility approach is employed allowing larger time steps while ensuring that the time-dependent flow is close to incompressible. Pressure averaging and artificial compressibility allow the use of the semi-explicit solution procedure for solving the incompressible Navier–Stokes equations and thus are an integral part of the procedure. For the problems investigated, a choice of eight equilibrium iterations and an implicitness parameter of 0·6 permitted CFL numbers of about 10 to be used.

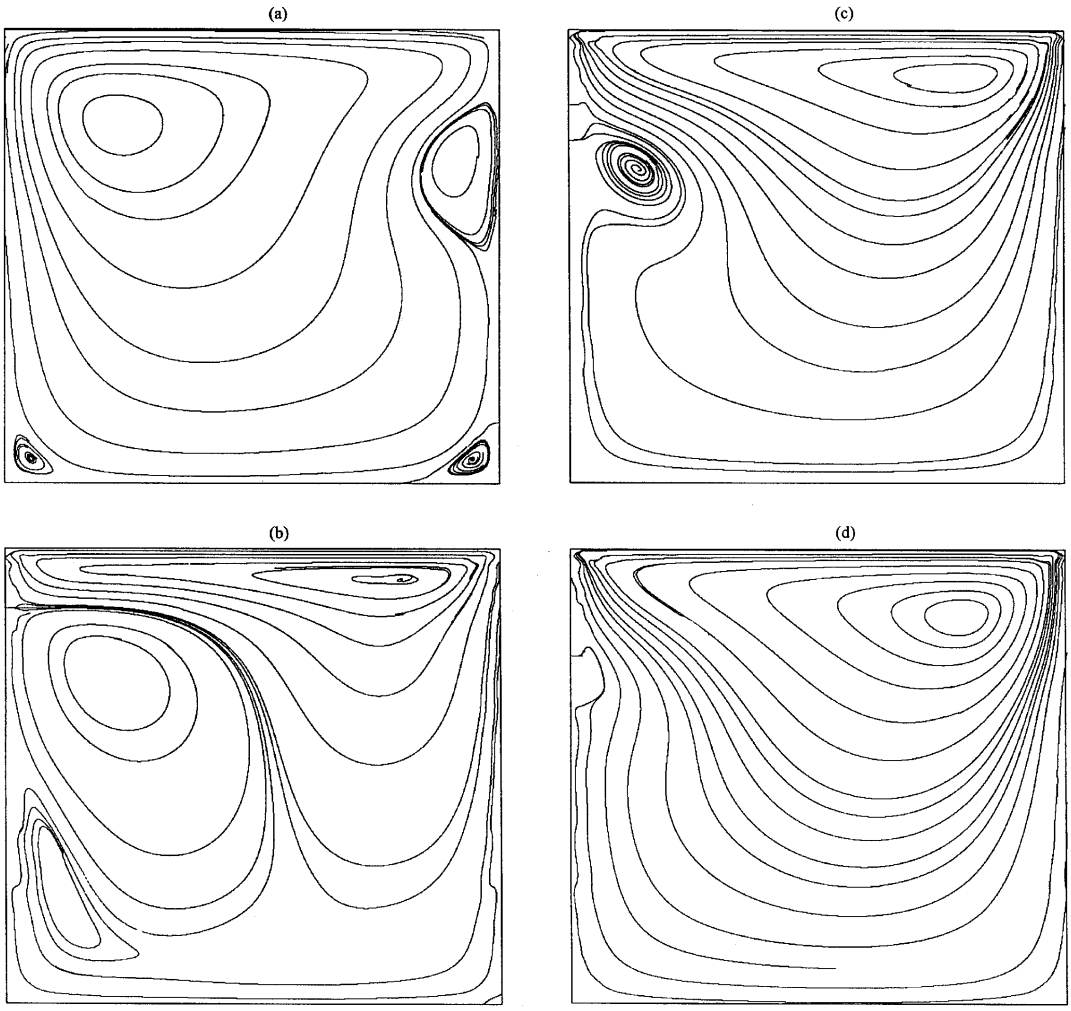

Figure 12. Snapshots of streamlines during periodic steady state at times (a) 0, (b) $T/8$, (c) $T/4$ and (d) $3T/8$

A standard Galerkin finite element approximation for all the terms in the Navier–Stokes equations was found adequate for the Reynolds numbers computed here, which are less than 500. Apparently the physical dissipation at these Reynolds numbers is sufficient to stabilize the convective term. This is unlikely to be the case at higher Reynolds numbers, where an upwind FEM formulation or explicit addition of artificial viscosity may become necessary.

ACKNOWLEDGEMENTS

This work was partially supported by the National Science Foundation under Grant CTS-93-15991. The authors would like to thank Dr. Michael Crow and Dr. Dimitri Anastassiou for use of the SGI Onyx machine at the Center for Telecommunication Research at Columbia University. The speed-up tests were carried out on a Silicon Graphics Power Challenge at the National Center for

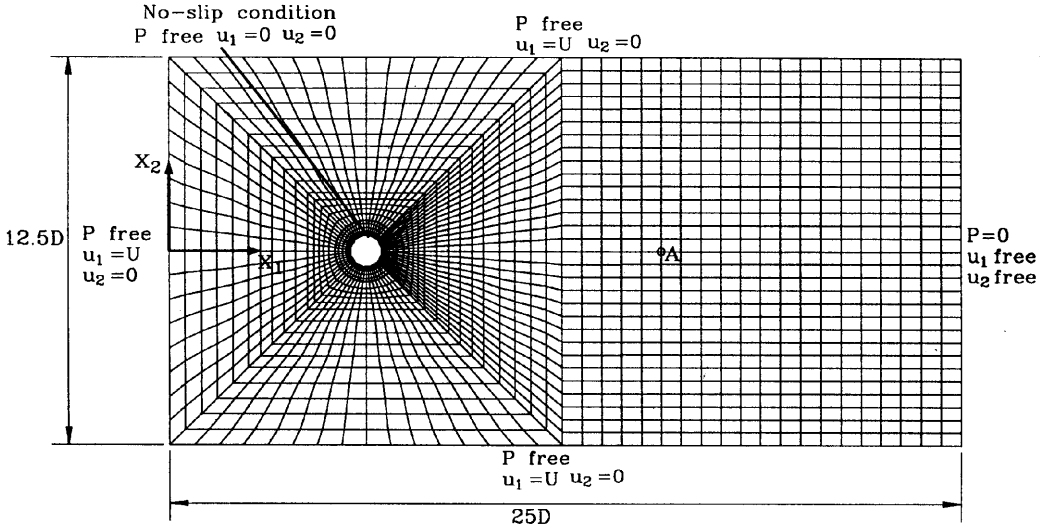

Figure 13. Finite element mesh and boundary conditions for flow past circular cylinder

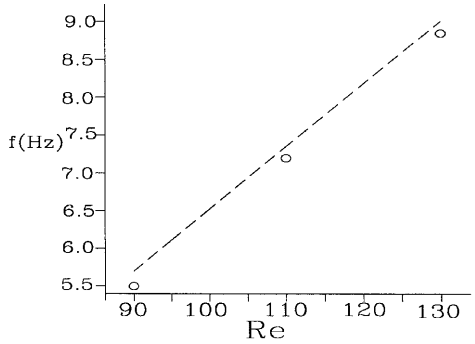

Figure 14. Variation of vortex-shedding frequency with Reynolds number: ---, from $S = 0{\cdot}212(1 - 21{\cdot}2/Re)$ proposed by Roshko;[36] ○, numerical simulation results

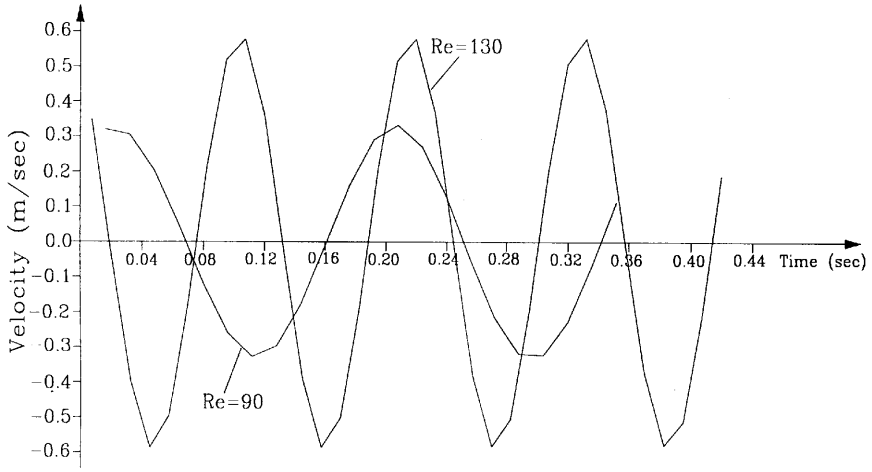

Figure 15. Vertical velocity versus time at point A ($x_1 = -0{\cdot}025, x_2 = 0{\cdot}0$) (of Figure 13)

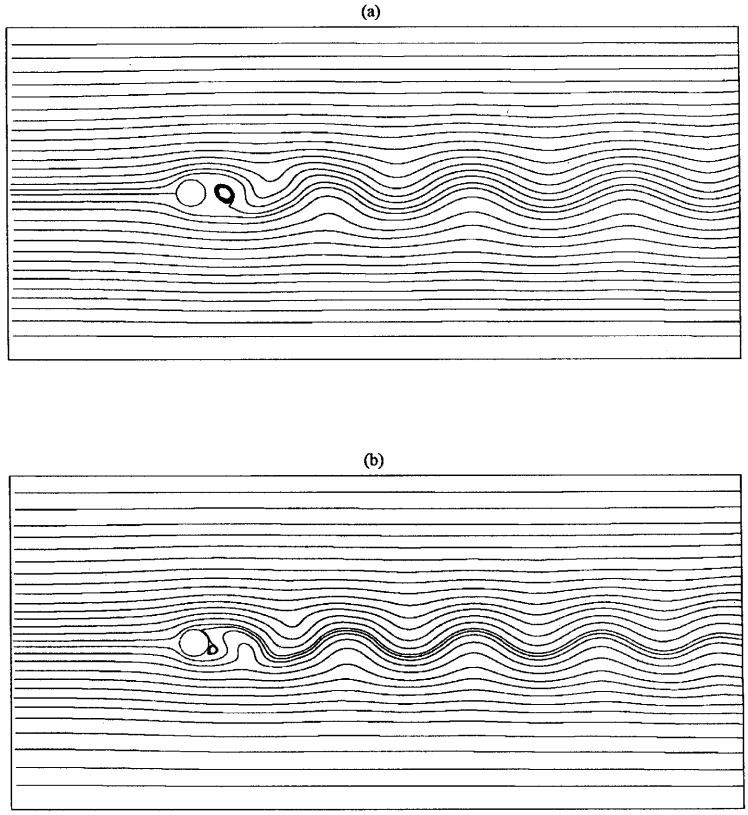

Figure 16. Snapshots of streamlines during periodic steady state: (a) $Re = 90$; (b) $Re = 130$

Supercomputing Applications (NCSA) at the University of Illinois at Urbana-Champaign under CTS950024N.

REFERENCES

1. P. F. Fischer and A. T. Patera, 'Parallel simulation of viscous incompressible flows', *Ann. Rev. Fluid Mech.*, **26**, 483–527 (1994).
2. O. C. Zienkiewicz and J. Wu, 'General explicit or semi-explicit algorithm for compressible and incompressible flows', *Int. j. numer. meth. engng.*, **35**, 457–479 (1992).
3. C. Farhat and L. Crivelli, 'Large scale FE parallel nonlinear computations using a homotopy method', in G. Rodrigue (ed.), SIAM, Philadelphia, PA, 1988, pp. 265–269.
4. C. Farhat and L. Crivelli, 'A general approach to nonlinear FE computations on shared memory multiprocessors', *Comput. Meth. Appl. Mech. Engng.*, **72**, 153–172 (1989).
5. M. E. 'Applications of parallel computing in computational fluid dynamics: a review', in H. Tyrar (ed.) *Advances in Distributed Memory and Parallel Processing*, Vol. 1, 1993.
6. N. M. Newmark, 'A method of computation for structural dynamics', *J. Engng. Mech. Div. ASCE*, **85**, 67–94 (1959).
7. I. Babuska, 'Error bounds for finite element method', *Numer. Math.*, **16**, 322–333 (1971).
8. F. Brezzi, 'On the existence, uniqueness and approximation of saddle point problems arising from Lagrange multipliers', *RAIRO, Ser. Rouge Anal. Numer.*, **8**, (1976).
9. T. J. R. Hughes, W. K. Liu and A. Brooks, 'Finite element analysis of incompressible viscous flows by the penalty function formulation', *J. Comput. Phys.*, **30**, 1–60 (1979).
10. R. L. Lee, P. M. Gresho and R. L. Sani, 'Numerical smoothing techniques applied to some finite element solutions of the Navier–Stokes equations', *Int. j. for numer. meth. engng.*, (1979)

11. T. J. R. Hughes, L. P. Franca and M. Mallet, 'A new finite element formulation for computational fluid dynamics: I. Symmetric forms of the compressible Euler and Navier Stokes equations and the second law of thermodynamics', *Comput. Meth. Appl. Mech. Engng.*, **54**, 223–234 (1986).
12. T. J. R. Hughes, L. P. Franca and M. Balestra, 'A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska–Brezzi condition: a stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations', *Comput. Meth. Appl. Mech. Engng.*, **59**, 85–99 (1986).
13. T. J. R. Hughes, L. P. Franca and Hulbert, 'A new finite element formulation for computational fluid dynamics: VIII. The Galerkin/least-squares method for advective–diffusive equations', *Comput. Meth. Appl. Mech. Engng.*, **73**, 173–189 (1989).
14. P. Hansbo and Szepessy, 'A velocity–pressure streamline diffusion finite element method for the incomrpessible Navier–Stokes equations', *Comput. Meth. Appl. Mech. Engng.*, **84**, 175–192 (1990).
15. L. P. Franca, T. J. R. Hughes, *et al.*, 'Stabilized finite element methods: I. Application to the advective–diffusive model', *Comput. Meth. Appl. Mech. Engng.*, **95**, 253–276 (1992).
16. T. E. Tezduyar, S. Mittal, S. E. Ray and R. Shih, 'Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity–pressure elements', *Comput. Meth. Appl. Mech. Engng.*, **95**, 221–242 (1992).
17. J. Donea, 'A Taylor–Galerkin method for convective transport problems', *Int. j. numer. meth. engng.*, **20**, 101–119 (1984).
18. J. Donea, T. Belytschko and P. Smolinski, 'A generalized Galerkin method for steady convection–diffusion problems with application to quadratic shape function elements', *Comput. Meth. Appl. Mech. Engng.*, **48**, 25–43 (1985).
19. J. J. R. Tamaddon, P. Townsend and M. F. Webster, 'Numerical solution of unsteady viscous flows', *Comput. Meth. Appl. Mech. Engng.*, **95**, 301–315 (1992).
20. A. J. Chorin, 'A numerical method for solving incompressible viscous flow problems', *J. Comput. Phys.*, **2**, (1967).
21. J. T. Oden, 'Optimal h–p finite element methods', *Comput. Meth. Appl. Mech. Engng.*, **112**, 309–331 (1994).
22. T. J. R. Hughes and T. E. Tezduyar, 'Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations', *Comput. Meth. Appl. Mech. Engng.*, **45**, 217–284 (1984).
23. A. N. Brooks and T. J. R. Hughes, 'Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on incompressible Navier–Stokes equations', *Comput. Meth. Appl. Mech. Eng.*, **32**, 199–259 (1982).
24. J. C. Heinrich and O. C. Zienkiewicz, 'The finite element method and ''upwinding'' technique in the numerical solution of convection dominated problems', *ASME Winter Ann. Meet., Finite Element Methods in Convection Dominated Flows*, ASME, New York, 1979.
25. I. Christie and A. R. Mitchell, 'Upwinding of high order Galerkin methods in conduction–convection problems', *Int. j. numer. meth. eng.*, **12**, 1764–1771 (1978).
26. K. J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1982.
27. Z. Johan, K. K. Mathur, S. L. Johnsson and T. J. R. Hughes, 'Scalability of finite element applications on distributed-memory parallel computers', *Comput. Meth. Appl. Mech. Engng.*, **119**, 61–72 (1994).
28. O. C. Zienkiewicz and R. L. Taylor, *The Finite Element Method*, 4th edn, McGraw-Hill, New York, 1989.
29. A. J. Baker, *Finite Element Computational Fluid Mechanics*, McGraw-Hill, New York, 1983.
30. J. C. Heinrich and O. C. Zienkiwicz, 'Quadratic finite element schemes for two-dimensional convective transport problems', *Int. j. numer. meth. engng.*, **11**, 1831–1844 (1977).
31. M. Bercovier and M. Engelman, 'A finite element for the numerical solution of viscous incompressible flows', *J. Comput. Phys.*, **30**, 181–201 (1979).
32. J. C. Heinrich, 'On quadratic elements in finite element solutions of steady state convection–diffusion equation', *Int. j. numer. meth. engng.*, **15**, 1041–1052 (1980).
33. R. Codina, O. Eugenio and M. Cervera, 'The intrinsic time for the streamline upwind/Petrov–Galerkin formulation using quadratic elements', *Comput. Meth. Appl. Mech. Eng.*, **94**, 239–262 (1992).
34. B. F. Armaly, F. Durst, J. C. Pereira and M. Yianneskis, 'Experimental and theoretical investigation of backward-facing step bow', *J. Fluid Mech.*, **127**, 473–496 (1983).
35. R. Iwatsu, J. M. Hyun and K. Kuwahara, 'Numerical simulation of flows driven by a torsionally oscillating lid in a square cavity', *J. Fluids Engng.*, **114**, 143–151 (1992).
36. A. Roshko, 'On the development of turbulent wakes from vortex streets', *NACA TN 2913*, 1953.
37. M. O. Bristeau, M. Mallet, J. Perriaux and G. Roge, 'Development of finite element method for compressible flow simulations in aerospace design', *AIAA Paper 90-0403*, 1990.

