INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 13, 403–423 (1991)

A FINITE ELEMENT SCHEME BASED ON THE VELOCITY CORRECTION METHOD FOR THE SOLUTION OF THE TIME-DEPENDENT INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

AGNES KOVACS* AND MUTSUTO KAWAHARA

Department of Civil Engineering, Chuo University, Kasuga 1-13-27, Bunkyo-ku, Tokyo 112, Japan

SUMMARY

In this paper a finite element solution for two-dimensional incompressible viscous flow is considered. The velocity correction method (explicit forward Euler) is applied for time integration. Discretization in space is carried out by the Galerkin weighted residual method. The solution is in terms of primitive variables, which are approximated by piecewise bilinear basis functions defined on isoparametric rectangular elements. The second step of the obtained algorithm is the solution of the Poisson equation derived for pressure. Emphasis is placed on the prescription of the proper boundary conditions for pressure in order to achieve the correct solution. The scheme is completed by the introduction of the balancing tensor viscosity; this makes this method stable (for the advection-dominated case) and permits us to employ a larger time increment. Two types of example are presented in order to demonstrate the performance of the developed scheme. In the first case all normal velocity components on the boundary are specified (e.g. lid-driven cavity flow). In the second type of example the normal derivative of velocity is applied over a portion of the boundary (e.g. flow through sudden expansion). The application of the described method to non-isothermal flows (forced convection) is also included.

KEY WORDS Velocity correction method Bilinear interpolation functions Pressure boundary conditions

INTRODUCTION

In recent years two-dimensional laminar flows of incompressible Newtonian fluids have been analysed in terms of different finite element schemes using the velocity–pressure formulation of the Navier–Stokes equations.

The primitive variable approach for steady flow problems was used by Hood and Taylor,[1] who applied mixed interpolation to the variables in order to avoid the difficulty encountered in equal-order interpolation.[2] Choosing interpolation functions similar to Reference 1, Kawahara *et al.*[3] utilized a finite element technique for the analysis of steady and unsteady flow problems. In the unsteady flow analysis, carried out with the perturbation method, the time derivative term was expressed by backward substitution (backward Euler). Huyakorn *et al.*[4] compared different types of mixed interpolation elements from the point of view of their numerical performance and generation of spurious pressure modes, obtaining results via solving the momentum and continuity equations simultaneously (as in References 1–3).

* Present address: SAFHL, University of Minnesota, Mississippi River at 3rd Ave. SE, Minneapolis, MN 55414, U.S.A.

0271–2091/91/140403–21$10.50

Received July 1988
Revised October 1990

Spurious pressure modes (caused by matrix singularities) were explained in detail by Sani *et al.*[5] They also presented a solution for the steady Stokes equations to which the penalty function method was applied. In this approximation the computation of the velocity field was carried out first. The pressure field was calculated from the continuity equation using the previously obtained velocity field as the second step of the algorithm. In the penalty function method, pressure modes still occurred.

The scheme presented by Gresho *et al.*[6] used the discretized approximation of the Laplacian operator for the calculation of the pressure. The diffusion matrix was singular, in this context, with respect to '$2\Delta x$' waves. This 'zero-energy mode', similar to the spurious checkerboard mode, was eliminated by an hour-glass correction term to the diffusion matrix. To obtain usable pressure results, it was necessary to introduce filtering and smoothing techniques[7] in the mixed interpolations contaminated by the spurious pressure mode.

In the method developed by Tuann and Olson[8] the velocity field was obtained from the momentum equations: a restricted variational principle was applied for the finite element formulation. The time derivatives were replaced with the backward difference formula and the new velocity field was obtained iteratively using the pressure value of the previous time step. From the new velocity distribution the pressure field was computed via the solution of the directly derived pressure Poisson equation. The boundary conditions for pressure were derived from the momentum equations in the solution of the pressure Poisson equation. Using an incomplete cubic interpolation function on triangular elements for both the velocities and the pressure, they encountered no problems in evaluating the pressure boundary conditions. The application of the derived pressure Poisson equation led to a simple and direct approximation of the Laplacian operator, which was necessarily constructed using the natural boundary conditions for pressure. There was no pressure mode other than the hydrostatic one.

Mizukami and Tsuchiya[9] also described a fractional step method (or segregated method[10]). Their procedure was based on the explicit Euler scheme. The procedure of Helmoltz's decomposition led to a Poisson equation for the ϕ-potential function. Thus it was necessary to give a boundary condition for ϕ. The prescription of the boundary condition of vanishing ϕ did not make it possible to use any boundary condition for the outlet other than constant p (e.g. $p=0$). This method exhibited no spurious pressure mode.

Gresho and Sani[11] analysed the pressure boundary conditions for both the consistent and the directly derived pressure Poisson equations. They discussed in detail the features of the use of either of these formulations from the perspective of well-posedness, solvability/consistency and the satisfaction of the continuity equation.

In the first step of the present method, time integration is carried out through the velocity correction method. This time integration scheme can be interpreted either as the first guess of the implicit/iterative method developed by Chorin[12] or a modification of the explicit forward Euler scheme used, for example, by Gresho *et al.*[6] For discretization in space the Galerkin weighted residual method is applied employing the bilinear interpolation function for both the velocities and the pressure. The second step of the algorithm is the solution of the directly derived, consistent-pressure Poisson equation.[11] To solve this equation, the Neumann boundary condition derived from the momentum equations is applied to the portion of the boundary where normal velocity is prescribed. For the computation of the natural boundary condition ($\partial p/\partial n$) the technique defined and demonstrated by Gresho *et al.*[13] is applied. If the normal velocity is given on the entire boundary (e.g. lid-driven cavity flow) and the pressure is prescribed at one node, the Neumann boundary condition ($\partial p/\partial n$) cannot be satisfied at the same node. The present method treats this pressure Dirichlet boundary condition in a flexible way so as to satisfy the natural boundary condition for the whole boundary. In flow-through problems the pressure Dirichlet boundary condition at the outlet is calculated via integration of the tangential derivative of

pressure ($\partial p/\partial \tau$) along this portion of the boundary. The balancing tensor viscosity is introduced similarly to Reference 6.

In the following sections this technique will be described in detail and demonstrated through numerical examples.

GOVERNING EQUATIONS AND BOUNDARY CONDITIONS

The governing equations here are the two-dimensional unsteady Navier–Stokes equations for an incompressible constant-property fluid. In dimensional form these are

$$\rho[\partial \mathbf{u}/\partial t+(\mathbf{u}\cdot\nabla)\mathbf{u}]+\nabla p=\mu\nabla^2\mathbf{u}+\rho\mathbf{g}, \tag{1}$$

$$\nabla\cdot\mathbf{u}=0, \tag{2}$$

where t is the time (s), $\mathbf{u}=\mathbf{u}(\mathbf{x}, t)=(u(x, y, t), v(x, y, t))$ is the velocity, with Cartesian components in two (x, y) directions (m s^{-1}), $p=p(\mathbf{x}, t)=p(x, y, t)$ is the pressure (N m^{-2}), ρ is the density (kg m^{-3}), μ is the dynamic viscosity (kg m^{-1} s^{-1}), $\nu=\mu/\rho$ is the kinematic visocity (m^2 s^{-1}) and $\mathbf{g}=(g_x, g_y)$ is the gravitational acceleration (m s^{-2}). Boundary conditions (Dirichlet BCs) are

$$\mathbf{u}=\hat{\mathbf{u}}(\mathbf{x}, t) \quad \text{on } \Gamma, \quad \text{with} \quad \int_\Gamma \mathbf{n}\cdot\hat{\mathbf{u}}\,d\Gamma=0, \tag{3}$$

where $\hat{\mathbf{u}}$ is a known function on the entire boundary Γ of the domain Ω and $\mathbf{n}$ is the outward-pointing normal vector on Γ. Initial conditions are

$$\mathbf{u}=\mathbf{u}(\mathbf{x}, t=0)=\mathbf{u}_0(\mathbf{x}) \quad \text{in } \Omega\cup\Gamma, \quad \text{with } \nabla\cdot\mathbf{u}_0=0 \quad \text{in } \Omega\cup\Gamma, \tag{4}$$

where $\mathbf{u}_0$ is a prescribed function.

TIME INTEGRATION

For discretization in time the velocity correction method is employed. This method was applied and explained in detail in References 14 and 15 as well. The obtained semidiscretized scheme is as follows.

Step 1

Calculation of the 'intermediate-velocity' field—apply the explicit Euler first-order scheme for equation (1) and omit the pressure (∇p^n) and the gravitational ($\rho\mathbf{g}$) terms:

$$\tilde{\mathbf{u}}=\mathbf{u}^n-\Delta t[-\nu\nabla^2\mathbf{u}^n+(\mathbf{u}^n\cdot\nabla)\mathbf{u}^n]. \tag{5}$$

Step 2

Solution of the pressure Poisson equation—substitute the actual $\mathbf{u}^{n+1}$ $(=\tilde{\mathbf{u}}-(\Delta t/\rho)\times(\nabla p^n-\rho g))$ into equation (2) to satisfy the incompressibility constraint:

$$\nabla^2 p^n=(\rho/\Delta t)\nabla\cdot\tilde{\mathbf{u}}. \tag{6}$$

Step 3

Correction of the 'intermediate-velocity' field—add the terms neglected in equation (5) to the 'intermediate-velocity' field:

$$\mathbf{u}^{n+1}=\tilde{\mathbf{u}}-(\Delta t/\rho)(\nabla p^n-\rho\mathbf{g}). \tag{7}$$

The superscript n indicates the number of the time step. Δt is the time increment. It can be seen that for the solution of equation (6) it is necessary to introduce additional boundary conditions for pressure.

SPATIAL DISCRETIZATION

The finite element discretization of equations (5)–(7) is performed using the Galerkin weighted residual method via the following expansions in the piecewise polynomial basis functions associated with the FEM:

$$\mathbf{u}(\mathbf{x}, t)=\sum_{i=1}^{N} \mathbf{u}_i(t)\varphi_i(\mathbf{x}), \tag{8a}$$

$$p(\mathbf{x}, t)=\sum_{i=1}^{N} p_i(t)\varphi_i(\mathbf{x}), \tag{8b}$$

where in the discretized domain there are N nodes for velocity and pressure. The weak form of equations (5)–(7) permits φ_i to be discontinuous in the first derivatives and introduces natural boundary conditions. Thus $\varphi_i(\mathbf{x})$ is chosen to be a C^0 piecewise bilinear basis function defined on isoparametric rectangular elements. Inserting (8) into the weak form of (5)–(7) leads to the discretized system of equations, which can be written in matrix form for the whole domain.

Step 1

$$\mathbf{M}\tilde{\mathbf{u}}=\mathbf{M}\mathbf{u}^n-\Delta t\nu\mathbf{S}\mathbf{u}^n-\Delta t\mathbf{A}^n\mathbf{u}^n+\Delta t\hat{\mathbf{\Omega}}_u^n. \tag{9}$$

Step 2

$$\mathbf{S}\mathbf{p}^n=(\rho/\Delta t)\mathbf{D}\tilde{\mathbf{u}}+\hat{\mathbf{\Omega}}_p^n. \tag{10}$$

Step 3

$$\mathbf{M}\mathbf{u}^{n+1}=\mathbf{M}\tilde{\mathbf{u}}-(\Delta t/\rho)(\mathbf{D}\mathbf{p}^n-\rho\mathbf{M}\mathbf{g}). \tag{11}$$

Here $\mathbf{u}^{n+1}$ and $\mathbf{u}^n$ are now global vectors containing all nodal values of u and v at the $(n+1)$th and nth time steps respectively (e.g. $\mathbf{u}^{nT}=(\mathbf{u}^{nT}, \mathbf{v}^{nT})$, with superscript T denoting transpose), $\tilde{\mathbf{u}}$ is the global vector for the 'intermediate-velocity' field, $\mathbf{p}^n$ is the global pressure vector containing all nodal values of p at the nth time step (note that p is defined on the same nodes as $\mathbf{u}$), $\hat{\mathbf{\Omega}}_u^n$ and $\hat{\mathbf{\Omega}}_p^n$ are the vectors of natural boundary conditions for velocity and pressure respectively (detailed in the next section), $\mathbf{M}$ is the mass matrix (which is lumped via row sum at element level), $\mathbf{S}$ is the diffusion or Laplacian matrix, $\mathbf{D}$ is the divergence matrix ($\mathbf{D}\tilde{\mathbf{u}}=\mathbf{D}_x\tilde{\mathbf{u}}+\mathbf{D}_y\tilde{\mathbf{v}}$) and $\mathbf{A}^n=\mathbf{A}(\mathbf{u}^n)$ is the advection matrix.

The element matrices associated with equations (9)–(11) are evaluated as

$$\mathbf{M}^e=\int\int_{\Omega^e}\boldsymbol{\phi}\boldsymbol{\phi}^{\mathrm{T}}\,\mathrm{d}x\,\mathrm{d}y \quad \text{(without lumping)}, \tag{12a}$$

$$\mathbf{S}^e=\int\int_{\Omega^e}[(\partial\boldsymbol{\phi}/\partial x)(\partial\boldsymbol{\phi}^{\mathrm{T}}/\partial x)+(\partial\boldsymbol{\phi}/\partial y)(\partial\boldsymbol{\phi}^{\mathrm{T}}/\partial y)]\,\mathrm{d}x\,\mathrm{d}y, \tag{12b}$$

$$\mathbf{D}_x^e=\int\int_{\Omega^e}\boldsymbol{\phi}(\partial\boldsymbol{\phi}^{\mathrm{T}}/\partial x)\,\mathrm{d}x\,\mathrm{d}y, \qquad \mathbf{D}_y^e=\int\int_{\Omega^e}\boldsymbol{\phi}(\partial\boldsymbol{\phi}^{\mathrm{T}}/\partial y)\,\mathrm{d}x\,\mathrm{d}y, \tag{12c}$$

$$\mathbf{A}^{en} = \int\int_{\Omega^e} \boldsymbol{\phi}[\mathbf{u}^{nT}\boldsymbol{\phi}(\partial\boldsymbol{\phi}^T/\partial x) + \mathbf{v}^{nT}\boldsymbol{\phi}(\partial\boldsymbol{\phi}^T/\partial y)]\,\mathrm{d}x\mathrm{d}y, \tag{12d}$$

where $\boldsymbol{\phi}^T = (\varphi_1, \varphi_2, \varphi_3, \varphi_4)$ is the vector of basis functions[16] for element e. The algebraic system of equations (9)–(11) together with the boundary conditions can then be solved. In the case of the diagonal lumped mass matrix the inversion is trivial and the computation evaluates the 'intermediate-velocity' field via equation (11) node by node. The element matrices (12a)–(12c) and the lumped mass matrix are generated once during a computation and retrieved every time they are needed. The advection matrix (12d) is calculated at every time step. For numerical integration the Gaussian quadrature formula (for $n=2$)[16] is utilized. For equation (10) the skyline solver is applied.

BOUNDARY CONDITIONS FOR THE DERIVED ALGORITHM

Two cases (A and B) are considered.

Step 1

The last term on the right-hand side of equation (9) includes the natural BC for velocity.

Case A. The normal velocity, via equation (3), is prescribed over the entire boundary (e.g. lid-driven cavity flow):

$$\hat{\boldsymbol{\Omega}}_u^n = 0 \quad \text{on } \Gamma. \tag{13a}$$

Case B. The normal velocity, via equation (3), is not prescribed over a portion of the boundary (e.g. flow-through problems). The normal derivative of velocity is specified on Γ_2 such that

$$\hat{\boldsymbol{\Omega}}_u^{nT} = (\hat{\mathbf{f}}_x^{nT}, \hat{\mathbf{f}}_y^{nT}) \quad \text{on } \Gamma_2, \tag{13b}$$

where

$$\hat{\mathbf{f}}_x^n = \nu \int_{\Gamma_2} \boldsymbol{\phi}(\partial u^n/\partial n)\,\mathrm{d}\Gamma, \qquad \hat{\mathbf{f}}_y^n = \nu \int_{\Gamma_2} \boldsymbol{\phi}(\partial v^n/\partial n)\,\mathrm{d}\Gamma \tag{13c}$$

and $\Gamma_1 \cup \Gamma_2 = \Gamma$. In this case on the Γ_1-boundary the normal velocity is given via equation (3). Here n is the direction of the outward-pointing normal vector on the boundary. $\partial u^n/\partial n$ and $\partial v^n/\partial n$ are approximated by the derivative at the centre of the corresponding element.

Step 2

The last term on the right-hand side of equation (10) contains the natural BC for the pressure Poisson equation. Gresho and Sani[11] pointed out the significance of the proper boundary conditions for pressure.

Case A (same as Case A for Step 1)

$$\hat{\boldsymbol{\Omega}}_p^n = \int_{\Gamma_{M-1}} \boldsymbol{\phi}(\partial p^n/\partial n)\,\mathrm{d}\Gamma \quad \text{on } \Gamma_{M-1}, \tag{14}$$

with $\Gamma_{M-1} \cup \Gamma_M = \Gamma$, where Γ_{M-1} has $M-1$ nodes, while Γ_M has the Mth node on it. In

equation (14) M is the number of nodes on the boundary and

$$\partial p^n/\partial n = \mathbf{n}\cdot\nabla p^n = \rho\{\nu\nabla^2 u_n^n - [(\partial u_n^n/\partial t) + (\mathbf{u}^n\cdot\nabla)u_n^n]\} \tag{15}$$

is the consistent Neumann boundary condition. For walls with no penetration $\partial u_n^n/\partial t = 0$. Equations (15) and (14) can be considered as 'Step 0' of the algorithm. The Dirichlet BC is prescribed at one node of the boundary:

$$\hat{p} = 0 \quad \text{on } \Gamma_M, \tag{16}$$

in other words at the Mth node. $\hat{p}$ is the hydrostatic pressure mode, an arbitrary additive constant defining the corresponding pressure.

Case B (same as Case B for Step 1). The natural BC with equations (14) and (15) is prescribed on Γ_1. On Γ_2 the Dirichlet BC for pressure is to be calculated through the integration of its tangential derivative:

$$\hat{p}^n = \int_{\Gamma_2} (\partial p^n/\partial\tau)\,\mathrm{d}\Gamma \quad \text{on } \Gamma_2, \tag{17}$$

where

$$\partial p^n/\partial\tau = \boldsymbol{\tau}\cdot\nabla p^n = \rho\{\nu\nabla^2 u_\tau^n - [(\partial u_\tau^n/\partial t) + (\mathbf{u}^n\cdot\nabla)u_\tau^n]\}. \tag{18}$$

The solution of equations (15) and (18) will be explained in the next section. Note that equations (15) and (18) apply under the constraint that $L/R \ll 1$, where L is a length scale characterizing the size of Ω and R is a scale characterizing the raidus of curvature of Γ.

Step 3

For equation (11) the Dirichlet BC for velocity is to be introduced (via equation (3)):

$$\hat{\mathbf{u}} = \hat{\mathbf{u}}(\mathbf{x}, (n+1)\Delta t) \quad \begin{cases} \text{on } \Gamma & \text{(for Case A)} \\ \text{on } \Gamma_1 & \text{(for Case B).} \end{cases} \tag{19}$$

'STEP 0' OF THE ALGORITHM

The solution of equations (15) and (18) is based on the method described in Reference 13. This technique is applied for the computation of derived boundary quantities. In the present case the normal ($\partial p^n/\partial n$) and tangential ($\partial p^n/\partial\tau$) components of equation (1) are computed as force or flux to be determined on the boundary. The values of u_n^n or u_τ^n and vector $\mathbf{u}^n$ are known from the previous time step. The calculation of $\partial p^n/\partial n$ will be shown in the rest of this section.

Construction of the weak form of equation (15) begins with

$$\int_\Omega \psi_i p_n^n\,\mathrm{d}\Omega = \int_\Omega \psi_i \rho\{\nu\nabla^2 u_n^n - [(\partial u_n^n/\partial t) + (\mathbf{u}^n\cdot\nabla)u_n^n]\}\,\mathrm{d}\Omega, \quad i = 1, 2, \ldots, M-1, \tag{20a}$$

where ψ_i is the test function, non-zero only at the boundary nodes, and $p_n^n = \partial p^n/\partial n$ is the pressure gradient. In accordance with the Galerkin method p_n^n is expanded in the $\{\psi_i\}$-basis as

$$p_n^n = \sum_{i=1}^{M-1} p_{ni}^n \psi_i(x), \tag{20b}$$

where p_{ni}^n is to be determined. Integrating equation (20a) by parts and inserting (20b) and (8a) into

the weak form of equation (20a), the following algebraic system is achieved to evaluate p_{ni}^n on Γ_{M-1}:

$$\mathbf{M}^{**}\mathbf{p}_n^n = -\rho\mathbf{M}^*(\mathbf{u}_n^n - \mathbf{u}_n^{n-1})/\Delta t - \rho\nu\mathbf{S}^*\mathbf{u}_n^n - \rho\mathbf{A}^*\mathbf{u}_n^n + \rho\hat{\mathbf{\Omega}}_n^*, \tag{21}$$

where $\mathbf{M}^*$, $\mathbf{S}^*$ and $\mathbf{A}^*$ are similar to $\mathbf{M}$, $\mathbf{S}$ and $\mathbf{A}$ in equation (9), $\mathbf{M}^{**}$ is the boundary mass matrix and $\hat{\mathbf{\Omega}}_n^*$ contains the normal derivative of velocity of Γ_{M-1} (as in equation (13)). The contributions to equation (21) can be formed in the usual way (at element level) except that only those elements with nodes on Γ_{M-1} need to be considered. ψ_i, similarly to φ_i, is chosen to be a C^0 piecewise bilinear basis function (thus $\mathbf{M}^{**}=\mathbf{M}^*$). The calculation of the tangential component $(\partial p^n/\partial\tau)$ can be carried out in the same way. For the computation of $\hat{p}$ in the case of an open boundary Shimura and Kawahara[14,17] suggested a different procedure.

INTRODUCTION OF THE BALANCING TENSOR VISCOSITY

The above-prescribed scheme is a modification of the explicit forward Euler method. Thus it has the same difficulty associated with negative diffusivity as the technique developed by Gresho *et al.*[6] That is, for advection-dominated flows the stability limit on Δt is very restrictive and for pure advection the method is unconditionally unstable. A Taylor series analysis of equation (1) (with the omission of the pressure and gravitational terms) leads to the introduction of an additional tensor viscosity to balance the reduction of the effective viscosity. Thus 'Step 1' of the algorithm is to solve

$$\tilde{\mathbf{u}} = \mathbf{u}^n - \Delta t[-\nabla\cdot\boldsymbol{\nu}^*\nabla\mathbf{u}^n + (\mathbf{u}^n\cdot\nabla)\mathbf{u}^n] \tag{22a}$$

instead of equation (5), where ν is evaluated for each element separately as

$$\boldsymbol{\nu}^* = \begin{bmatrix} \nu + (\Delta t/2)\bar{u}^n\bar{u}^n & (\Delta t/2)\bar{u}^n\bar{v}^n \\ (\Delta t/2)\bar{u}^n\bar{v}^n & \nu + (\Delta t/2)\bar{v}^n\bar{v}^n \end{bmatrix}. \tag{22b}$$

The elementwise-calculated $\boldsymbol{\nu}^*$ viscosity tensor is symmetric and constant. $\bar{u}^n$ and $\bar{v}^n$ are the average velocities in the element (arithmetic average of the nodal values). For further details on the balancing tensor viscosity see Reference 6.

RELAXATION OF '$\hat{p}=0$' DIRICHLET BOUNDARY CONDITION

In order to explore the features of the method developed here, the lid-driven cavity flow problem for $Re=100$ has been selected as the first test of the algorithm. The results are obtained on a graded mesh of 25×25 nodes. The sketch of Figure 1 shows the boundary conditions. Other parameters of the computation are $\mu=10$, $\rho=1000$, $\mathbf{g}=0$, $\Delta t=0{\cdot}005$ and NDT (number of time steps)$=6000$. The time increment (Δt) is chosen in consideration of the stability limit given by Gresho *et al.*[6] for the improved (by BTV) scheme. The steady state is reached after about 3000 time steps. The obtained pressure distribution is presented in Figure 2.

Upon examining this pressure distribution, it may be noted that the prescription of the $\hat{p}=0$ Dirichlet boundary condition has caused a local pressure drop (or peak) at and around the node where it is prescribed. This local pressure drop (or peak) is the consequence of the fact that at this node the Neumann boundary condition is not satisfied. Following the notation of Figure 3, the value of this local pressure drop (or peak) is estimated as

$$\partial p = (p_2 + 0{\cdot}5p_3 + p_4)/2{\cdot}5 - p_1 \tag{23}$$

where p_2, p_3, p_4 and p_1 are the pressure values computed at nodes 2, 3, 4 and 1 respectively. The

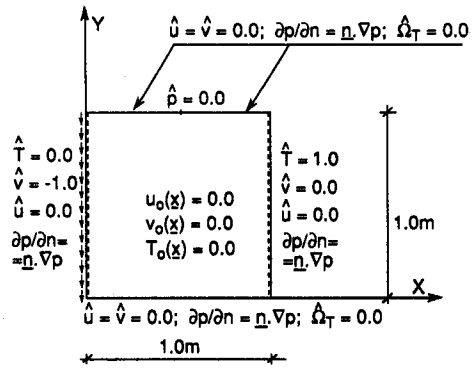

Figure 1. Lid-driven cavity flow; initial and boundary conditions

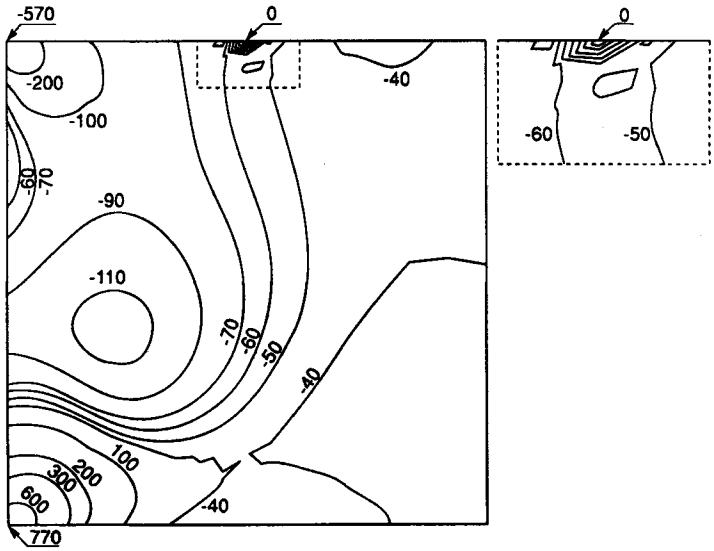

Figure 2. Pressure distribution; $Re = 100$, $\hat{p} = 0$

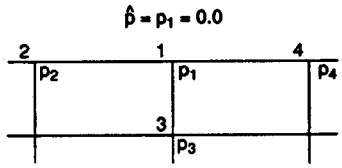

Figure 3. Calculation of the local pressure discrepancy

local pressure discrepancy for the case $\hat{p} = 0$ is shown in Figure 4(a). It is denoted 'dissat ∂p' and calculated from equation (23).

To satisfy the natural boundary condition on the whole boundary, the condition $\hat{p} = 0$ is relaxed and the value of $\hat{p}$ is extrapolated from the pressure values of the adjacent nodes at every

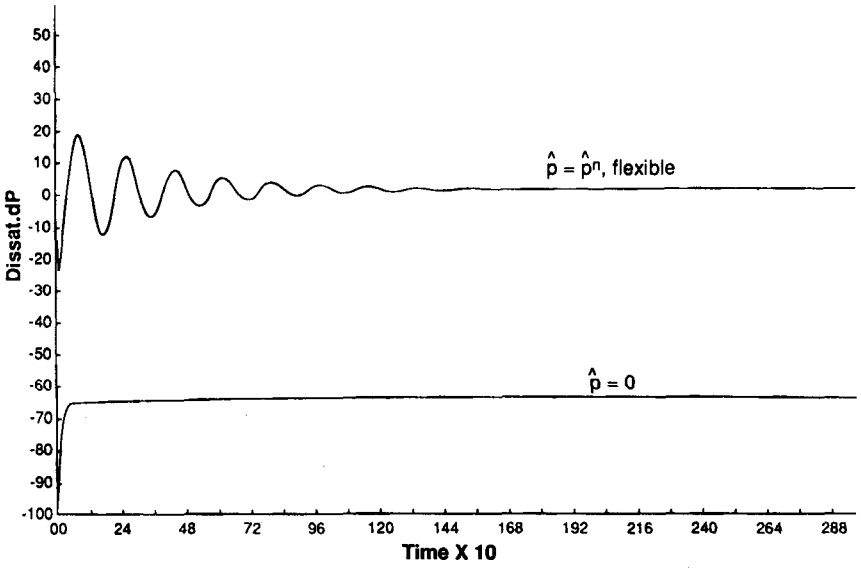

Figure 4(a). Time history of 'dissatisfied ∂p' at the node of $\hat{p}$; $Re = 100$

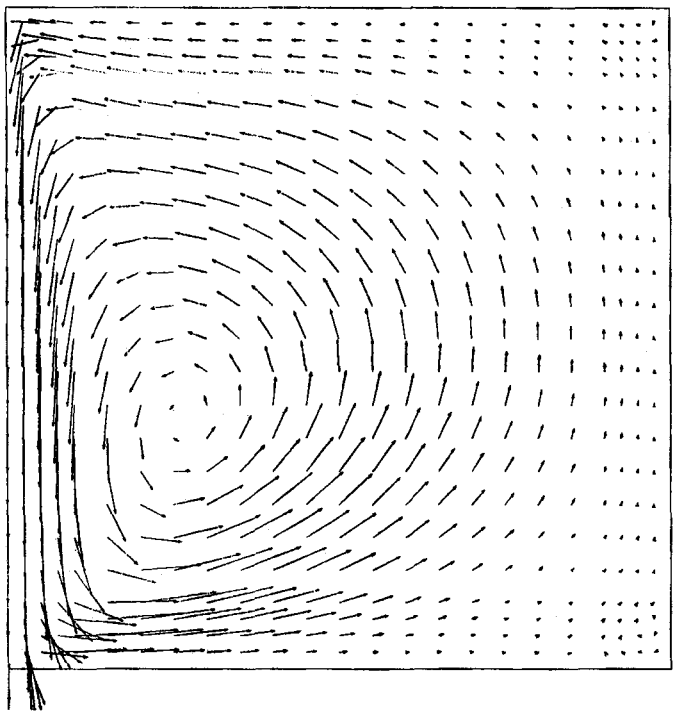

Figure 4(b). Velocity distribution; $Re = 100$, $\hat{p}$ is flexible

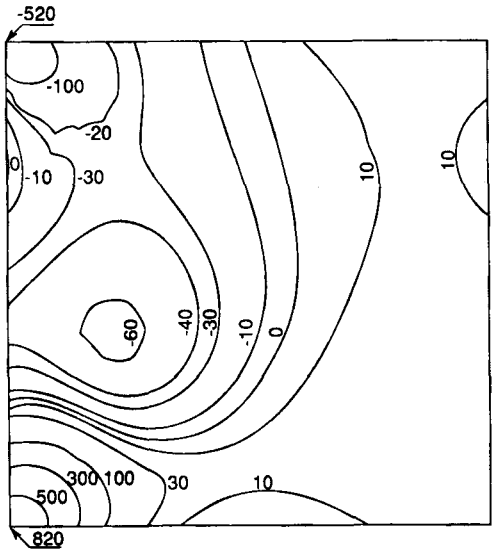

Figure 4(c). Pressure distribution; $Re=100$, $\hat{p}$ is flexible

time step. Thus in this computation

$$\hat{p}^n=p_1^n=(p_2^{n-1}+0{\cdot}5p_3^{n-1}+p_4^{n-1})/2{\cdot}5+p_t, \tag{24a}$$

where

$$p_t=(p_2^{n-1}+p_4^{n-1}-p_2^{n-2}-p_4^{n-2})/2{\cdot}0. \tag{24b}$$

$\hat{p}^n$ denotes the extrapolated value of $\hat{p}$ for $t=n\Delta t$. The local pressure deviation for this new flexible Dirichlet BC is also presented in Figure 4(a). For this case 'dissat ∂p' is seen to converge to zero. Figures 4(b) and 4(c) show the velocity and pressure distributions respectively. This pressure field is normalized for $\hat{p}^n=0$ and it can be seen that the local pressure drop has disappeared.

APPLICATION TO NON-ISOTHERMAL FLOWS

Applying the Boussinesq approximation, the gravitational term on the right-hand side of equation (1) is modified and the energy equation is introduced in terms of temperature. Thus

$$\rho\mathbf{g}\equiv\rho_0[1{\cdot}0-\beta_0(T-T_0)]\mathbf{g}, \tag{25a}$$

$$\partial T/\partial t+(\mathbf{u}\cdot\nabla)T=\alpha\nabla^2T, \tag{25b}$$

where T is the temperature (°C or K), α is the thermal diffusivity ($m^2\,s^{-1}$), β_0 is the thermal expansion coefficient ($°C^{-1}$ or K^{-1}) and ρ_0 is the density ($kg\,m^{-3}$) at the reference temperature T_0 (°C or K). Boundary conditions for equation (25b) are

$$T=\hat{T}(t) \quad \text{on } \Gamma_3, \tag{25c}$$

$$\begin{aligned} q=\hat{q}(t)=\alpha\mathbf{n}\cdot\nabla T \quad &\text{on } \Gamma_4 \quad \text{(outflow boundary)},\\ q=\hat{q}(t)=0 \quad &\text{on } \Gamma_5 \quad \text{(no-slip boundary)}, \end{aligned} \tag{25d}$$

where $\Gamma_3\cup\Gamma_4\cup\Gamma_5=\Gamma$. Initial conditions are

$$T=T(\mathbf{x},t=0)=T_0(\mathbf{x}) \quad \text{in } \Omega\cup\Gamma. \tag{25e}$$

The spatial discretization of equation (25b) is performed using the Galerkin weighted residual method via the following expansion:

$$T(\mathbf{x}, t)=\sum_{i=1}^{N} T_i(t)\varphi_i(\mathbf{x}), \tag{26}$$

where N, the number of nodes, is the same as that in equation (8), because $\varphi_i(\mathbf{x})$ is again chosen to be a C^0 piecewise bilinear basis function. Inserting (26) into the weak form of equation (25b) and introducing the natural boundary conditions, the following GFEM equations are obtained:

$$\mathbf{M}(\partial\mathbf{T}/\partial t)+(\mathbf{A}+\alpha\mathbf{S})\mathbf{T}=\hat{\mathbf{\Omega}}_q, \tag{27}$$

where matrices **M**, **A** and **S** are the same as those of equation (9).

For integration in time the explicit forward Euler scheme is applied; thus

$$\mathbf{M}\mathbf{T}^{n+1}=\mathbf{M}\mathbf{T}^n-\Delta t[\mathbf{A}(\mathbf{u}^n)+\alpha\mathbf{S}]\mathbf{T}^n+\Delta t\hat{\mathbf{\Omega}}_q^n. \tag{28a}$$

The vector $\hat{\mathbf{\Omega}}_q^n$ incorporates the natural boundary condition such that

$$\hat{\mathbf{\Omega}}_q^n=\int_{\Gamma_4}\boldsymbol{\phi}\hat{q}^n\,\mathrm{d}\Gamma, \quad \text{with } \hat{q}^n=\hat{q}\ (t=n\Delta t). \tag{28b}$$

Having obtained the $\mathbf{u}^{n+1}$-velocity distribution from the solution of equations (9)–(11) and the $\mathbf{T}^{n+1}$-temperature distribution from equation (28a), the natural boundary condition for temperature can be calculated in the following way (via the consistent flux method,[13] in a way similar to the computation of $\partial p^n/\partial n$):

$$\hat{\mathbf{\Omega}}_q^{n+1}=\mathbf{M}_q(\mathbf{T}_q^{n+1}-\mathbf{T}_q^n)/\Delta t+[\mathbf{A}_q(\mathbf{u}^{n+1})+\alpha\mathbf{S}_q]\mathbf{T}_q^{n+1}, \tag{29}$$

where matrices $\mathbf{M}_q$, $\mathbf{A}_q^{n+1}$ and $\mathbf{S}_q$ are formed in the usual way (at element level) considering only those elements which are adjacent to Γ_4. vectors $\mathbf{T}_q^n$ and $\mathbf{T}_q^{n+1}$ contain the temperature values of those boundary nodes which belong to the elements mentioned above. At these nodes the natural BC (via equation (25d)) is needed for $t=(n+2)\Delta t$.

The introduction of the balancing tensor diffusivity (BTD) for equation (25b) is also necessary. Thus the equation to solve is

$$\partial T/\partial t+(\mathbf{u}\cdot\nabla)T=\nabla\cdot\boldsymbol{\alpha}^*\nabla T, \tag{30a}$$

with the elementwise-calculated BTD

$$\boldsymbol{\alpha}^*=\begin{bmatrix}\alpha+\bar{u}^n\bar{u}^n(\Delta t/2) & \bar{u}^n\bar{v}^n(\Delta t/2)\\ \bar{u}^n\bar{v}^n(\Delta t/2) & \alpha+\bar{v}^n\bar{v}^n(\Delta t/2)\end{bmatrix}. \tag{30b}$$

Figure 5 shows the temperature distribution for the example detailed in the previous section (see Figure 1 for the boundary conditions). The steady state temperature field is reached after about 4000 time steps ($\alpha=0{\cdot}02$).

In the next section three more examples are presented for different Reynolds numbers (Re) to demonstrate the performance of the developed scheme.

NUMERICAL EXAMPLES

Lid-driven cavity flow

In the previous sections the lid-driven cavity flow problem (in the present case a flow-through cavity) has been presented for $Re=100$ and $Pr=0{\cdot}5$ ($Pr=\nu/\alpha$). Now, using the same coarse but

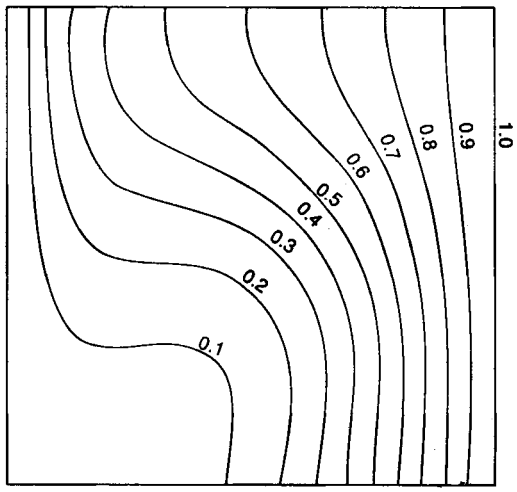

Figure 5. Temperature distribution; $Re=100$, $Pr=0{\cdot}5$

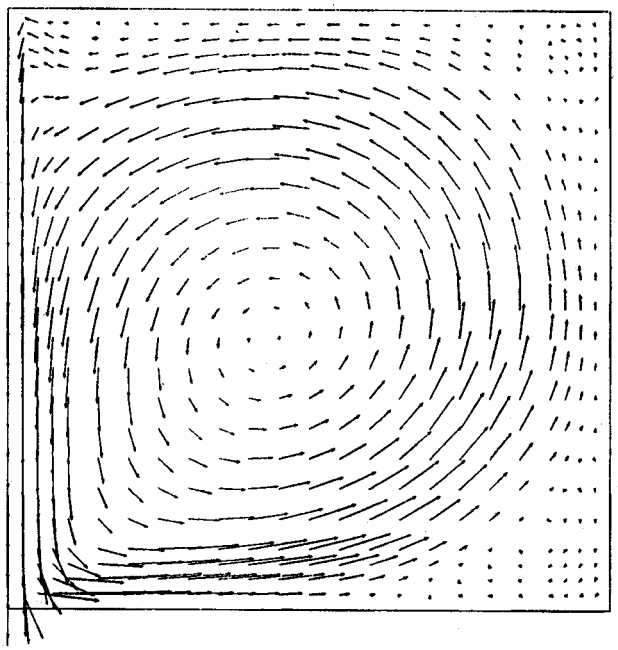

Figure 6(a). Velocity distribution; $Re=1000$

graded mesh of 625 nodes, the lid-driven cavity flow problem is solved for $Re=1000$. Even though the mesh is very coarse, the change of the flow pattern for the different Re can be compared to the work of Ghia *et al.*[18] The BCs are the same as those in Figure 1. Other parameters of the computations are $\mu=1{\cdot}0$, $\rho=1000$, $\mathbf{g}=0$ and $\Delta t=0{\cdot}01$. The steady state has been obtained at about the 4000th time step ($t=40$ s). The pressure distribution is shown in Figure 6(b). Although

this mesh is obviously not fine enough to correctly simulate the flow and to make numerical comparison, it can be seen that the primary vortex has moved towards the geometric centre of the cavity and that the secondary (corner) eddies are more developed for $Re = 1000$ (Figure 6(a)) than for $Re = 100$ (Figure 4(b)).

Figure 7 shows the pressure field for $Re = 100$ when an acceleration ($g_x = 1{\cdot}0$) is included in the calculation. It is clear that the velocity distribution has not changed for this case.

To obtain correct results for high Re, it is necessary to use a much finer mesh.

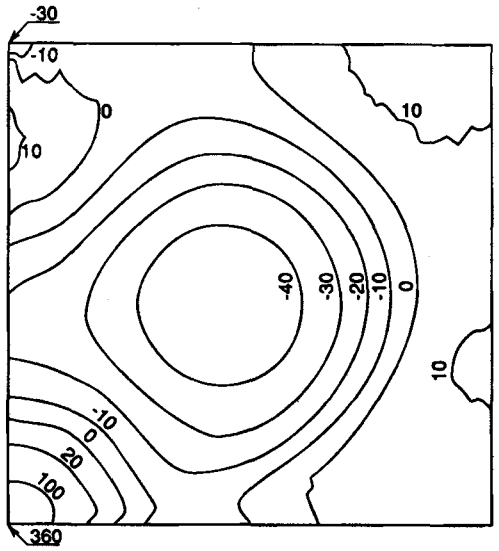

Figure 6(b). Pressure distribution; $Re = 1000$

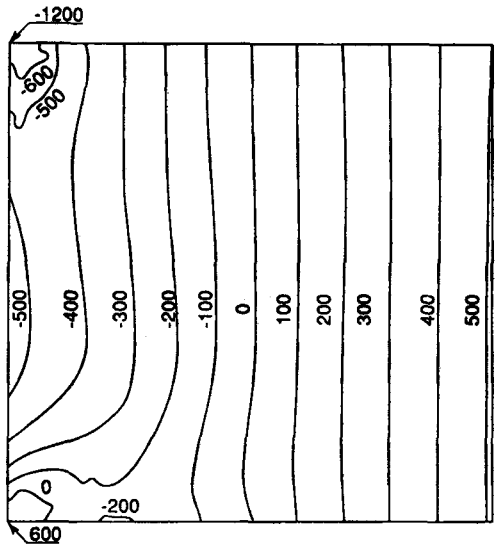

Figure 7. Pressure distribution for the lid-driven cavity flow; $Re = 100$, $g_x = 1$

'Doubled' lid-driven cavity flow with hotplate

To examine the symmetry of the calculation, the computational domain of the previous example is doubled and the right-hand-side as well as the left-hand-side walls are moving with uniform ($v = -1{\cdot}0$) velocity. The parameters of the calculation are the same as those of the 'single' lid-driven cavity for $Re = 100$, and the hotplate is placed in the middle of the upper horizontal wall. The velocity, pressure and temperature distributions for $Re = 100$ are presented in Figure 8. It can be seen that the results are symmetrical. In comparison to the 'single' lid-driven cavity flow, the primary vortices have moved towards the centre of the domain and there are no secondary vortices.

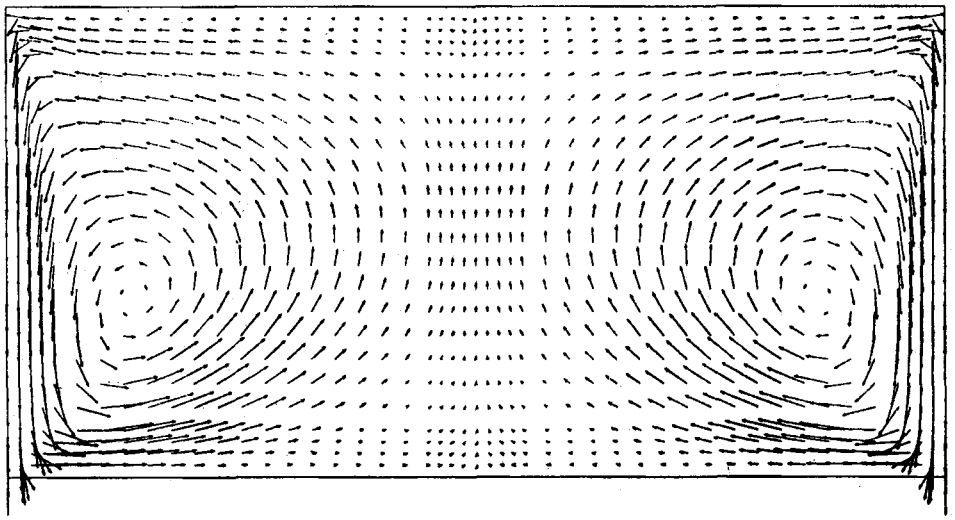

Figure 8(a). Velocity distribution; $Re = 100$

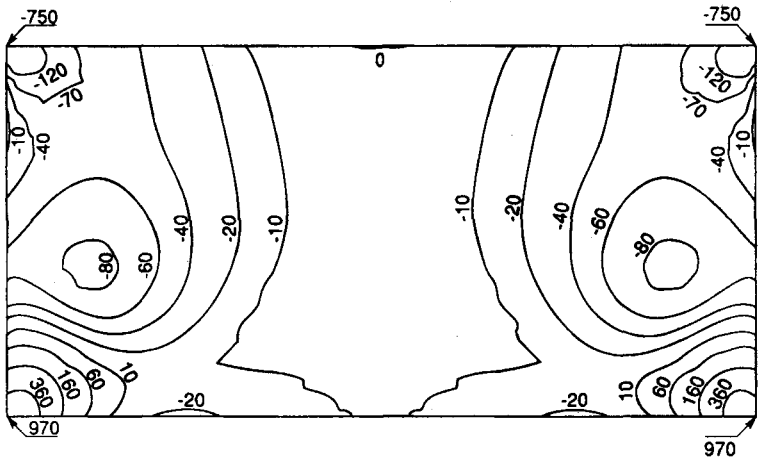

Figure 8(b). Pressure distribution; $Re = 100$

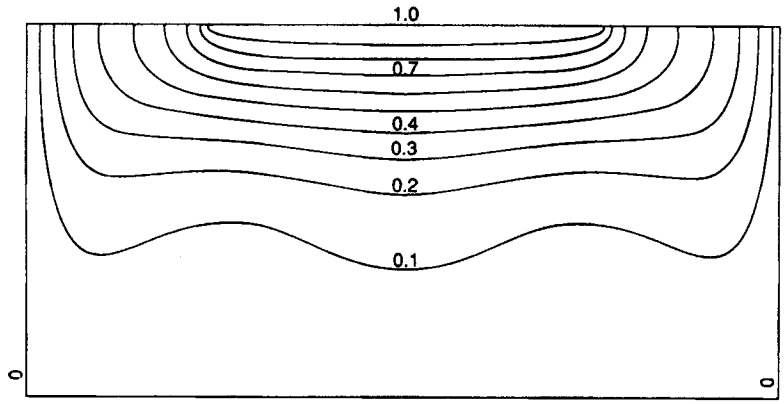

Figure 8(c). Temperature distribution; $Re=100$, $Pr=0{\cdot}5$

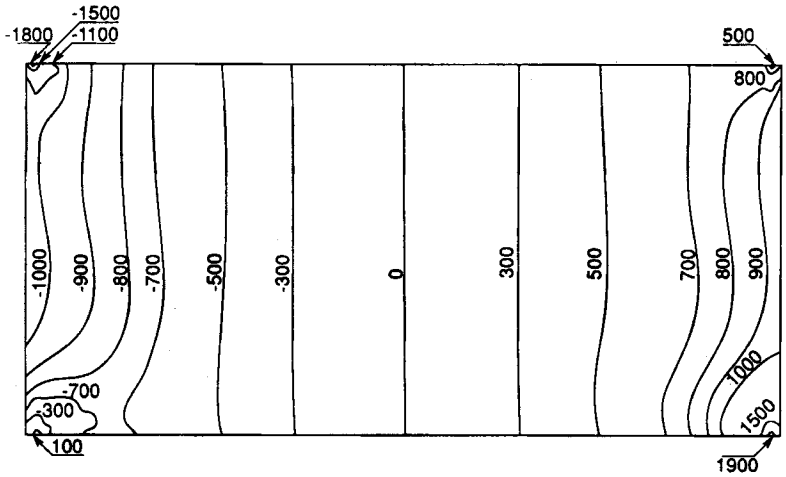

Figure 9. Pressure distribution for 'doubled' lid-driven cavity flow with hotplate; $Re=100$, $g_x=1$

Figure 9 shows the results for the same Re but with an acceleration ($g_x=1{\cdot}0$) in the x-direction. The velocity and temperature distributions for the case with acceleration do not show any noticeable change; thus in Figure 9 only the pressure distribution is presented.

In Figure 10 the velocity, pressure and temperature distributions can be seen when only the left-hand-side wall is moving and $\mu=2{\cdot}0$; thus $Re=500$ and $Pr=0{\cdot}1$.

Flow through sudden expansion

The geometry of the flow region together with the boundary and initial conditions are illustrated in Figure 11. At the inlet fully developed Poiseuille flow is assumed; this can only be approximated by the four-node elements. The example of flow through a sudden expansion for $Re=60$ was presented in References 4 and 7 as well as in Reference 14. Here the results are given for $Re=10$ and 60. The flow region has been discretized by a graded mesh of 1148 elements and 1231 nodes. Other parameters of the computation are—for $Re=10$: $\mu=6{\cdot}0$, $\rho=1{\cdot}0$ $\alpha=1{\cdot}0$,

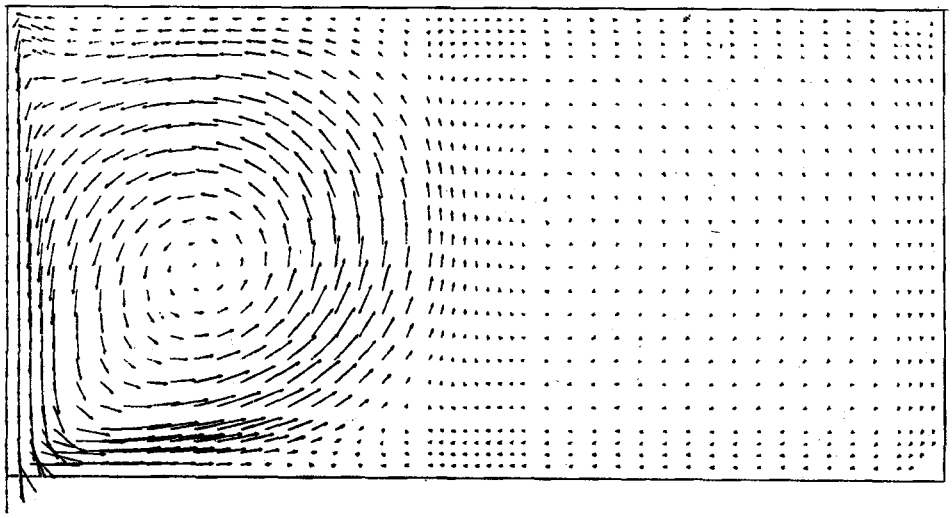

Figure 10(a). Velocity distribution; $Re = 500$

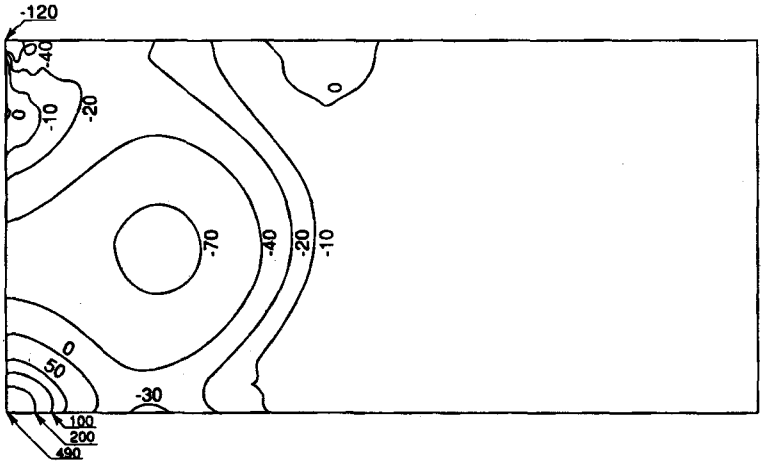

Figure 10(b). Pressure distribution; $Re = 500$

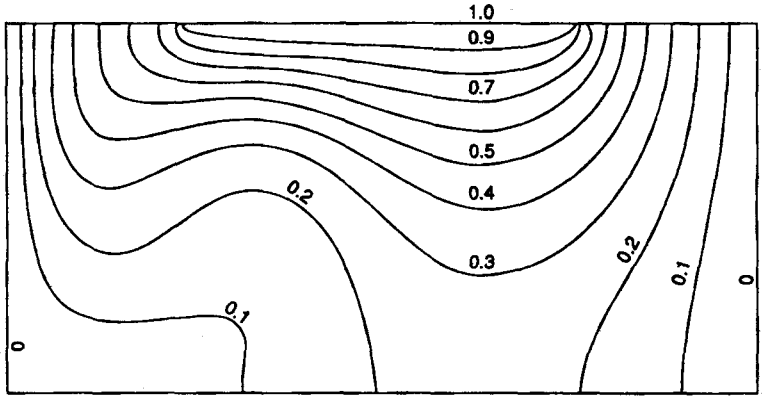

Figure 10(c). Temperature distribution; $Re = 500$, $Pr = 0{\cdot}1$

$Pr=6{\cdot}0$; for $Re=60$: $\mu=1{\cdot}0$, $\rho=1{\cdot}0$, $\alpha=1{\cdot}0$, $Pr=1{\cdot}0$. With respect to the outlet boundary conditions the computation for $Re=10$ is carried out in two ways.

Case A (same as Case A earlier). At the outlet, similarly to the inlet, fully developed Poiseuille flow is prescribed. At $x=2$, $y=0$ the reference pressure $\hat{p}$ is calculated via equation (24).

Case B (same as Case B earlier). At the outlet the natural boundary condition for velocity is specified through equations (13b) and (13c), while the Dirichlet boundary condition for pressure is given by equations (17) and (18).

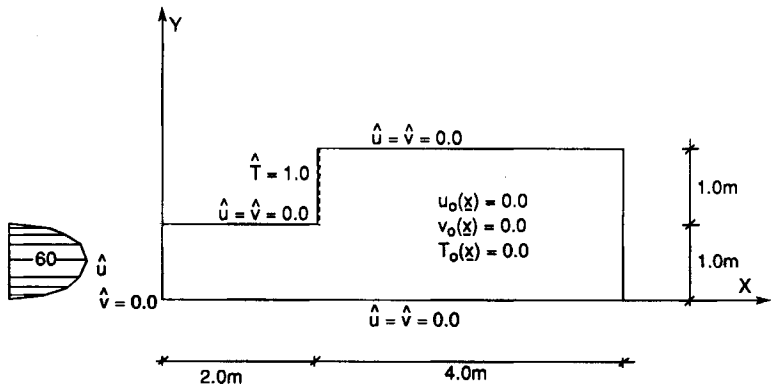

Figure 11. Flow through sudden expansion; initial and boundary conditions

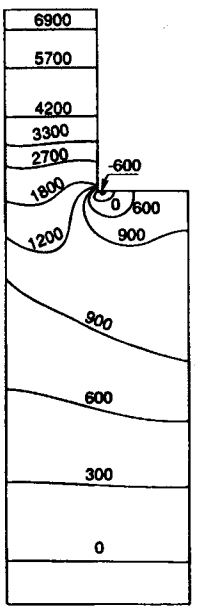

Figure 12(a). Pressure distribution with Dirichlet BC at the outlet; $Re=10$

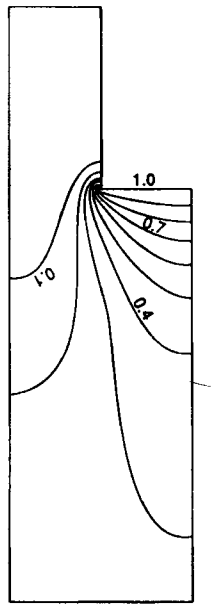

Figure 12(b). Temperature distribution with Dirichlet BC at the outlet; $Re=10$, $Pr=6$

Figures 12 and 13 show the results for Case A and B respectively. The velocity distribution for Case A is not presented because it is virtually indistinguishable from that of Case B. The pressure and temperature fields, however, do demonstrate that the different boundary conditions lead to different solutions.

For $Re=60$ (since the computational domain is obviously truncated) the boundary conditions are the same as those of Case B for $Re=10$. The velocity, pressure and temperature distributions are presented in Figure 14. In Figure 14(b) the dashed lines show the pressure distribution (only where they are distinguishable) obtained in Reference 7. It can be seen that there is no qualitative difference between the two solutions although they differ slightly near the outlet (see $p=0$ and $p=50$ lines). The pressure calculated in Reference 14 resembles both solutions displayed in Figure 14(b). In the calculation of Reference 14 the lines corresponding to $p<-100$ are similar to those of the present method, while the line corresponding to $p=0$ is like the one presented in Reference 7. For the most part in can be concluded that these solutions agree well.

For both $Re=10$ and 60 the time increment is chosen to be $\Delta t=0{\cdot}0002$; to reach the steady state solution, about 4000 time steps are necessary.

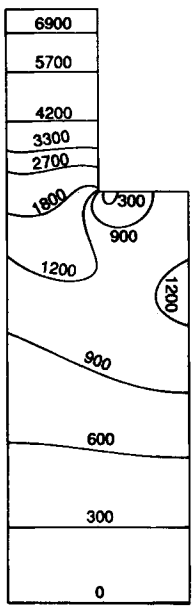

Figure 13(a). Pressure distribution with natural BC at the outlet; $Re=10$

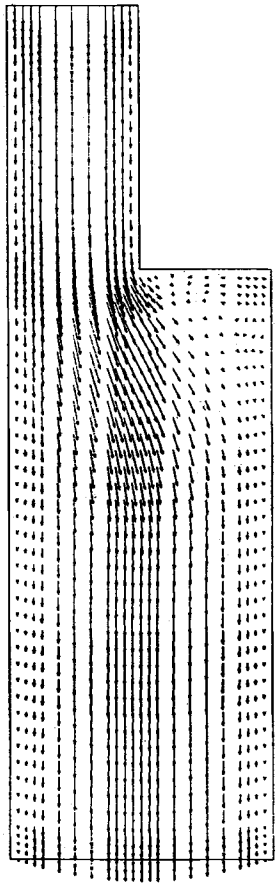

Figure 13(b). Velocity distribution with natural BC at the outlet; $Re=10$

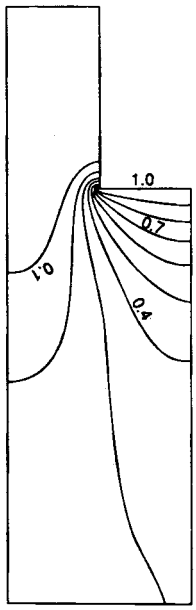

Figure 13(c). Temperature distribution with natural BC at the outlet; $Re=10$, $Pr=6$

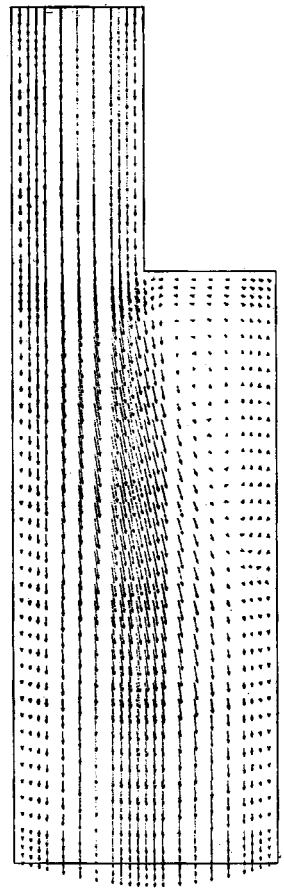

Figure 14(a). Velocity distribution; $Re=60$

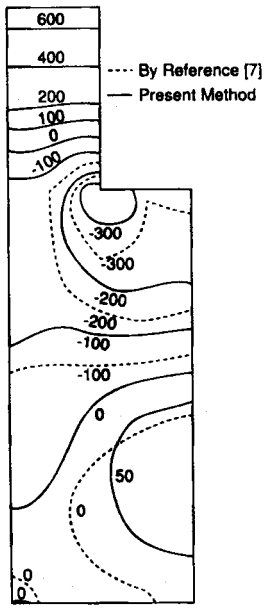

Figure 14(b). Pressure distribution; $Re=60$

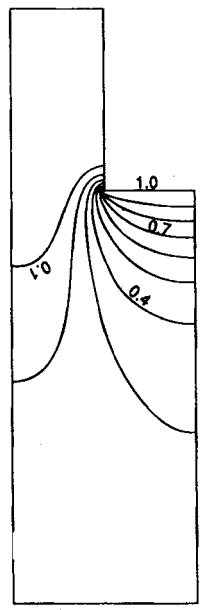

Figure 14(c). Temperature distribution; $Re=60$

CONCLUSIONS

A finite element solution has been presented to solve the incompressible Navier–Stokes equations for various Reynolds numbers using an explicit time integration scheme and the Galerkin weighted residual method for discretization in space.

To solve the directly derived pressure Poisson equation, in the second step of the algorithm the consistent Neumann boundary condition for pressure is introduced. When the normal velocity is given on the entire boundary to avoid the discontinuity of the Neumann boundary condition for pressure, the Dirichlet boundary condition for pressure is prescribed in a flexible way.

Owing to the application of the directly derived pressure Poisson equation, there is no spurious pressure mode. For the calculation of the tangential and normal derivatives of pressure the consistent flux method is utilized.

The balancing tensor viscosity/diffusivity is applied to make the explicit time integration scheme stable (for advection-dominated flow) and to increase the time increment.

Employing bilinear interpolation functions for the approximation of the primitive variables, two-point quadrature is used to evaluate the Galerkin integrals and the mass matrix is lumped via row-sum at element level.

To obtain velocity, pressure and temperature distributions in the first example, which utilizes 625 nodes, the computation of 3000 time steps required 16 min CPU time on a FACOM-VP30 computer. In order to make this technique more cost-effective, future effort includes an effective code vectorization and the introduction of a subcycling process for pressure.

ACKNOWLEDGEMENTS

The computation of the examples has been carried out on the FACOM-VP30 of the Chou University. A. K. is very grateful to the students of the Computational Hydraulic Laboratory, Chuo University for their help. The support received from the National Science Foundation (Grant CTS-8901598) was vital for the revision of the paper.

REFERENCES

1. P. Hood and C. Taylor, 'Navier–Stokes equations using mixed interpolations', in *Finite Element Methods in Flow Problems*, pp. 57–66, edited by J. T. Oden, O. C. Zienkiewicz, R. H. Gallagher and C. Taylor, UAH Press, Huntsville, 1974.
2. C. Taylor and P. Hood, 'A numerical solution of the Navier–Stokes equations using the finite element method', *J. Comput. Fluids*, **1**, 73–100 (1973).
3. M. Kawahara, N. Yoshimura, K. Nakagawa and H. Oshaka, 'Steady and unsteady finite element analysis of the incompressible viscous fluid', *Int. j. numer. methods eng.*, **10**, 437–456 (1976).
4. P. S. Huyakorn, C. Taylor, R. L. Lee and P. M. Gresho, 'A comparison of various mixed-interpolation finite elements in the velocity–pressure formulation of the Navier–Stokes equations', *J. Comput. Fluids*, **6**, 25–35 (1978).
5. R. L. Sani, P. M. Gresho, R. L. Lee and D. F. Griffiths, 'The cause and cure (?) of the spurious pressures generated by certain FEM solutions of the incompressible Navier–Stokes equations: Part 1', *Int. j. numer. methods fluids*, **1**, 17–43 (1981).
6. P. M. Gresho, S. T. Chen, R. L. Lee and C. D. Upson, 'A modified finite element method for solving the time dependent, incompressible Navier–Stokes equations. Part 1: Theory', *Int. j. numer. methods fluids*, **4**, 557–598 (1984).
7. R. L. Lee, P. M. Gresho and R. L. Sani, 'Smoothing techniques for certain primitive variable solutions of the Navier–Stokes equations', *Int. j. numer. methods eng.*, **14**, 1785–1804 (1979).
8. S. Tuann and M. D. Olson, 'A transient finite element solution method for the Navier–Stokes equations', *J. Comput. Fluids*, **6**, 141–152 (1978).
9. A Mizukami and M. Tsuchiya, 'A finite element model for the three-dimensional non-steady Navier–Stokes equations', *Int. j. numer. methods fluids*, **4**, 349–357 (1984).
10. S. Tuann and M. D. Olson, 'Numerical studies of the flow around a circular cylinders by a finite element method', *J. Comput. Fluids*, **6**, 219–240 (1978).

11. P. M. Gresho and R. L. Sani, 'On the pressure boundary conditions for the incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **7**, 1111–1145 (1987).
12. A. J. Chorin, 'Numerical solution of the Navier–Stokes equations', *J. Math. Comput.*, **XXII**, 101–104 (1968).
13. P. M. Gresho, R. L. Lee, R. L. Sani, M. K. Maslanik and B. E. Eaton, 'The consistent Galerkin FEM for computing derived boundary quantities in thermal and/or fluid problems', *Int. j. numer. methods fluids*, **7**, 371–394 (1987).
14. M. Shimura and M. Kawahara, 'Finite element flow analysis using the velocity correction method', in *Proc. Int. Conf. on Numerical Methods in Engineering: Theory and Applications*, Swansea, 6–10 July 1987, NUMETA '87: Volume 2, edited by G. N. Pande and J. Middleton, Dordrecht Boston, Lancaster, Matinus Nijhoff Publishers, 1987, pp. T32, 1–8.
15. B. Ramaswamy and M. Kawahara, 'Lagrangian finite element analysis applied to viscous free surface fluid flow', *Int. j. numer. methods fluids*, **7**, 953–984 (1987).
16. O. C. Zienkiewicz, *The Finite Element Method*, 3rd edn, McGraw-Hill, London, 1977.
17. M. Shimura and M. Kawahara, 'Two-dimensional finite element flow analysis using the velocity correction method', *Proc. JSCE*, Tokyo, Japan: Japan Society of Civil Engineers, 1987.
18. U. Ghia, K. N. Ghia and C. T. Shin, 'High-*Re* solutions for the incompressible flow using the Navier–Stokes equations and a multigrid method', *J. Comput. Phys.*, **48**, 387–411 (1982).