INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 17, 667–685 (1993)

PRECONDITIONED FINITE ELEMENT ALGORITHMS FOR 3D STOKES FLOWS

RICHARD Q. N. ZHOU*

Department of Chemical Engineering and Applied Chemistry, University of Toronto, Toronto, Ontario, Canada M5S 1A4

SUMMARY

Preconditioned conjugate gradient algorithms for solving 3D Stokes problems by stable piecewise discontinuous pressure finite elements are presented. The emphasis is on the preconditioning schemes and their numerical implementation for use with Hermitian based discontinuous pressure elements. For the piecewise constant discontinuous pressure elements, a variant implementation of the preconditioner proposed by Cahouet and Chabard for the continuous pressure elements is employed. For the piecewise linear discontinuous pressure elements, a new preconditioner is presented. Numerical examples are presented for the cubic lid-driven cavity problem with two representative elements, i.e. the Q2-P0 and the Q2-P1 brick elements. Numerical results show that the preconditioning schemes are very effective in reducing the number of pressure iterations at very reasonable costs. It is also shown that they are insensitive to the mesh Reynolds number except for nearly steady flows ($Re_m \to 0$) and are almost independent of mesh sizes. It is demonstrated that the schemes perform reasonably well on non-uniform meshes.

KEY WORDS 3D Stokes flow Finite element method Uzawa algorithm Preconditioned conjugate gradient method

1. INTRODUCTION

Many numerical methods for solving incompressible Navier–Stokes equations, such as semi-implicit methods and explicit methods, include a Stokes problem solver. In semi-implicit methods, the non-linear convection term and the incompressibility condition are decoupled. That reduces the problem to several subproblems, including a Stokes problem. If the Reynolds number is small, an explicit method can be used, and the problem is reduced to solving a sequence of Stokes problems. Therefore, solving Stokes problems efficiently, especially in three-dimensional space, is crucial to the overall efficiency of these methods.

Many finite-element algorithms have been used to solve the Stokes problems. For three-dimensional (3D) and large-scale 2D problems, the direct methods, e.g. penalty methods, are inefficient and are limited to relatively small mesh[1] due to large CPU time and storage requirements. In these cases, iterative methods are most attractive because they require relatively small storage and can run more efficiently, especially on vector and parallel computers. It has been shown that the preconditioned Uzawa-type conjugate gradient methods are effective in solving Stokes problems. Most of the preconditioning schemes are based on incomplete factorization of the system matrix or its approximation which needs to be explicitly constructed. This approach is general, but the process can be very expensive. Although a proper preconditioner can substantially accelerate the convergence, it is essential to minimize its cost and complexity,

* Present address: Atomic Energy Canada Limited, Chalk River Laboratories, Chalk River, Ontario, Canada K0J 1J0.

0271–2091/93/200667–19$14.50

Received 26 October 1992
Revised 2 March 1993

especially for three-dimensional problems, so that the gain in efficiency achieved by reducing the number of iterations is not offset by the preconditioning costs.

Recently, based on Fourier analysis, Cahouet and Chabard[2] proposed a continuous preconditioning operator. Its unique advantage is that the preconditioning is done by solving two continuous equations. Consequently, the expensive construction and incomplete factorization of the system matrix or its approximation is avoided. Furthermore, these continuous equations can be solved on a mesh much coarser than that on which the momentum equation is based. Hence, solving the preconditioning equations requires less storage and CPU time than solving momentum equation. They also showed that the convergence rate was almost independent of the mesh sizes for a given problem.

However, numerical results of the preconditioning idea have been reported only with respect to the elements where the pressure is continuous across the element boundaries.[2-4] There is no information on the effectiveness of the scheme when discontinuous pressure elements are used. It is conceivable that the implementation of the preconditioner would be different if piecewise discontinuous pressure elements were used, where the pressure space does not contain the solution of the continuous preconditioning equation. Therefore, it is desirable to develop numerical algorithm specifically for use of discontinuous pressure elements.

Our goal is to develop a finite element scheme to simulate 3D cardiovascular flow problems, where large recirculation zones are expected. It has been shown[5,6] that continuous pressure elements often result in unsatisfactory flow patterns in the regions where there is a large amount of recirculation. It is generally thought that stable discontinuous pressure elements are superior in this case because they guarantee elementwise mass conservation, and thus yield better flow patterns.[7] Elementwise mass conservation must exclude continuous pressure elements. Therefore, it is of practical importance and interest to develop an efficient preconditioned conjugate gradient finite element scheme specifically for the stable discontinuous pressure elements.

The 2D Q2-P1 element with complete biquadratic velocity and linear discontinuous pressure is probably the best element known for incompressible flow simulation.[8] Its 3D counterpart, the 3D Q2-P1 element with complete triquadratic velocity and piecewise linear discontinuous pressure, has been shown to yield good results,[1] probably due to its second-order accuracy. On the other hand, more computational labour is required to reach convergence in comparison to piecewise constant discontinuous pressure elements. Generally, for a fixed convergence criterion more iterations are needed while using piecewise discontinuous linear pressure elements than while using piecewise discontinuous constant pressure elements, e.g. Q2-P0, since in addition to the elementwise mass conservation, the three moments of the elemental velocity divergence with respect to the centroid of an element must also vanish.

In this paper, we present preconditioned Uzawa-type conjugate gradient finite element algorithms for stable Hermitian based piecewise discontinuous pressure elements. We are primarily concerned with developing efficient preconditioning methods. The algorithms can be effectively applied to any stable hexahedral Hermitian piecewise discontinuous pressure elements where the pressure is either linear or constant, since the only difference between the elements with the same pressure interpolation is the number of velocity unknowns. To reduce coding labour, we choose the Q2-P0 element (triquadratic velocity and piecewise constant pressure) to represent the class of stable hexahedral piecewise constant discontinuous pressure elements, such as $Q_1^+–P_0$ and R_2^+-P_0,[8] and the Q2-P1 element to represent the class of stable hexahedral piecewise linear discontinuous pressure elements, e.g. Q_1^+-P_1.[9]

Section 2 describes the problem and the finite element formulation. In Section 3 the Uzawa-type conjugate gradient methods and the preconditioning procedure are outlined. Section 4 discusses the numerical results. Conclusions are given in Section 5.

2. FINITE ELEMENT FORMULATIONS

We consider the transient Stokes flow problem in a 3D domain Ω with boundary Γ:

$$\alpha\mathbf{u}-\nu\nabla^2\mathbf{u}+\nabla p=\mathbf{f},$$

$$\nabla\cdot\mathbf{u}=0, \tag{1}$$

$$\mathbf{u}|_\Gamma=\mathbf{g},$$

where $\mathbf{u}$ and p are, respectively, the velocity vector and pressure at the current time level; $\mathbf{f}$ is a known vector; the constant α depends on the time discretization scheme and ν is the reciprocal of the Reynolds number. Global mass conservation requires that

$$\int_\Gamma \mathbf{n}\cdot\mathbf{g}\,\mathrm{d}\Gamma=0, \tag{2}$$

where $\mathbf{n}$ is the outward normal vector on Γ.

Using a finite element triangulation $\mathcal{T}_h$ of Ω into hexahedrons and taking into account the velocity boundary conditions, the discrete form of formulation (2) is

$$\mathbf{AU}+\mathbf{B}^\mathrm{T}\mathbf{P}=\mathbf{F}. \tag{3}$$

$$\mathbf{BU}=\mathbf{H}, \tag{4}$$

where $\mathbf{U}$ and $\mathbf{P}$ are the vectors of the unknown nodal velocity and pressure, respectively, $\mathbf{F}$ and $\mathbf{H}$ are known vectors, $\mathbf{B}$ is the divergence matrix and

$$\mathbf{A}=\alpha\mathbf{M}+\nu\mathbf{S}, \tag{5}$$

where $\mathbf{M}$ is the consistent mass matrix and $\mathbf{S}$ is the diffusion matrix. Note that when the discontinuous piecewise linear pressure is used, $\mathbf{P}$ also contains the unknown pressure gradient values at the element centroids. The solution of equations (3) and (4) can be formally expressed as

$$\mathbf{BA}^{-1}\mathbf{B}^\mathrm{T}\mathbf{P}=\mathbf{BA}^{-1}\mathbf{F}-\mathbf{H}, \tag{6}$$

$$\mathbf{U}=\mathbf{A}^{-1}(\mathbf{F}-\mathbf{B}^\mathrm{T}\mathbf{P}). \tag{7}$$

Since the system matrix $\mathbf{L}=\mathbf{BA}^{-1}\mathbf{B}^\mathrm{T}$ is symmetric, positive-definite and is only implicitly defined, an iterative approach is most attractive.

The linear system of equations (3) and (4) can be solved using the Uzawa-type preconditioned conjugate gradient algorithm defined as follows:

```
Given initial guess $\mathbf{P}_0$ and convergence criterion $\varepsilon$;

$\mathbf{AU}_0=\mathbf{F}-\mathbf{B}^\mathrm{T}\mathbf{P}_0$,
$\mathbf{R}_0=\mathbf{BU}_0-\mathbf{H}$,
If $\|\mathbf{R}_0\|\leq\varepsilon$ stop;
$\mathbf{CG}_0=\mathbf{R}_0$;
$\mathbf{W}_0=\mathbf{G}_0$;
$\mathbf{AZ}_0=\mathbf{B}^\mathrm{T}\mathbf{W}_0$;
For $n\geq 0$:
$\rho_n=\mathbf{R}_n^\mathrm{T}\mathbf{G}_n/\mathbf{G}_n^\mathrm{T}\mathbf{BZ}_n$;
$\mathbf{P}_{n+1}=\mathbf{P}_n-\rho_n\mathbf{W}_n$;
```

```
$\mathbf{U}_{n+1}=\mathbf{U}_n-\rho_n\mathbf{Z}_n$;
$\mathbf{R}_{n+1}=\mathbf{R}_n-\rho_n\mathbf{B}\mathbf{Z}_n$;
If $\|\mathbf{R}_{n+1}\|\leq\varepsilon$ stop;
$\mathbf{C}\mathbf{G}_{n+1}=\mathbf{R}_{n+1}$;
$\lambda_n=\mathbf{R}^{\mathrm{T}}_{n+1}\mathbf{G}_{n+1}/\mathbf{R}^{\mathrm{T}}_n\mathbf{G}_n$;
$\mathbf{W}_{n+1}=\mathbf{G}_n+\lambda_n\mathbf{W}_n$;
$\mathbf{A}\mathbf{Z}_{n+1}=\mathbf{B}^{\mathrm{T}}\mathbf{W}_{n+1}$.                                  (8)
```

Note that in each iteration, the momentum equation $\mathbf{AZ}=\mathbf{B}^{\mathrm{T}}\mathbf{W}$ has to be solved for the velocity field or its associated direction vector $\mathbf{Z}$. It is clear that the efficiency of the algorithm is determined by the number of pressure iterations and the efficiency in solving the preconditioning equation $\mathbf{CG}=\mathbf{R}$ and the momentum equation. Since solving the momentum equation is generally expensive, a good preconditioning scheme must be effective in reducing the number of pressure iterations which in turn reduces the number of momentum equations to be solved.

Since the three velocity components in the momentum equation (7) are decoupled, they are solved separately. Thus, each velocity component is found by solving a smaller linear system. In this paper, the momentum equation for each velocity component is solved by the preconditioned conjugate gradient method, where the preconditioning scheme is the incomplete Cholesky factorization of $\mathbf{A}$ (ICCG). A compact storage scheme is used to store $\mathbf{A}$, in which only the non-zero entries are stored. The details can be found in Reference 10. In general, the convergence criterion ε_{m} for solving equation (7) depends on the pressure iteration convergence criterion ε. Unless otherwise stated, we set $\varepsilon=10^{-6}$ and $\varepsilon_{\mathrm{m}}=10^{-8}$. We also compare the efficiency of the ICCG method with that of the conjugate gradient with diagonal scaling preconditioning for solving the momentum equation (DSIC) (see Table VII).

3. PRECONDITIONERS

Recently, Cahouet and Chabard[2] proposed a continuous preconditioning operator

$$\mathbf{C}_1^{-1}=\nu\mathbf{I}^{-1}-\alpha\nabla^{-2}, \tag{9}$$

where $\mathbf{I}$ is the identity operator and ∇^{-2} is the inverse Laplacian operator. Taking the velocity divergence to be a simple harmonic function, they showed that the gradient method with $\mathbf{C}_1$ as the preconditioning operator is independent of the wavelength. Hence, the gradient method works equally well over the entire spectrum. This indicates that (see Appendix for proof)

$$\mathbf{C}_1=-\nabla\cdot(\alpha\mathbf{I}-\nu\nabla^2)^{-1}\nabla. \tag{10}$$

The right-hand side term of equation (10) is exactly the continuum system operator corresponding to the discrete system matrix $\mathbf{L}$. Equation (10) means that $\mathbf{C}_1$ is the best continuum preconditioner in the sense that in the continuum domain, the preconditioned gradient method gives the exact solution in one pressure iteration regardless of the initial pressure guess.

It should be pointed out that equation (10) does not guarantee the equality of their discrete matrices. A discrete form of $\mathbf{C}_1$ is not equal to the discrete system matrix $\mathbf{L}$. However, the idea is that a preconditioning matrix $\mathbf{C}$ should be spectrally close to $\mathbf{L}$ such that $\mathbf{C}^{-1}\mathbf{L}\approx\mathbf{I}$. Since $\mathbf{C}_1$ is identical to the continuum system operator, we hope that its discrete matrix would not be *too different from* $\mathbf{L}$. The other crucial criterion for a preconditioner is that it must be easy to implement and not too expensive to solve. Equation (9) suggests that the preconditioning can be decomposed into solving an algebraic equation and a Laplace equation, both of which are easy and inexpensive to solve.

In practice, the solution of the continuum preconditioning equation is decomposed as follows:

$$g=\mathbf{C}_1^{-1}\nabla\cdot\mathbf{u}=\nu\mathbf{I}^{-1}\nabla\cdot\mathbf{u}-\alpha\nabla^{-2}\nabla\cdot\mathbf{u}=g_s+g_u, \tag{11}$$

where the subscripts s and u symbolize the steady and unsteady parts. It is clear that g_s and g_u are governed by

$$\mathbf{I}g_s=\nu\nabla\cdot\mathbf{u}. \tag{12}$$

$$\nabla^2 g_u=-\alpha\nabla\cdot\mathbf{u}, \qquad \left.\frac{\partial g_u}{\partial n}\right|_\Gamma=0, \tag{13}$$

where the homogeneous Neumann boundary condition is imposed on g_u. Note that g_s does not need boundary condition. It is well known that the solution of equation (13) is arbitrary up to an additive constant. To fix this constant, we require that

$$\int_\Omega g_u\,\mathrm{d}\Omega=0. \tag{14}$$

The boundary condition for g_u is the difference of that for g and the boundary values of g_s. The preconditioned gradient pressure iteration is given by

$$p^{n+1}=p^n-\lambda_n g^n. \tag{15}$$

If a pressure boundary condition which p^n must satisfy is given, g must satisfy the corresponding homogeneous boundary condition. Unfortunately, for most flow problems, the pressure boundary condition is often undefined. We are left with no boundary condition for g, hence for g_u. Using the argument of boundary layer, Cahouet and Chabard[2] recommended the homogeneous. Neumann condition for the pressure

$$\left.\frac{\partial p}{\partial n}\right|_\Gamma=0. \tag{16}$$

Equation (16) can be quite inaccurate for certain flow problems, such as the outflow boundary condition of a duct flow, where it is clearly violated. The lack of boundary condition for pressure makes the choice of that for g and g_u somewhat arbitrary. A general rule is that g must vanish with $\nabla\cdot\mathbf{u}$. Since g_s satisfies this rule, so must g_u. The homogeneous Neumann condition, combined with equation (14), guarantees that $g_u\to0$ as $\nabla\cdot\mathbf{u}\to0$, while it imposes *least* constraints on g_u. It is also the easiest for finite volume implementation.

While a weak solution g_s of equation (12) belongs to $L_2(\Omega)$, a weak solution g_u of equation (13) must belong to $H_1(\Omega)$. The preconditioning operator $\mathbf{C}_1$ has been applied to the continuous pressure elements, i.e. P2–P1 element[2] and P1–P1 iso P2 element,[3,4] and a great improvement in convergence was achieved. For the discontinuous pressure elements, the pressure space does not belong to $H_1(\Omega)$. Hence, a solution g_u in the discontinuous pressure space does not exist. Therefore, a weak solution g does not exist.

3.1. Preconditioning scheme for Q2-P0 element

To extend $\mathbf{C}_1$ to Q2–P0 element, we assume that the preconditioned discrete gradient direction $\mathbf{G}$ in algorithm (8) is given by

$$\mathbf{G}=\mathbf{G}_s+\mathbf{G}_u, \tag{17}$$

where $\mathbf{G}$, $\mathbf{G}_s$ and $\mathbf{G}_u$ are vectors of dimension n_{el} which is the number of elements. We further assume that $g_u \in H_1(\Omega)$, and $g_s = G_s^k$ on the kth element e_h^k where G_s^k is the kth component of $\mathbf{G}_s$. It is obvious from equation (12) that G_s^k is given by

$$\int_{e_h^k} G_s^k \, de_h^k = V_k G_s^k = \nu \int_{e_h^k} \nabla \cdot \mathbf{u} \, de_h^k = \nu R_k, \quad k = 1, \ldots, n_{el}, \tag{18}$$

where V_k is the volume of e_h^k. Solving equation (18) takes only one vector operation and hence, needs virtually no computational labour.

Let g_u^k be the value of g_u at the centroid of e_h^k; equation (13) can be solved for g_u^k using the finite-volume method[11]

$$\int_{e_h^k} \nabla^2 g_u \, de_h^k = \int_{\partial e_h^k} \frac{\partial g_u}{\partial n} \, d\partial e_h^k = -\alpha \int_{e_h^k} \nabla \cdot \mathbf{u} \, de_h^k = -\alpha R_k, \quad \left. \frac{\partial g_u}{\partial n} \right|_{\partial e^k \cap \Gamma} = 0, \quad k = 1, \ldots, n_{el}, \tag{19}$$

where ∂e_h^k is the boundary of e_h^k and $\mathbf{n}$ the outward normal vector of ∂e_h^k.

Applying the finite-volume method to equation (19) results in a seven-point computational stencil. The coefficient matrix is symmetric and semipositive-definite due to the homogeneous Neumann boundary conditions. The resulting algebraic equation is solved using the preconditioned conjugate gradient method where the preconditioning matrix is the tridiagonal part of the coefficient matrix. Although the algebraic system matrix of equation (19) is semidefinite and hence is singular, we found that it converges faster without specifying the arbitrary constant in the solution of equation (19). Once a converged solution of equation (19) is found, we subtract from it a constant to satisfy equation (14). The kth component of $\mathbf{G}_u$ is then set by

$$G_u^k = g_u^k. \tag{20}$$

The algorithm is particularly suitable for vectorization.

Note that the right-hand side terms of equations (18) and (19) are exactly the velocity divergence residual vector $\mathbf{R} = \mathbf{BU} - \mathbf{H}$ in equation (8) and are available from the pressure iteration. This preconditioning scheme is therefore very efficient.

3.2. Preconditioning scheme for Q2-P1 element

For the Q2-P1 element, the pressure space is the set of piecewise discontinuous linear polynomials. As with Q2-P0 element, the preconditioned gradient vector $\mathbf{G}$ is split into two parts as given by equation (17). Now $\mathbf{G}$, $\mathbf{G}_s$ and $\mathbf{G}_u$ are vectors of dimension $4\mathbf{n}_{el}$ because the unknown pressure $\mathbf{P}$ also contains the pressure gradients at the element centroids.

Our numerical experiments suggest that equation (12) can be extended to the Q2-P1 element. On the element e_h^k, we assume the solution g_s takes the form

$$g_s = \sum_{i=1}^{4} G_{si}^k \psi_i^k, \tag{21}$$

where $\psi_i^k = x_i - x_{ic}^k$, $i = 1, 2, 3$, $\psi_4^k = 1$, x_{ic}^k is the centroidal co-ordinate of e_h^k, and G_{si}^k is a component of $\mathbf{G}_s$. It is clear that G_{s4}^k can be interpreted as the centroidal value of g_s and G_{si}^k, $i = 1, 2, 3$ is the centroidal values of $\partial g_s / \partial x_i$. Substituting equation (21) into equation (12) yields

$$G_{s4}^k = \nu R_4^k / V_k, \quad k = 1, \ldots, n_{el},$$

$$\sum_{j=1}^{3} G_{sj}^k \int_{e_h^k} \psi_i^k \psi_j^k \, de_h^k = \nu R_i^k - G_{s4}^k \int_{e_h^k} \psi_i^k \, de_h^k, \quad i = 1, \ldots, 3; \; k = 1, \ldots, n_{el}, \tag{22}$$

where the weighted velocity divergence residual

$$R_i^k = \int_{e_h^k} \psi_i^k \nabla \cdot \mathbf{u} \, de_h^k, \; i=1, \ldots, 4; \; k=1, \ldots, n_{\mathrm{el}} \tag{23}$$

are components of the residual vector **R**, hence, do not need to be recalculated. Equation (22) involves n_{el} 3×3 symmetric positive-definite linear systems and the computational cost is negligible with respect to solving the momentum equation (see Table V). The algorithm (22) can easily be vectorized.

Unfortunately, our numerical experiments appear to indicate that equation (13) cannot be used for the Q2-P1 element due to the presence of the pressure gradient at the element centroids. The numerical results suggest that the centroidal gradients $\partial g_u / \partial x_i$ are not the proper components of $\mathbf{G}_u$. Therefore, a new preconditioner for calculating $\mathbf{G}_u$ has to be found.

Since $\mathbf{BM}^{-1}\mathbf{B}^{\mathrm{T}}$ is the discrete form of the Laplacian, we define the unsteady preconditioned gradient direction $\mathbf{G}_u$ by

$$\mathbf{BM}_l^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{G}_u = \alpha(\mathbf{BU}-\mathbf{H}) = \alpha\mathbf{R}. \tag{24}$$

where $\mathbf{M}_l$ is the lumped mass matrix. The preconditioning operator defined by equations (22) and (24) can be written as

$$\tilde{\mathbf{C}}_1^{-1} = \nu\mathbf{I}^{-1} + \alpha(\mathbf{BM}_l^{-1}\mathbf{B}^{\mathrm{T}})^{-1}. \tag{25}$$

Note that for $\tilde{\mathbf{C}}_1$, the steady part is implemented by solving a continuous equation, while the unsteady part is implemented discretely.

Equation (24) is solved by the ICCG method where the coefficient matrix and its incomplete Cholesky factorization are constructed and stored. Note that the matrix **B**, however, is not globally formed to save storage. $\mathbf{BM}_l^{-1}\mathbf{B}^{\mathrm{T}}$ is constructed directly from elemental contribution. Again, only the non-zero entries are stored. The matrix $\mathbf{BM}_l^{-1}\mathbf{B}^{\mathrm{T}}$ has, compared to **A**, slightly fewer non-zero entries and lower sparsity. Its construction time is about the same as that of **A** and its incomplete factorization time is slightly more than half of that for **A**. We set the convergence criterion for solving equation (24) to be equal to ε. We found that more iterations are required for equation (24) than for the momentum equation (7). However, the numerical results show that it does not take more time to solve equation (24) than equation (7) (see Table VI). Therefore, solving equation (24) requires less CPU time and storage than solving equation (7).

We have also tested $\mathbf{C}_2$ and $\mathbf{C}_3$ defined by

$$\mathbf{C}_2^{-1} = \nu\mathbf{I}^{-1} + \alpha(\mathbf{B}(\operatorname{diag}\mathbf{M})^{-1}\mathbf{B}^{\mathrm{T}})^{-1} \quad \text{and} \quad \mathbf{C}_3^{-1} = (\mathbf{B}(\operatorname{diag}\mathbf{A})^{-1}\mathbf{B}^{\mathrm{T}})^{-1}. \tag{26}$$

These cases have been examined for P1-P1 iso P2 element by Carriere and Jeandel,[4] who reported that $\mathbf{C}_3$ gave the best convergence rate. However, we found that for the Q2-P1 element, $\mathbf{C}_3$ does not converge for the steady flow using the refined non-uniform mesh. For the coarse uniform mesh, $\mathbf{C}_3$ converges with a significantly slower rate in comparison to $\tilde{\mathbf{C}}_1$. As to $\mathbf{C}_2$, we found it to be slower than $\tilde{\mathbf{C}}_1$. The performance of $\mathbf{C}_2$ deteriorates as α increases, indicating that $\mathbf{C}_2$ is not suitable for transient flow analysis with small time step. Both $\mathbf{C}_2$ and $\mathbf{C}_3$ require the same computational costs as $\tilde{\mathbf{C}}_1$.

Another preconditioning scheme for the Q2-P1 element is the block scaling where the preconditioning is done by equation (22) only. The advantage of this idea is that it is very cheap. Our experience with the block scaling were that it does give a convergent solution, although for some cases we have run, a slightly smaller ε_m was necessary. Generally, the pressure iteration convergence is slow and deteriorates quickly with α as expected. For the unsteady flow ($Re_m = 10$) with the intermediate uniform mesh (see Table I), CG scheme with the block scaling needs 43

Table I. The characteristics of uniform and non-uniform meshes

Mesh	Number of elements	Number of free velocity nodes	Number of pressure nodes Q2-P0	Number of pressure nodes Q2-P1	Number of DOF Q2-P0	Number of DOF Q2-P1
Coarse $5\times5\times5$	125	729	125	500	2312	2687
Intermediate $10\times10\times10$	1000	6859	1000	4000	21 577	24 577
Refined $15\times15\times15$	3375	24 389	3375	13 500	76 542	86 667

iterations to converge while CG with $\tilde{\mathbf{C}}_1$ requires only 8 iterations. The total CPU time for the block scaling CG is 493 s. The total time on iteration for CG with $\tilde{\mathbf{C}}_1$ is 175 s, and the time on construction and incomplete Cholesky decomposition of $\mathbf{BM}_l^{-1}\mathbf{B}^T$ is 89 s. Hence, the block scaling CG method requires about twice CPU time as does the present scheme. The difference widens with the mesh non-uniformity and decreasing mesh size. Therefore, the block scaling is less effective than is $\tilde{\mathbf{C}}_1$ in reducing the pressure iteration number, and the overall cost of the block scaling CG is far more expensive than the present scheme, especially for large mesh. For steady flow, $\tilde{\mathbf{C}}_1$ reduces to the block scaling.

The effectiveness of the present preconditioners, i.e. equations (18) and (19) for the Q2-P0 element and (25) for the Q2-P1 element, is demonstrated by applying the preconditioned conjugate gradient algorithm to the 3D lid-driven cavity flow in an unit cube with the top face moving at the unit speed. The numerical integrations were performed by the $3\times3\times3$ Gaussian quadrature. All computations were performed using double precision and were run on the Apollo DN10000 at The University of Michigan and the DEC5000 workstation at Chalk River Laboratories of Atomic Energy of Canada Limited.

4. NUMERICAL RESULTS AND DISCUSSIONS

We define the average convergence rate as

$$\chi=\left(\frac{\|\mathbf{R}_n\|}{\|\mathbf{R}_0\|}\right)^{1/n}=\left(\frac{\|\mathbf{BU}_n-\mathbf{H}\|}{\|\mathbf{BU}_0-\mathbf{H}\|}\right)^{1/n}, \tag{27}$$

where n is the smallest number of pressure iterations which reduces the norm of the initial velocity divergence residual $\|\mathbf{R}_0\|$ to less than $\varepsilon\|\mathbf{R}_0\|$, where ε is the convergence criterion. In practice, ε depends on individual problems and the accuracy one wants to obtain. In this paper, we choose $\varepsilon=10^{-6}$. In the following figures, the norm of the velocity divergence residual $\|\mathbf{R}\|$ is normalized with respect to the initial value $\|\mathbf{R}_0\|$.

We construct three uniform and three non-uniform meshes consisting $5\times5\times5$ (coarse), $10\times10\times10$ (intermediate) and $15\times15\times15$ (refined) elements. The details of the mesh specification are given in Table I. The numbers of the velocity nodes are equal to those of the three uniform meshes used by Cahouet and Chabard[2] for the P2-P1 elements, so that a comparison can be made.

The elemental corner nodal points of the non-uniform meshes are defined by

$$x_i=1-\cos\frac{i-1}{N}\frac{\pi}{2},\quad i=1,2,\ldots,N+1,$$

$$y_j=\sin\frac{j-1}{N}\frac{\pi}{2},\quad j=1,2,\ldots,N+1,$$

$$z_k=\sqrt{\left(\frac{k-1}{N}\right)},\quad k=1,2,\ldots,N+1,\tag{28}$$

where $N=5$, 10, 15. For $N=10$ (intermediate mesh), the ratio of the largest to the smallest scale is equal to 26. This results in a system with worse condition number (than that of a uniform mesh, where this ratio is unity). Since this ratio increases with N, the system condition number also increases with N. Therefore, they are valuable test cases to illustrate how the scheme performs for stiff systems.

The efficiency of the preconditioned scheme depends on the mesh Reynolds number given by

$$Re_{\mathrm{m}}=\frac{\alpha h^2}{\nu},\tag{29}$$

where $h=1/N$ is the characteristic length of the mesh. Obviously, this length is the average value when the non-uniform meshes are used. The mesh Reynolds number characterizes the influence of the sizes of the time and spatial discretization. For each mesh, two mesh Reynolds numbers are used. The case $Re_{\mathrm{m}}=0$ corresponds to steady flow, and the case $Re_{\mathrm{m}}=10$ corresponds to unsteady flow. These two values characterize the typical efficiency of the scheme. For a given uniform mesh, χ approaches a constant as $Re_{\mathrm{m}}\rightarrow\infty$ as shown in Figure 4.

Figures 1(a) and 1(b) show the convergence of the normalized velocity divergence residual of the preconditioned scheme using uniform Q2-P0 elements. The results of the non-preconditioned scheme are also shown for comparison. It can be seen that the residual decays faster for the unsteady cases than for the steady cases. For the unsteady cases, the average rate of convergence χ defined by equation (27) is about the same for all three meshes. This indicates that for unsteady flow, the condition number of the preconditioned system of a uniform Q2-P0 element mesh is independent of the mesh size h. For the steady cases, Figure 1(a) suggests that χ aproaches a constant as the mesh is uniformly refined.

The effectiveness of the preconditioner is clearly demonstrated. Figure 1(a) shows that for the steady case, the residual of the non-preconditioned scheme is initially identical with that of the preconditioned scheme and then starts to deviate and oscillate. This oscillation is delayed by increasing the mesh size because the non-preconditioned system is better conditioned for a smaller mesh. It is caused by the errors in solving the momentum equation. Figure 3 shows that if ε_{m} is reduced from 10^{-8} to 10^{-12}, the oscillation occurs significantly later. If a direct method, such as the skyline method which has an equivalent convergence criterion of $\varepsilon_{\mathrm{m}}=10^{-15}$, is used to solve the momentum equation, the velocity divergence residual decreases monotonically to 10^{-11} before it starts to oscillate. However, a direct method has very large storage requirements and cannot be used for large meshes. Therefore, preconditioning is necessary not only for reducing the number of iterations, but also for obtaining a converged solution.

It is found that the slow convergence of the non-preconditioned scheme for the unsteady cases as shown in Figure 1(b) is not caused by ε_{m} provided that $\varepsilon_{\mathrm{m}}\leq\varepsilon/10$. This can be seen from the fact that the decay history of the non-preconditioned scheme deviates from that of the preconditioned ones right from the beginning.

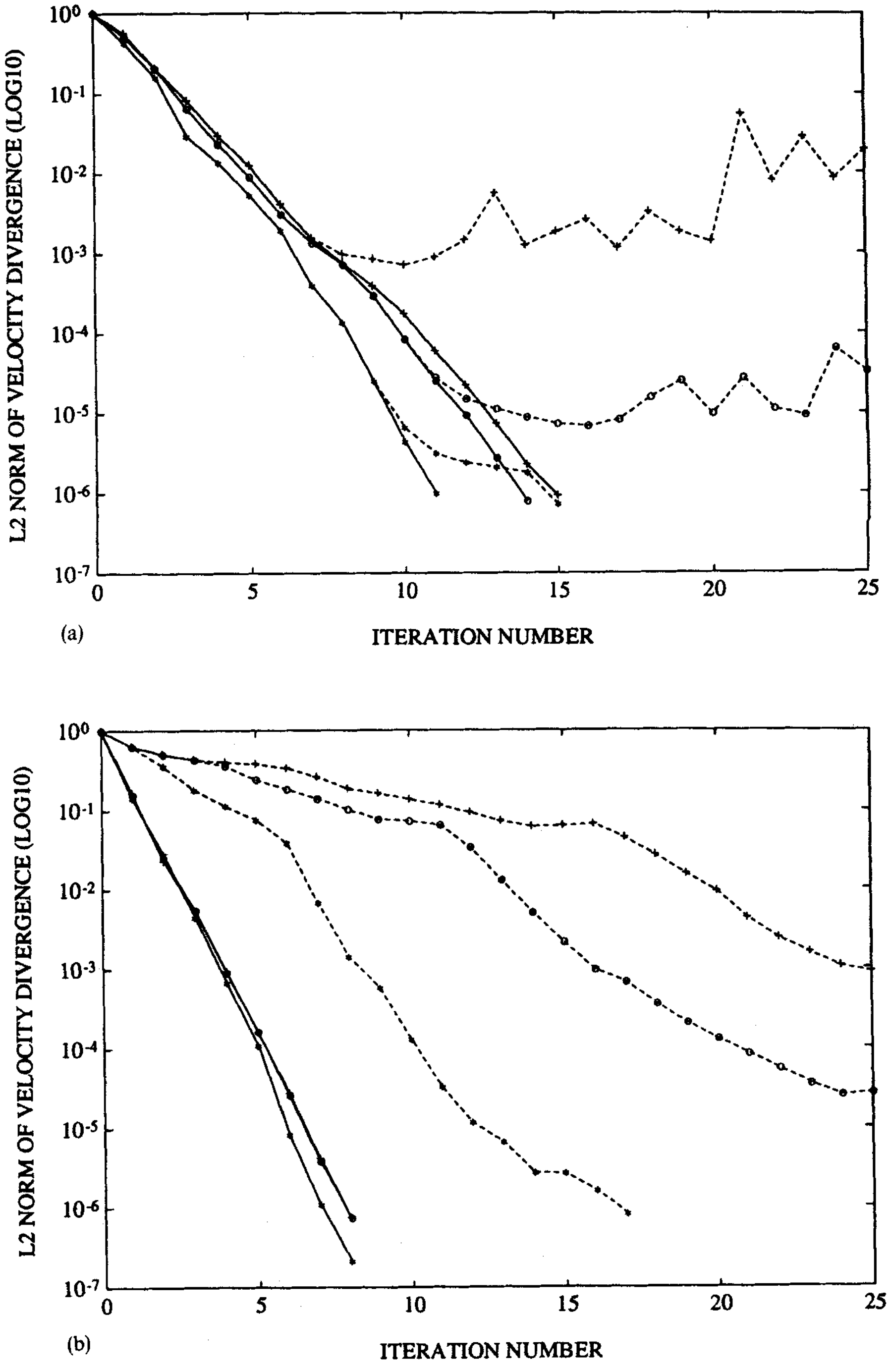


Figure 1. Convergence of velocity divergence residual for 3D lid-driven cavity flow using Q2-P0 element, uniform mesh: (a) steady flow, $Re_m=0$; (b) unsteady flow, $Re_m=10$. —*—, preconditioned CG scheme—coarse mesh; —○—, preconditioned CG scheme—intermediate mesh; —+—, preconditioned CG scheme—refined mesh; ---*---, non-preconditioned CG scheme—coarse mesh; ---○---, non-preconditioned CG scheme—intermediate mesh; ---+---, non-preconditioned CG scheme—refined mesh

Figures 2(a) and 2(b) show the decay history of the normalized velocity divergence residual using the Q2-P1 element. Again, convergence is faster for the unsteady cases. The convergence rate is insensitive to the mesh size for both steady and unsteady flows indicating that the preconditioned system condition number is independent of h. By comparison, the residuals of the non-preconditioned algorithm shown in Figures 2(a) and 2(b) are oscillatory and non-convergent because $\varepsilon_m = 10^{-8}$ is too large. Figure 3 shows that when $\varepsilon_m = 10^{-12}$, one obtains an oscillatory but convergent history of velocity divergence residual. The oscillation in the residual can be eliminated only through proper preconditioning. As for the Q2-P0 case, reducing ε_m does not help to accelerate the convergence.

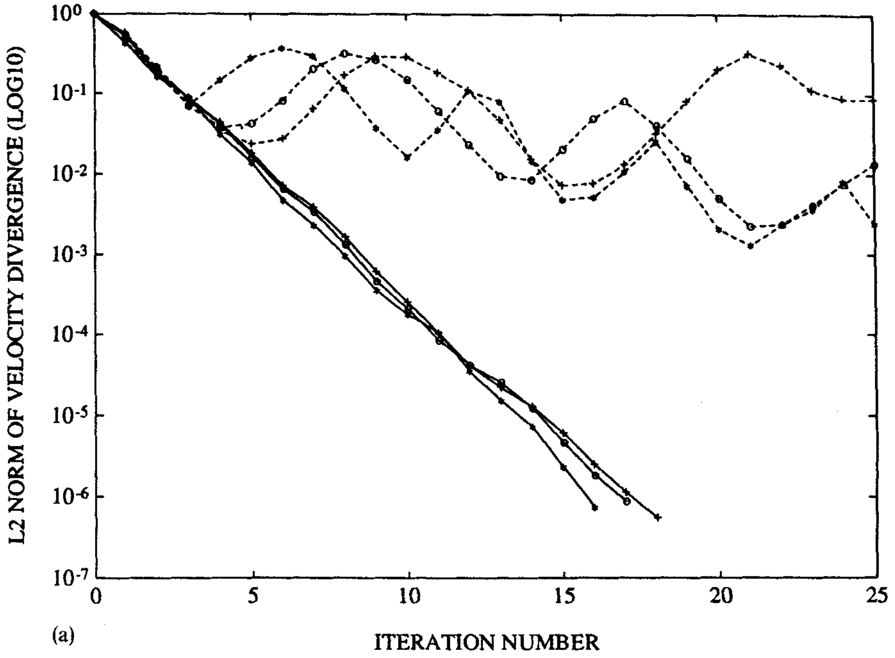
For the uniform meshes, more momentum equation iterations are required for the steady case than for the unsteady case, indicating that the matrix **A** of the momentum equation is better conditioned for the unsteady flow. This gives a better conditioned system matrix **L** for the unsteady case. Therefore, the preconditioned scheme converges faster for the unsteady flow than for the steady flow.

Figure 4 shows the dependency of the average convergence rate χ on the mesh Reynolds number Re_m with respect to the uniform intermediate mesh. For $Re_m \geq 5$, χ is almost constant for both elements, indicating that the preconditioned algorithms are insensitive to time step size except for very large times steps or for very small h, which effectively correspond to nearly steady flows. The slow convergence occurs for the steady cases where $Re_m = 0$.

Many flow problems require non-uniform meshes when the solutions vary rapidly from one region to another. A good numerical scheme must maintain its characteristics when non-uniform meshes are used. Figures 5(a) and 5(b) show the normalized velocity divergence residual versus iteration number for the non-uniform Q2-P0 meshes defined by equation (28). The preconditioned scheme still performs reasonably well, although, as expected, it converges considerably slower than it does with the uniform meshes. Contrary to the cases of the uniform meshes, the scheme converges slower for the unsteady cases than for the steady cases when the non-uniform meshes are used. This is probably due to the fact that the truncation error in solving equation (19) is of the order $O(h)$ with respect to the non-uniform meshes while it is of the order $O(h^2)$ with respect to the uniform mesh.

Figures 6(a) and 6(b) show the performance of the preconditioned scheme with respect to the non-uniform Q2-P1 element. One can see that the convergence strongly depends on the mesh size. The computational cost using non-uniform Q2-P1 elements is significantly higher in comparison with that of the uniform meshes because the number of iterations for solving the pressure equation, the momentum equation and the preconditioning equation (24), all, significantly increase.

The performance of the present preconditioned schemes is summarized in Tables II–VI where the results of the ten-node P2-P1 tetrahedral element used by Cahouet and Chabard[2] are also listed for comparison. Table II shows that for the unsteady cases, the convergence rates of the present preconditioned algorithm are about the same as that of the compatible discretization with the P2-P1 element. Table III shows that the present algorithms converge faster than that using the P2-P1 element for the steady cases, while the steady preconditioning costs virtually nothing (see Table V). The average convergence rates for the non-uniform meshes are listed in Table IV. Tables V and VI show the average CPU time per pressure iteration and per preconditioning using the DEC5000 workstation. It is shown that the costs of solving the preconditioning equations (18), (19) and (22) are indeed negligible. The slightly smaller CPU time per iteration with the Q2-P1 element than that with the Q2-P0 element, as shown in Table V for the intermediate mesh, is because Q2-P1 element requires more iterations and the last several iterations take far smaller CPU time than the average value. Thus, although the total iteration time using the Q2-P1



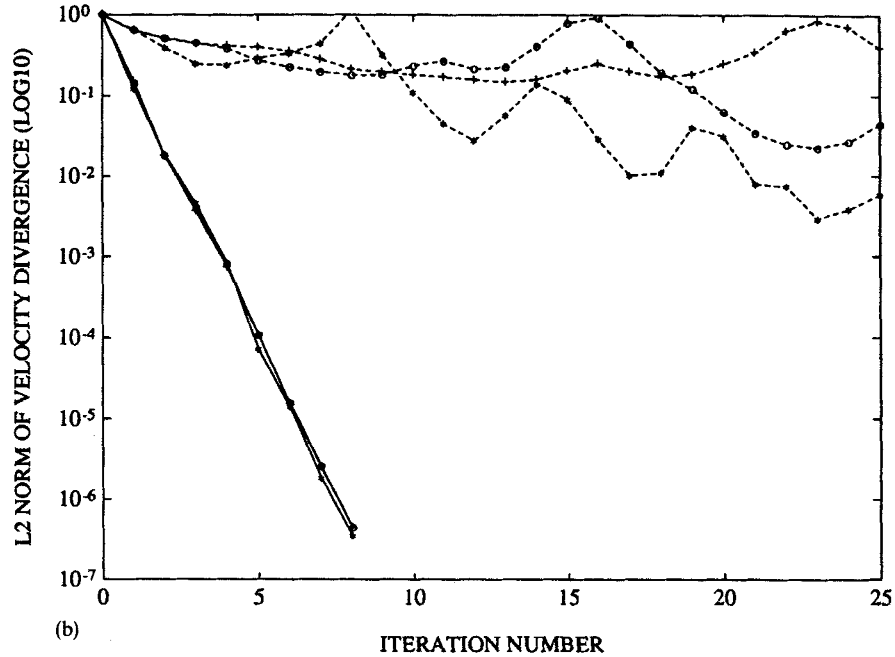


Figure 2. Convergence of velocity divergence residual for 3D lid-driven cavity flow with Q2–P1 element, uniform mesh: (a) steady flow, $Re_m = 0$; (b) unsteady flow, $Re_m = 10$. —∗—, preconditioned CG scheme—coarse mesh; —○—, preconditioned CG scheme—intermediate mesh; —+—, preconditioned CG scheme—refined mesh; - - -∗- - -, non-preconditioned CG scheme—coarse mesh: - - -○- - -, non-preconditioned CG scheme—intermediate mesh; - - -+- - -, non-preconditioned CG scheme—refined mesh

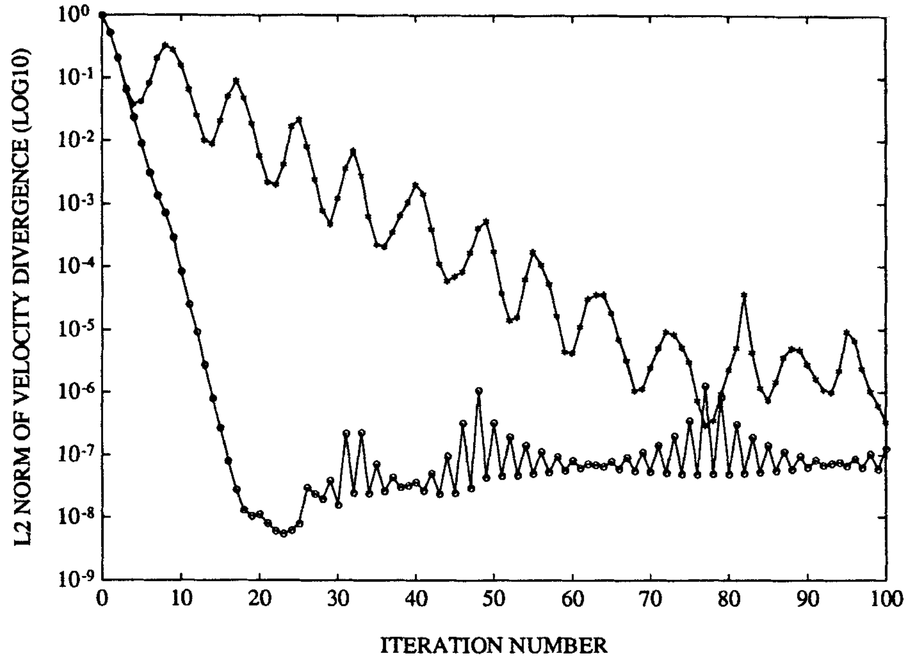


Figure 3. Convergence of velocity divergence residual for steady 3D lid-driven cavity flow using the non-preconditioned CG scheme and the uniform intermediate mesh: —∘—, Q2-P0 element; —∗—, Q2-P1 element. The convergence criterion for solving the momentum equation is $\varepsilon_m = 10^{-12}$

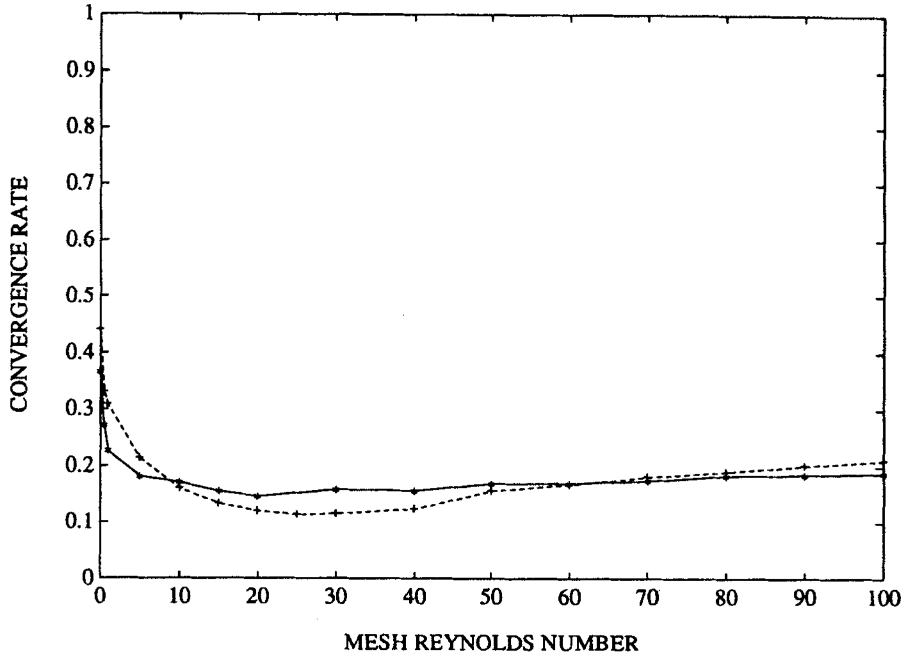


Figure 4. Influence of the mesh Reynolds number on the convergence for 3D lid-driven cavity flow with respect to the uniform intermediate mesh: —∗—, Q2-P0 element; ---+---, Q2-P1 element

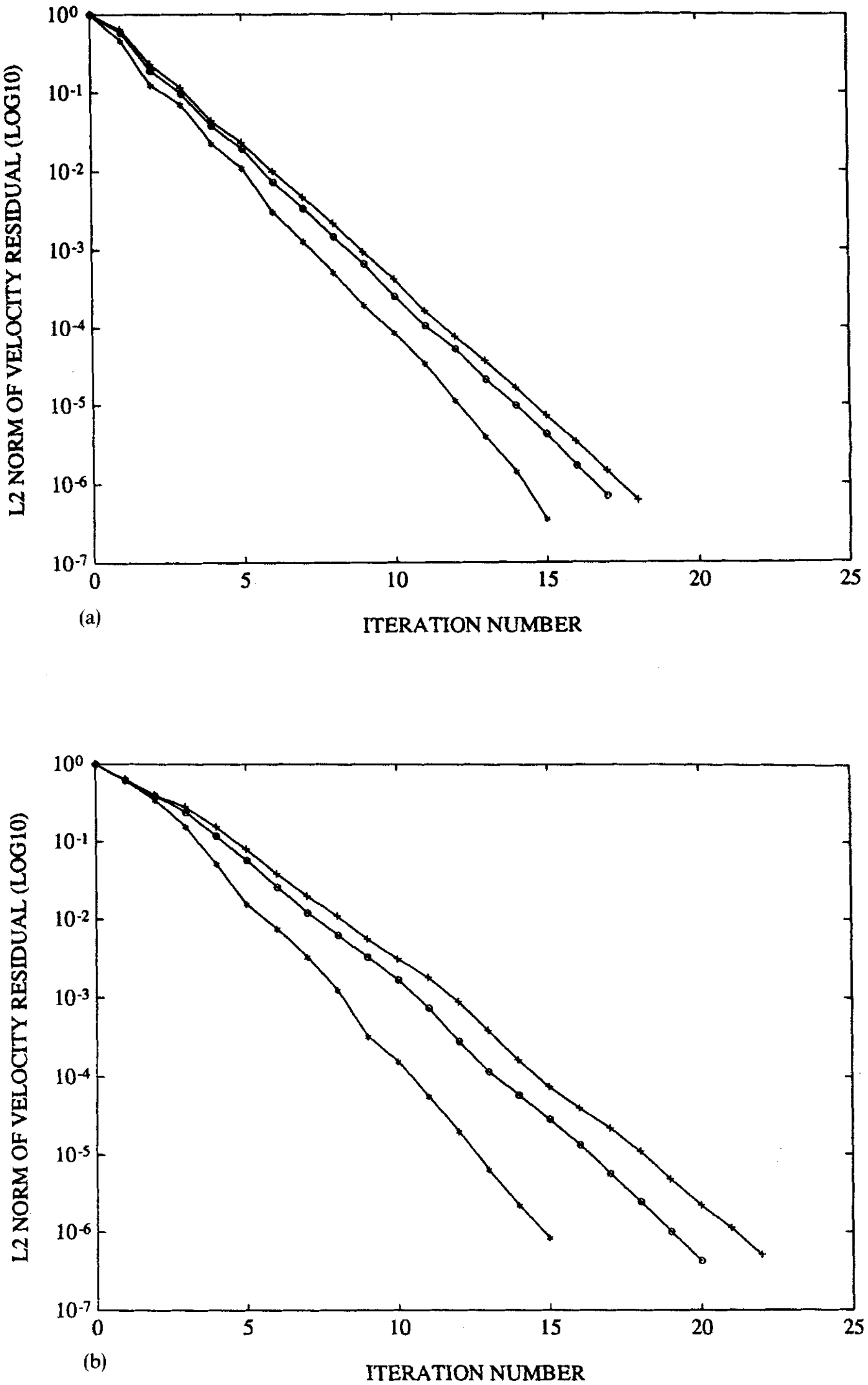


Figure 5. Convergence of velocity divergence residual for 3D lid-driven cavity flow with Q2-P0 element, non-uniform mesh: (a) steady flow, $Re_m = 0$; (b) unsteady flow, $Re_m = 10$. —*—, coarse mesh; —○—, intermediate mesh: —+—, refined mesh

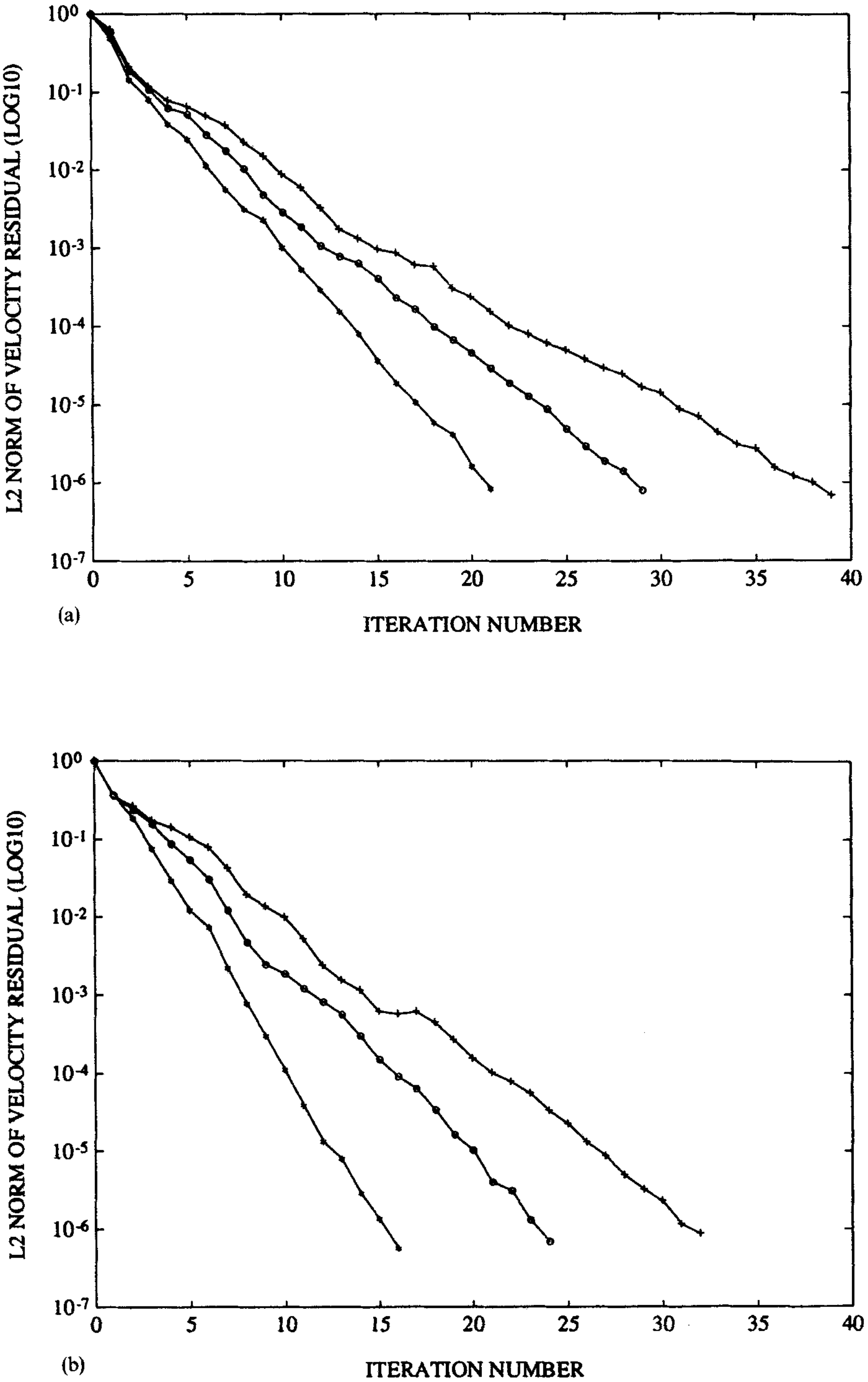


Figure 6. Convergence of velocity divergence residual for the steady 3D lid-driven cavity flow with Q2-P1 element, non-uniform mesh: (a) steady flow, $Re_m = 0$; (b) unsteady flow, $Re_m = 10$. coarse mesh; —○—, intermediate mesh; —+—, refined mesh

Table II. Convergence rates for the 3D unsteady lid-driven cavity flow ($Re_m = 10$, uniform mesh)

Mesh	Q2-P0	Q2-P1	P2-P1 (clas.)*	P2-P1 (comp.)*
Coarse	0·15	0·16	$0·3 \leq \chi \leq 0·49$	0·14
Intermediate	0·17	0·16	$0·3 \leq \chi \leq 0·5$	0·15
Refined	0·17	0·16	$0·3 \leq \chi \leq 0·5$	0·14

* The results of P2-P1 element are from Cahouet and Chabard.[2] clas. = direct discretization of (13), comp. = compatible discretization

Table III. Convergence rates for the 3D steady lid-driven cavity flow ($Re_m = 0$, uniform mesh)

Mesh	Q2-P0	Q2-P1	P2-P1*
Coarse	0·28	0·41	0·58
Intermediate	0·37	0·44	0·56
Refined	0·40	0·45	0·57

* The results of P2–P1 element are from Cahouet and Chabard[2]

Table IV. Convergence rates for the 3D lid-driven cavity flow using non-uniform meshes defined by (28). For the unsteady case, $Re_m = 10$, and for the steady case, $Re_m = 0$

Mesh	Q2-P0 Steady	Q2-P1 Steady	Q2-P0 Unsteady	Q2-P1 Unsteady
Coarse	0·37	0·51	0·39	0·41
Intermediate	0·43	0·62	0·48	0·55
Refined	0·45	0·69	0·51	0·65

Table V. Average CPU seconds per pressure iteration and per preconditioning for the steady flow where $Re_m = 0$

Mesh	Q2-P0 Precond.*	Q2-P0 Pressure†	Q2-P1 Precond.*	Q2-P1 Pressure†
Coarse	0·0004	1·58	0·001	1·60
Intermediate	0·002	28·70	0·005	28·3
Refined	0·007	146·38	0·036	150·21

* Precond. = CPU second per preconditioning
† Pressure = CPU second per pressure iteration

element is always greater than that using the Q2-P0 element, the average time per iteration could be actually smaller. Table VI shows that solving equation (24) requires slightly less CPU time than that required to solve the momentum equation (7).

For large 3D flow problems, full triquadratic velocity elements are generally expensive due to the large size of the momentum equation resulting from the large number of velocity nodes per element. The proposed preconditioned algorithms can be applied (with equal effectiveness, we believe) to other types of stable piecewise Hermitian discontinuous pressure elements (satisfying inf–sup condition) where there are fewer velocity nodes per element.

Applying the present preconditioned algorithm to the low order, unstable Q1-P0 elements[12] (trilinear velocity and piecewise constant pressure), shows that it converges only for the coarse mesh. For the intermediate and the refined meshes, the algorithm does not converge, as shown by Vincent and Boyer.[13] This indicates that the algorithm cannot be used with unstable elements which do not satisfy inf–sup condition. Recently, stabilized Q1-P0 elements to which the iterative methods can be applied,[14] have been developed. However, because of the presence of the stabilizing operator, the concept of splitting the preconditioner into the steady and unsteady part is not applicable, and a preconditioner such as that proposed by Vincent and Boyer[13] must be adopted.

Finally, we compare the efficiency of ICCG method with that of DSCG method for solving the momentum equation (7). The storage requirement of ICCG method is always greater than that of DSCG by the storage requirement of **A** since the incomplete Cholesky factorization of **A** is stored. Table VII summarizes their performance with respect to the Q2-P0 element. The total CPU

Table VI. Average CPU seconds per pressure iteration and per preconditioning for the unsteady flow where $Re_m = 10$

Mesh	Q2-P0 Precond.*	Q2-P0 Pressure†	Q2-P1 Precond.*	Q2-P1 Pressure†
Coarse	0·01	0·83	0·52	1·47
Intermediate	0·24	11·08	9·83	21·86
Refined	1·69	43·68	45·29	101·55

* Precond. = CPU second per preconditioning
† Pressure = CPU second per pressure iteration

Table VII. Comparison of CPU time of ICCG and DSCG methods for solving the momentum equation

Mesh	ICCG method			DSCG method		
	NOI/CPU		ICF set up	NOI/CPU		DS set up
	Steady	Unsteady		Steady	Unsteady	
Uniform meshes						
Coarse	11/1·58	8/0·83	1·36	11/1·74	8/0·91	0·004
Intermediate	14/28·70	8/11·08	94·87	14/39·08	8/12·78	0·02
Refined	15/146·38	8/43·68	1045·00	15/215·65	8/50·13	0·05
Non-uniform meshes						
Coarse	15/1·65	15/1·08	1·36	15/3·0	15/1·83	0·004
Intermediate	17/32·43	20/14·03	94·87	17/73·17	20/29·43	0·02
Refined	18/171·80	22/61·12	1044·10	18/412·49	22/135·24	0·05

ICF = Incomplete Cholesky factorization
DS = Diagonal scaling
NOI = Number of iterations
CPU = Average CPU seconds per iteration

seconds spent on the pressure iteration is the product of the number of pressure iterations and the CPU seconds per iteration. The overall CPU seconds is obtained by adding the set-up time. It is seen that the CPU time per iteration using ICCG is always smaller than that using DSCG. If one has a transient problem and the mesh does not change with time, ICCG method should be used unless the storage is a serious concern. For a large non-uniform mesh, say larger than the intermediate mesh, ICCG should also be used since in this case, the efficiency gain by reducing the pressure iteration number exceeds the incomplete factorization set-up cost.

5. CONCLUSIONS

Preconditioned Uzawa-type conjugate gradient finite element algorithms for solving 3D Stokes flow problems with respect to stable hexahedral discontinuous pressure elements have been presented and tested. The preconditioning scheme, which was developed[2] for use with elements of continuous pressure profile, has been extended and modified to work with elements of piecewise constant discontinuous pressure interpolation. A new preconditioning scheme has been proposed for the elements of piecewise linear discontinuous pressure interpolation.

It has been shown that the preconditioning schemes are very effective in reducing the number of pressure iterations, and the costs are very low for use with the Q2-P0 element and moderate for use with the Q2-P1 element. When a piecewise constant pressure profile is used, the preconditioning equations can be solved very efficiently by the finite volume method, and the solution converges faster than when a continuous pressure profile is used. For the discontinuous linear pressure elements, the proposed new preconditioning scheme has been verified by the test problem to be very effective, and not too costly.

A major obstacle to applying the preconditioning idea of Cahouet and Chabard[2] to the Q2-P1 element is the presence of the elemental centroidal pressure gradient in the numerical discretization. It would be interesting to examine its applicability to a Lagrangian based discontinuous pressure element where there are more than one pressure degree of freedom per element. For this type of elements, the pressure gradients do not occur.

ACKNOWLEDGEMENTS

The author is gratefully indebted to Professor W. Schultz for providing the computer facility, to Mr. J. Li for his help, which was essential to this work. He wishes to thank Drs. R. Ethier and S. Karpik for discussions and partial financial support. He is also grateful to the refrees for their comments and suggestions which greatly improved this work.

APPENDIX

To prove relation (10), it is sufficient to show that for any trial function g satisfying

$$\int_{-\infty}^{\infty} |g(\mathbf{x})|\,\mathrm{d}\mathbf{x} < \infty \quad \text{and} \quad \lim_{|\mathbf{x}|\to\infty} |g| = 0, \tag{30}$$

the following equation holds

$$\mathbf{C}_1^{-1}(-\nabla\cdot(\alpha\mathbf{I}-\nu\nabla^2)^{-1}\nabla)g = g. \tag{31}$$

Denoting

$$F(g) = \int_{-\infty}^{\infty} g(\mathbf{x})e^{\mathrm{j}\mathbf{k}\cdot\mathbf{x}}\,\mathrm{d}\mathbf{x} \quad \text{and} \quad \mathbf{U}_z = -\nabla\cdot(\alpha\mathbf{I}-\nu\nabla^2)^{-1}\nabla \tag{32}$$

where $j=\sqrt{(-1)}$, $\mathbf{x}=(x_1, x_2, x_3)$, $\mathbf{k}=(k_1, k_2, k_3)$, and $d\mathbf{x}=dx_1\,dx_2\,dx_3$, we have (see Reference 15),

$$F(\nabla^2 g)=-k^2F(g), \qquad F(\nabla^{-2}g)=-\frac{F(g)}{k^2}, \qquad F(\nabla\cdot\boldsymbol{\phi})=j\mathbf{k}\cdot F(\boldsymbol{\phi}), \qquad F(\nabla g)=j\mathbf{k}F(g),$$

$$F((\nu\mathbf{I}^{-1}-\alpha\nabla^{-2})g)=\frac{\alpha+\nu k^2}{k^2}F(g) \quad\text{and}\quad F((\alpha\mathbf{I}-\nu\nabla^2)^{-1}\boldsymbol{\phi})=\frac{F(\boldsymbol{\phi})}{\alpha+\nu k^2}, \tag{33}$$

where $k^2=\mathbf{k}\cdot\mathbf{k}=k_1^2+k_2^2+k_2^2$.

Applying the Fourier transform to the left-hand side of equation (31) and repeatedly making use of equation (33) yields

$$F(\mathbf{C}_1^{-1}\mathbf{U}_z g)=\frac{\alpha+\nu k^2}{k^2}F(\mathbf{U}_z g)=F(g). \tag{34}$$

REFERENCES

1. C. Rindt, A. van Steenhoven, J. Janssen and G. Vossers, 'Unsteady entrance flow in a 90° curved tube', *J. Fluid. Mech.*, **226,** 445–474 (1991).
2. J. Cahouet and J.-P. Chabard, 'Some fast 3D finite element solvers for the generalized Stokes problem', *Int. j. numer. methods fluids*, **8**, 869–895 (1988).
3. M. Buffat, 'Simulation of two- and three-dimensional internal subsonic flows using a finite element method', *Int. j. numer. methods fluids*, **12,** 683–704 (1991).
4. P. Carriere and D. Jeandel, 'A 3D finite element method for the simulation of thermoconvective flows and its performances on a vector-parallel computer', *Int. j. numer. methods fluids*, **12,** 929–946 (1991).
5. R. Thatcher and D. J. Silvester, 'A locally mass conserving quadratic velocity, linear pressure element', *Numerical Analysis Report*, **147,** University of Manchester, Manchester, UK (1987).
6. D. Tidd, R. Thatcher and A. Kaye, 'The free surface of Newtonian and non-Newtonian fluids trapped by surface tension', *Numerical Analysis Report*, **127,** University of Manchester, Manchester, UK (1986).
7. M. Gunzburger, *Finite Element Methods for Viscous Incompressible Flows*, Academic Press, New York, 1989.
8. M. Fortin, 'Old and new finite elements for incompressible flows', *Int. j numer. methods fluids*, **1,** 347–364 (1981).
9. M. Fortin and A. Fortin, 'A generalization of Uzawa algorithm for the solution of the Navier–Stokes equations', *Commun. appl. numer. methods*, **1,** 205 (1985).
10. M. Seager, 'A SLAP for the Masses', *Lawrence Livermore Nat. Laboratory Technical Report, UCRL-100267*, 1988.
11. S. V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, New York, 1980.
12. R. L. Sani, R. 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,** 17–41 (1981).
13. C. Vincent and R. Boyer, 'A preconditioned conjugate gradient Uzawa-type method for the solution of the Stokes problem by mixed Q1-P0 stabilized finite elements', *Int. j. numer. methods fluids*, **14,** 289–298 (1992).
14. D. J. Silvester and N. Kechkar, 'Stabilized bilinear-constant velocity–pressure finite elements for the conjugate gradient solution of the Stokes problem', *Comput. Methods Appl. Mech. Eng.*, **79,** 71–86 (1990).
15. I. N. Sneddon, *Fourier Transforms*, McGraw-Hill, New York, 1951.