ELSEVIER

Journal of Computational and Applied Mathematics 79 (1997) 147–165

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

An element-by-element BICGSTAB iterative method for three-dimensional steady Navier–Stokes equations[1]

Morten M.T. Wang, Tony W.H. Sheu*

Department of Naval Architecture and Ocean Engineering, National Taiwan University, 73 Chou-Shan Road, Taipei, Taiwan, Republic of China

Received 13 June 1995

Abstract

Construction of a stabilized Galerkin upwind finite element model for steady and incompressible Navier–Stokes equations in three dimensions is the main theme of this study. In the time-independent context, the weighted residuals statement is kept biased in favor of the upstream flow direction by adding an artificial damping term of physical plausibility to the Galerkin framework. This upwind approach has significant advantage of seeking solutions free from cross-stream diffusion error. Finite element solutions have been found by mixed formulation, implemented in quadratic cubic elements which are characterized as possessing the so-called LBB (Ladyzhenskaya–Babuška–Brezzi) condition. An element-by-element BICGSTAB solution solver is intended to alleviate difficulties regarding the asymmetry and indefiniteness arising from the use of a mixed formulation for incompressible fluid flows. The developed three-dimensional finite element code is first rectified by solving a problem amenable to analytic solution. A well-known lid-driven cavity flow problem in a cubical cavity is also studied.

Keywords: Steady; Incompressible Navier–Stokes; Mixed formulation; BICGSTAB

AMS classification: 65; 76

1. Introduction

Many engineering problems of practical importance have strong association with incompressible flows. Besides the lack of turbulence modeling and the deficiency of computer capacity which prevent realistic simulation, technical difficulties in obtaining a solution for this class of Navier–Stokes equations lie also in the demand for the divergence-free constraint condition. Numerous research activities have been devoted to arriving at this discrete solenoidality. As discussed by Gunzburger [1], major approaches so far explored differ in their way of enforcing this constraint

* Corresponding author. e-mail: sheu@indy.na.ntu.edu.tw.

[1] This research was supported by funds from the National Science Council under the contract of NSC84-2212-E-002-060.

PII S0377-0427(96)00172-0

condition. As an alternative to the most commonly recognized Pressure Poisson equation (PPE) approach [2], the pseudo-compressibility method [3] is also attractive. In segregated approaches, the pressure–velocity coupling is gradually retained, thereby ensuring solenoidality at convergence. We successively update primitive velocities from momentum equations and then the working pressure from a pressure Poisson equation. A fundamental impediment to the application of this method to steady flow analysis is the question of what boundary conditions are to be used. This ambiguity leads us to adopt the mixed formulation.

Finite element simulations for problems involving incompressibility and viscosity suffer from serious oscillations from two main sources. The first source deals with the advective character of the equations. The second source is the absence of pressure in the continuity equation when using mixed formulation methods. While the incompressibility constraint is satisfied a priori in using a mixed finite element model, the advantage is obtained at sacrifice of the indispensable LBB stability condition [4–6]. This confines finite element interpolation functions to limited choices. The task of establishing guidance in determining whether or not an element is endowed with the LBB condition involves a profound proof that requires skill and solid mathematical discipline. For a three-dimensional analysis, it is not yet conclusive whether one pair of interpolation function should be abandoned in favor of the other one. In this work, a convergent cubical element will be investigated.

The second type of oscillations deals strongly with high Reynolds or Peclet number flows. Numerical solutions are often corrupted by node-to-node oscillations when a Galerkin method is employed. Unsymmetric treatment of convection terms has been identified as an effective ingredient in resolving these pathologic oscillations. The solutions, unfortunately, are overdiffusive. Along a direction normal to the streamline, in particular, the deterioration of accuracy is clearly evidenced in circumstances when angles of flow motion are not well-considered. This motivates us to develop a stable and accurate multi-dimensional flux discretization scheme in domains covered by quadratic elements. A wealth of description of upwinding alternatives has been given by Donea [7], among which the streamline upwind Petrov–Galerkin (SUPG) model [8] is free of cross-stream diffusion error and has gained wider acceptance. The present paper follows the concept of SUPG to analyze steady Navier–Stokes equations on quadratic elements in three space dimensions.

In addition to problems with the LBB constraint condition and the much larger matrix equation arising from the mixed formulation, the lack of symmetry and definiteness poses another challenge. The Gaussian-elimination-based direct method has long been considered to be the only viable means of resolving these difficulties. In three-dimensional simulations, the demand for the continued fill-in in the course of finding a solution using direct solver is beyond the scope of the present computer technology. Research scientists in the finite element community consequently have no choice but to turn towards iterative solvers. We are interested in iterative solvers that can circumvent difficulties, typical of the mixed formulation where a large portion of the zero-diagonal terms may exist in the unsymmetric stiffness matrix.

2. Basic formulation

To obtain steady-state finite element solutions, we directly analyze the time-independent Navier–Stokes equations in steady state instead of its time-dependent counterparts. For incompressible fluid flows, mathematical formulations can be numerous, depending on the choice of primitive,

vorticity-vector potential, and vorticity-velocity working variables. For a general review of the peculiarities of these formulations, the reader is referred to [1, 2], which provide a large number of valuable references. Among these methods, the primitive-variable formulation is most suitable and has an appealing advantage in that it has closure boundary conditions.

In a domain Ω of three dimensions, the target equations considered are written as

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

$$(\boldsymbol{u} \cdot \nabla \boldsymbol{u}) + \nabla p - \frac{1}{Re} \nabla^2 \boldsymbol{u} = 0. \tag{2}$$

This elliptic differential system is subject to the following Dirichlet type of boundary condition at $\partial\Omega = \Gamma$:

$$\boldsymbol{u} = \boldsymbol{g}, \tag{3}$$

where

$$\int_\Gamma \boldsymbol{n} \cdot \boldsymbol{g} \, \mathrm{d}\Gamma = 0. \tag{4}$$

The Reynolds number $Re = \rho u_{\mathrm{ref}} L_{\mathrm{ref}} / \mu$ comes forth as a result of carrying out a dimensionless procedure.

While being very practical, segregated approaches lack the pressure data pertaining to the boundary [9]. With this understanding, we prefer to use the mixed formulation in solving four primitive unknowns in a fully coupled manner. The mixed variational problem characterizing Eqs. (1)–(4) takes the following explicit artificial viscosity form: Along with the mesh-size-dependent damping operator d_c and the essential boundary condition $\boldsymbol{u} = \boldsymbol{g}$ on $\partial\Omega$, find the velocity–pressure pair $(\boldsymbol{u}, p) \in V \equiv (H_0^1 \times H_0^1) \times P$ from

$$\int_\Omega (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} \cdot \boldsymbol{w} \, \mathrm{d}\Omega + \frac{1}{Re} \int_\Omega \nabla \boldsymbol{u} : \nabla \boldsymbol{w} \, \mathrm{d}\Omega - \int_\Omega p \nabla \cdot \boldsymbol{w} \, \mathrm{d}\Omega = \int_\Omega \boldsymbol{d}_c \cdot \boldsymbol{w} \, \mathrm{d}\Omega, \tag{5}$$

$$\int_\Omega (\nabla \cdot \boldsymbol{u}) q \, \mathrm{d}\Omega = 0. \tag{6}$$

This variational statement holds for all admissive functions $\boldsymbol{w} \in H_0^1(\Omega) \times H_0^1(\Omega)$ and $q \in L^2(\Omega)/\mathscr{R}$. Important to note is that the consistency property is met as the grid spacing approaches zero.

2.1. Interpolation functions for pressure and velocities

At first glance, we are tempting to say that working variables in the weak variational system, as defined by Eqs. (5) and (6), can be independently approximated through shape functions as one wishes. However, employment of combinations that do not satisfy a compatibility condition (or inf–sup, or Babuška Brezzi condition) may yield undesirable pathologies in the approximation of pressure and velocity as well. The task of clarifying whether an element has the LBB condition is quite involved, especially for three-dimensional analyses. Commonly, we can perform a simple test by computing the constraint ratio so as to be able to loosely examine whether the investigated

pressure–velocity element falls into the class of stabilized element pairs. For the sake of programming simplicity, finite elements constructed in this paper belong to the uni-variant instead of the multi-variant family [10]. Here, primitive velocities are approximated by the following triquadratic interpolation functions N^i $(i = 1\text{–}27)$:

$$N^i = (\tfrac{3}{2}\overline{\xi}^2 + \tfrac{1}{2}\overline{\xi} + 1 + \xi^2 - \xi_i^2)(\tfrac{3}{2}\overline{\eta}^2 + \tfrac{1}{2}\overline{\eta} + 1 + \eta^2 - \eta_i^2)(\tfrac{3}{2}\overline{\zeta}^2 + \tfrac{1}{2}\overline{\zeta} + 1 + \zeta^2 - \zeta_i^2), \tag{7}$$

where

$$\overline{\xi} = \xi\xi_i, \quad \overline{\eta} = \eta\eta_i, \quad \overline{\zeta} = \zeta\zeta_i.$$

We denote ξ_i, η_i, and ζ_i as the normalized coordinates for the ith node. In a continuous context, pressure variables are approximated by degrees of freedom located at eight corners of the element via the following shape functions:

$$M^i = \tfrac{1}{8}(1 + \overline{\xi})(1 + \overline{\eta})(1 + \overline{\zeta}). \tag{8}$$

Since it is characterized as possessing a simple data structure and the inf–sup compatibility condition, the finite element defined by Eqs. (7) and (8) can be regarded as very attractive.

2.2. *Weighting function in* SUPG

The solution quality in simulating a flow problem of convection dominance depends on, among other factors, the accuracy and the stability yielded by a discretization scheme. While second-order spatial accuracy can be simply acquired by applying a center-based Galerkin approximation to the differential equations of interest, it simultaneously introduces unwanted velocity oscillations. In order to cope with this drawback, large numbers of upwind finite element models have been developed. Most notable among them are the currently popular Petrov–Galerkin methods which have had considerable success. An alternative to this class of methods is adding an artificial damping term, d_c, as shown in Eq. (5), to the underlying Galerkin formulation, thus rendering a biased scheme.

There remains the choice of an appropriate artificial damping term to suppress oscillatory velocities while maintaining a good degree of accuracy. Like the explicit artificial viscosity finite difference setting, finite element solutions deteriorate owing to the addition of a damping mechanism. In a multi-dimensional flow analysis, the degree of deterioration is particularly significant as the velocity vector deviates quite a lot from the grid lines. To reduce these false diffusion errors, research into the design of d_c is necessary. The underlying principle is to provide stabilized terms to spatial locations where they are needed most without sacrifice of solution accuracy. In order to acquire sufficient stability along the flow direction, we introduce a tensorial-type damping term, d_c, as follows:

$$\boldsymbol{d}_c = \tau|\boldsymbol{u}|^2\boldsymbol{u}_{,ss}. \tag{9}$$

The built-in coefficient τ largely accounts for the solution accuracy.

By substituting finite element approximations, namely $\boldsymbol{u}^h = \sum \boldsymbol{u}^i N^i$, $p^h = \sum p^i M^i$, into the Galerkin statement, we can derive the following matrix equations:

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{10}$$

where

$$A = \int_{\Omega}^{h} \begin{Bmatrix} C^{il} & 0 & 0 & -M^l \dfrac{\partial N^i}{\partial X_1} \\ 0 & C^{il} & 0 & -M^l \dfrac{\partial N^i}{\partial X_2} \\ 0 & 0 & C^{il} & -M^l \dfrac{\partial N^i}{\partial X_3} \\ M^i \dfrac{\partial N^l}{\partial X_1} & M^i \dfrac{\partial N^l}{\partial X_2} & M^i \dfrac{\partial N^l}{\partial X_3} & 0 \end{Bmatrix} \mathrm{d}\Omega^h,$$

$$\boldsymbol{x} = \{V_1^l, V_2^l, V_3^l, P^l\}^{\mathrm{T}},$$

$$C^{il} = (N^i + \bar{B}^i)(N^l \tilde{V}_j^l)\frac{\partial N^l}{\partial X_j} + \frac{1}{Re}\frac{\partial N^l}{\partial X_j}\frac{\partial N^l}{\partial X_j},$$

$$\bar{B}^i = \tau(N^l \tilde{V}_j^l)\frac{\partial N^i}{\partial X_j},$$

$$\boldsymbol{b} = -\int_{\Gamma_{\mathrm{out}}^h} N^i \begin{Bmatrix} pn_x - \dfrac{1}{Re}\dfrac{\partial u^j}{\partial n} \\ pn_y - \dfrac{1}{Re}\dfrac{\partial v^j}{\partial n} \\ pn_z - \dfrac{1}{Re}\dfrac{\partial w^j}{\partial n} \\ 0 \end{Bmatrix} \mathrm{d}\Gamma^h.$$

2.3. Discrete divergence-free velocities

Flow simulation for an incompressible fluid is an area of abundant research. Besides exempting finite element solutions from the checkerboard pressure mode, we demand also the satisfaction of the discrete divergence-free constraint condition. To demonstrate that the incompressibility property is accessible, we will carry out a modified equation analysis. On a grid system of uniform grid size h, the resulting modified equation for the discretized continuity equation takes the following form:

$$\nabla \cdot \boldsymbol{u} = -\tfrac{1}{3}h^2[(u_{,xxx} + u_{,xyy} + u_{,xzz}) + (v_{,xxy} + v_{,yyy} + v_{,yzz}) + (w_{,xxz} + w_{,yyz} + w_{,zzz})] + \mathrm{O}(h^4). \tag{11}$$

By invoking the Lax equivalent theorem, the convergence solution is ensured.

2.4. Determination of artificial viscosity τ

The guideline behind the determination of artificial viscosity concerns the flow orientation and solution accuracy. Therefore, analytic representation of τ is most desirable. When extended to multi-dimensional analysis, algebraic manipulations become considerable. The representation of τ is still

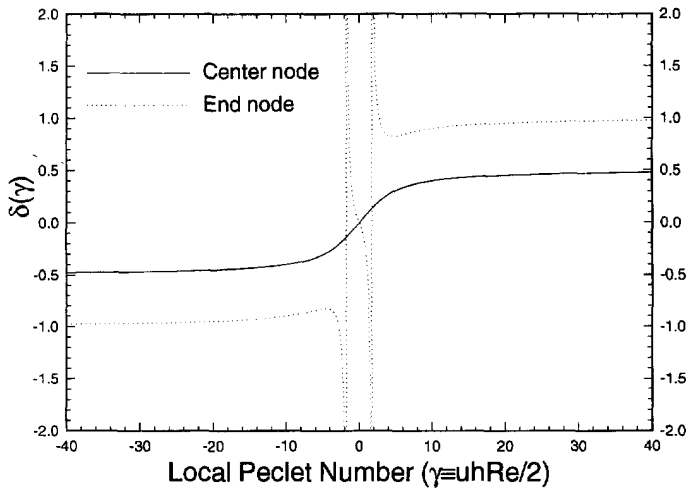


Fig. 1. Illustration of $\delta(\gamma)$, as defined in Eq. (13), against the local Peclet number γ.

analytically intractable, even in uniform grids. To keep the solution as accurate as possible, we have no choice but to adopt operator splitting approximations. This yields the proposed artificial viscosity:

$$\tau = \frac{\alpha_\xi V_\xi h_\xi + \alpha_\eta V_\eta h_\eta + \alpha_\zeta V_\zeta h_\zeta}{2V_j V_j}, \tag{12}$$

where

$$\alpha_{Y_i} = \delta(\gamma_{Y_i}),$$

$$\gamma_{Y_i} = \tfrac{1}{2} V_{Y_i} h_{Y_i} Re,$$

$$V_{Y_i} = \hat{\mathbf{e}}_{Y_i} \cdot \boldsymbol{u},$$

with Y_i being the local coordinate ξ, η, and ζ.

The case considered to render the yet undetermined functional form of $\delta(\gamma)$ is that of a single dimension since it is more amenable to analysis. The choice of the following $\delta(\gamma)$ leads to a variant of SUPG for which the discrete solution is nodally exact for the entire domain spanned by quadratic elements:

$$\delta(\gamma) = \begin{cases} \tfrac{1}{2}\coth(\gamma/2) - \dfrac{1}{\gamma} & \text{at center nodes,} \\ \dfrac{\cosh(\gamma) - 2}{\sinh(\gamma) - 4\tanh(\gamma/2)} - \dfrac{1}{\gamma} & \text{at end-nodes.} \end{cases} \tag{13}$$

Indicated in the plot of $\delta(\gamma)$, as shown in Fig. 1, against the Peclet number, there exists an undesirable localized jump at corner points. This is a clear manifestation of the loss of ellipticity. To solve this problem, we simply replace δ at corner nodes with those at center nodes. While the nodally exact solution is optimal in the sense of being considered in the one-dimensional context, it by no means provides an analytic solution in multiple dimensions by means of the operator splitting approximations.

3. Element-by-element iterative solver

3.1. Iterative methods for dealing with asymmetry

In situations where the convection effect is significant, finite element matrix equations arising from the steady Navier–Stokes equations take an unsymmetric form. For incompressible fluid flows, too many zeros remain in the diagonal of the matrix equations. As a result, the violation of the diagonal dominance property may worsen the distribution of condition numbers. In the absence of symmetry, complex eigenvalues cause the matrix equations arising from the mixed formulation to become indefinite. In these circumstances, the matrix equations are proven to be nearly singular, as evidenced by the computed eigenvalues in Fig. 2. With the help of Figs. 3 and 4, we can clearly point out the restriction of using unequal-order basis functions for the working variables. Besides very small number of Krylov subspace iterative solvers, a finite element solution for a peculiar matrix, as shown in Figs. 3 and 4, can be obtained only from solution solvers underlying the Gaussian elimination framework. As is usual, direct solvers are the favorable strategy in solving a matrix

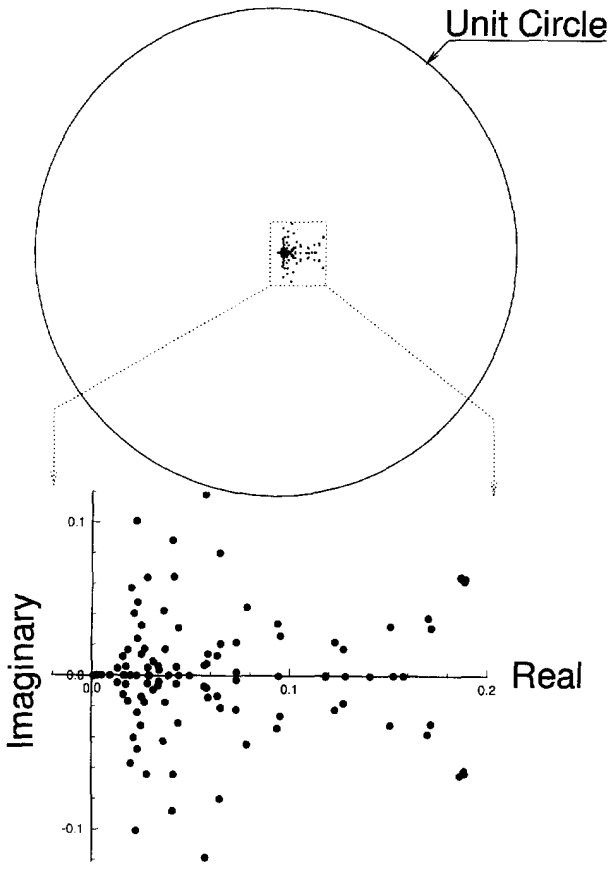


Fig. 2. Eigenvalues computed from the problem, defined in Section 4.1, in a domain containing $2 \times 2 \times 2$ elements.

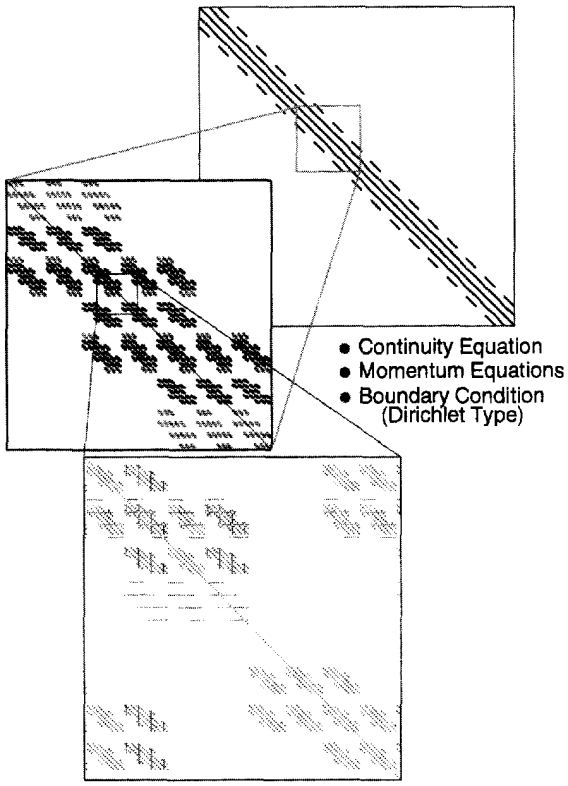


Fig. 3. Illustration of the non-zero profile in the matrix equation for the $16 \times 16 \times 16$ triquadratic meshes. Green, red, and blue dots represent the non-zeros contributed by continuity, momentum, and boundary condition, respectively.

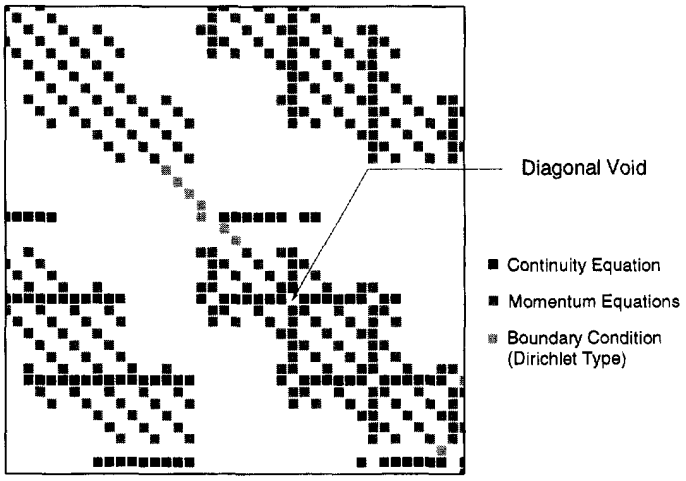


Fig. 4. Illustration of the diagonal void, due to the continuity equation, in the matrix equation.

equation involving only two dimensions. For very large problems, the demands on storage become prohibitive in the course of fill-in processes and this calls for iterative solvers.

Usually, iteration numbers increase dramatically with the increase of grid points. For this reason, polynomial acceleration is most applicable to a truly three-dimensional Navier–Stokes flow simulation. Typical of this class of methods is the conjugate gradient method of Hestenes and Stiefel [11]. The essence of this method is to replace the matrix equation problem with the minimization problem. Starting from the initial vector x_0, this method solves for x from $Ax = b$ iteratively via a sequence of $x_1, x_2, \ldots$:

$$x_i \in x_0 + \text{span}(r_0, Ar_0, \ldots, A^{i-1} r_0),$$

where

$$r_0 = b - Ax_0.$$

The span $(r_0, Ar_0, \ldots, A^{i-1}r_0)$ is known as the Krylov subspace. This optimization procedure works effectively only for a matrix equation having clustered eigenvalues. When asymmetry is experienced, pivoting breakdown is inevitable.

Research into the design of Krylov subspace methods to cope with matrix asymmetry is an active field, and new methods are still emerging. One possibility is to deal with normal equations, among which the Conjugate Gradient method on the normal residual (CGNR) and normal equations (CGNE) [12] are typical examples. It is fair to say that no computational benefit will be gained from this normalization effort to resolve the asymmetry problem since the condition number for an equivalent normal equation becomes much larger than that of the original stiffness matrix A.

In non-stationary iterative methods, two classes of methods capable of resolving matrix asymmetry are frequently referred to. In the first catalog, a polynomial acceleration procedure known as the Chebyshev method works only for positive-definite equations. This method requires knowledge of the spectrum a priori. Methods underlying the idea of orthogonalization fall into the class of Krylov subspace methods. To circumvent deficiencies inherent in the bi-conjugate graduate (Bi-CG) method [13], such as the irregular convergence behavior and the indispensable transpose operation of the coefficient matrix, the Arnoldi or Lanczos algorithm was proposed. As with the Arnoldi algorithm, the generalized minimized residuals (GMRES) method [14] accommodates a self-orthogonal sequence. Due to the restriction of more prohibitive storage demand, the residual is minimized optimally. In the course of iteration, convergence of no more than n steps for a matrix $A_{n\times n}$ will be expected. The remedy is to incorporate a restart capability. This, however, requires skill and experience. In the Lanczos context, product methods such as conjugate gradient squares (CGS) [15], quasi-minimal residual (QMR) [16], and bi-conjugate gradient stabilized (BICGSTAB) [17], are popular remedies for asymmetry. Product methods have dual orthogonal vector sets but distinguish themselves in the combination of construction polynomials. The QMR method of Freund and Nachtigal [16] avoids the irregular convergence behavior but still suffers from the necessity of transposing the stiffness matrix. CGS, on the other hand, obviates the need for A^T but inhibits the irregular convergence behavior because this method accommodates the contraction polynomial of Bi-CG. Besides the transpose-free version of QMR [18], the BICGSTAB method of Van der Vorst [17] is another viable alternative. BICGSTAB was developed under the framework of the Lanczos method while the local minimization is provided by GMRES(1). In manipulation of equal-order contraction

polynomials of different kinds, we can get rid of the irregular convergence behavior and transpose matrix. Nevertheless, there is much work still being done in the context of the avoidance of pivoting breakdown and Lanczos breakdown mainly because BICGSTAB still inhibits some features of Bi-CG.

3.2. Element-by-element BICGSTAB method

The effectiveness of an iterative method depends highly on the profile of the nonzeros of the coefficient matrix. In this regard, the strategies of ordering nodal points and allocating working variables become important because they directly determine the bandwidth of the coefficient matrix. In a matrix, the bandwidth accounts for sparsity. Given a sparse matrix, which is common to finite element analysis, a means of effectively storing this matrix in the core memory is needed. Just as with the compressed matrix in the finite difference setting, we can store a matrix at the element base in order to dispense with unnecessary storage of voids. This motivates us to devise an Element-by-Element (EBE) procedure and incorporate it into the present employed BICGSTAB iterative solver. The resulting EBE-BICGSTAB procedures can be briefly described as follows:

Compute $\boldsymbol{r}_0 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_0$ for some initial guess $\boldsymbol{x}_0$
Choose $\bar{\boldsymbol{r}}$, such that $(\bar{\boldsymbol{r}}, \boldsymbol{r}_0) \neq 0$
For $i = 1, 2, \ldots$
  $\rho_{i-1} = (\bar{\boldsymbol{r}}, \boldsymbol{r}_{i-1})$
  if $\rho_{i-1} < \varepsilon_1$ **(near breakdown)**
  if $i = 1$
    $\boldsymbol{p}_i = \boldsymbol{r}_{i-1}$
  else
    $\beta_{i-1} = (\rho_{i-1}/\rho_{i-2})(\alpha_{i-1}/\omega_{i-1})$
    $\boldsymbol{p}_i = \boldsymbol{r}_{i-1} + \beta_{i-1}(\boldsymbol{p}_{i-1} - \omega_{i-1}\boldsymbol{v}_{i-1})$
  endif
  solve $\boldsymbol{K}\bar{\boldsymbol{p}} = \boldsymbol{p}_i$
  $\boldsymbol{v}_i = \sum_{\text{elem}}(\boldsymbol{A}_{\text{elem}}\bar{\boldsymbol{p}})$ ⟵ element-by-element procedure
  $\alpha_i = \rho_{i-1}/(\bar{\boldsymbol{r}}, \boldsymbol{v}_i)$
  if $(\bar{\boldsymbol{r}}, \boldsymbol{v}_i) < \varepsilon_2$ **(near breakdown)**
  $\boldsymbol{s} = \boldsymbol{r}_{i-1} - \alpha_i \boldsymbol{v}_i$
  solve $\boldsymbol{K}\bar{\boldsymbol{s}} = \boldsymbol{s}$
  $\boldsymbol{t} = \sum_{\text{elem}}(\boldsymbol{A}_{\text{elem}}\bar{\boldsymbol{s}})$ ⟵ element-by-element procedure
  if $\|\boldsymbol{s}\|_2 < \varepsilon$
    $\omega_i = 0$
  else
    $\omega_i = (\boldsymbol{t}, \boldsymbol{s})/(\boldsymbol{t}, \boldsymbol{t})$
  endif
  $\boldsymbol{x}_i = \boldsymbol{x}_{i-1} + \alpha_i \bar{\boldsymbol{p}}_i + \omega_i \bar{\boldsymbol{s}}$
  $\boldsymbol{r}_i = \boldsymbol{s} - \omega_i \boldsymbol{t}$
  check convergence; continue if necessary $(\omega_i \neq 0)$
End

In contrast to the low-order transient analysis, the number of nonzero terms encountered herein, as seen in Fig. 3, is much larger, say from 100 to 230. Beside this, the values of these nonzeros have the same order. Tolerances for outer and inner iterations are set at 10^{-6} and 10^{-9}, respectively. The solution will be relinearized when "near breakdown" or decrease of the error norm by several orders in inner iteration occurs.

4. Computed results

4.1. Analytic Navier–Stokes equations in three dimensions

As a first step towards the verification of the developed code, it is best to solve an analytic problem. The assessment must focus on solution accuracy and convergence behavior for the underlying BICGSTAB iterative method. To this end, we have carried out the finite element calculation

Table 1
Comparison of L_2 error norms obtained from frontal and BICGSTAB solvers. Here, u and p represent velocity vector and pressure, respectively

Solver		Number of elements		
		4^3	8^3	16^3
D[a]	u	8.46708×10^{-3}	4.69061×10^{-3}	–
	p	2.99745×10^{-2}	1.40210×10^{-2}	–
I[b]	u	8.46709×10^{-3}	4.69064×10^{-3}	2.27519×10^{-3}
	p	2.99781×10^{-2}	1.40467×10^{-2}	7.15831×10^{-3}

[a] D denotes direct Frontal solver.
[b] I denotes BICGSTAB solver.

Table 2
Comparison of convergent order and CPU time (s) between frontal and BICGSTAB solvers

Solver			Number of elements		
			4^3	8^3	16^3
Frontal	co[a]	u		0.8521	–
		p		1.0961	–
	CPU		1047	16992	–
BICGSTAB	co	u		0.8521	1.0438
		p		1.0937	0.9725
	CPU		1381	21281	180810

[a] co denotes the convergent order.

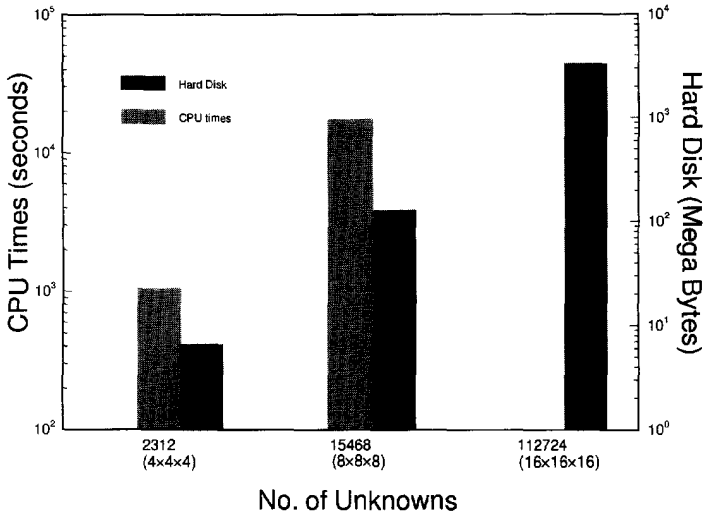


Fig. 5. Plots of CPU times and disk spaces against number of unknowns in using a frontal direct solver.

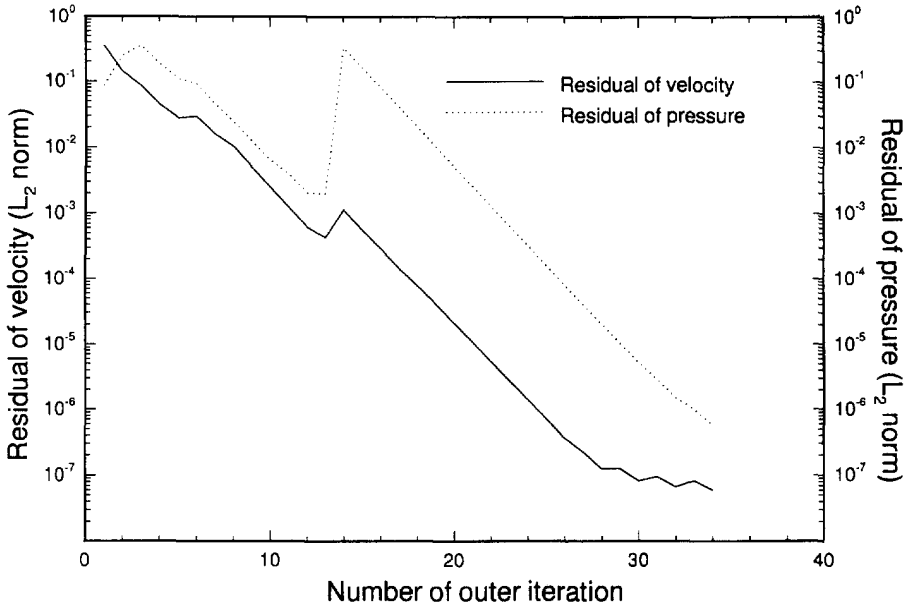
in a cubical hexahedron of length 1. Along the boundary surfaces, nodal velocities are analytically prescribed:

$$\begin{aligned} u &= \tfrac{1}{2}(y^2 + z^2), \\ v &= -z, \\ w &= y. \end{aligned} \tag{14}$$

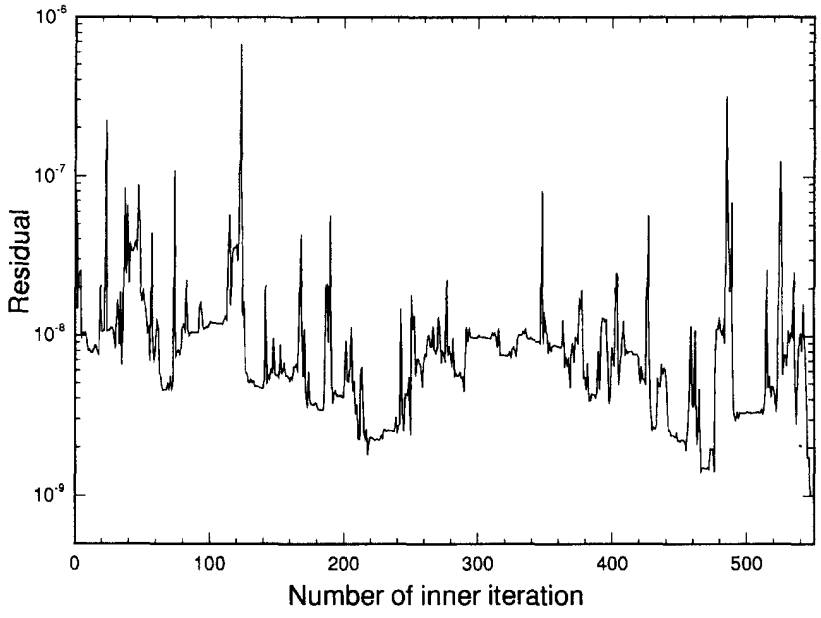
In accordance with above boundary velocities, the exact pressure takes the following form:

$$p = \tfrac{1}{2}(y^2 + z^2) + \frac{2}{Re}x. \tag{20}$$

For a given total number of unknowns, $\mathcal{N}$, the demand for the memory is estimated to be $O(\mathcal{N}^{4/3})$ for a frontal solver, from which we realize that it is impossible to solve a large size problem by using a frontal solver. Also, the disk space needed is also prohibitive, as seen in Fig. 5 which is plotted according to the present calculation. In the BICGSTAB solver, the memory needed is only $O(\mathcal{N})$, and it is applicable to practical problems. Three uniform grids have been used to perform a convergent order test. Table 1 tabulates the computed L_2 errors, which indicate that BICGSTAB iterative solutions are compatible with frontal direct solutions. To illustrate the error reduction, it is worthwhile to plot L_2-residuals against iteration numbers. Fig. 6 illustrates error reductions both for an inner and the whole outer iteration loops. The convergent order and CPU time are summarized in Table 2. From these results, we can see that direct frontal solvers are only preferable to iterative solvers for smaller size problems. Much of the CPU time has been spent on the inner iteration sequences for the analysis proceeding an outer iteration. This points out how difficult the unsymmetric and indefinite matrix equations can be dealt with. To illuminate the percentage of CPU time spent on the inner iterations, we plot Fig. 7.



(a) outer iteration sequences



(b) 18th inner iteration loop

Fig. 6. Illustration of L_2 error reduction against iterations: (a) outer iteration (where convergence tolerance is set to 10^{-6}), (b) 18th inner iteration loop (where the value of convergence tolerance is set at 10^{-9}).

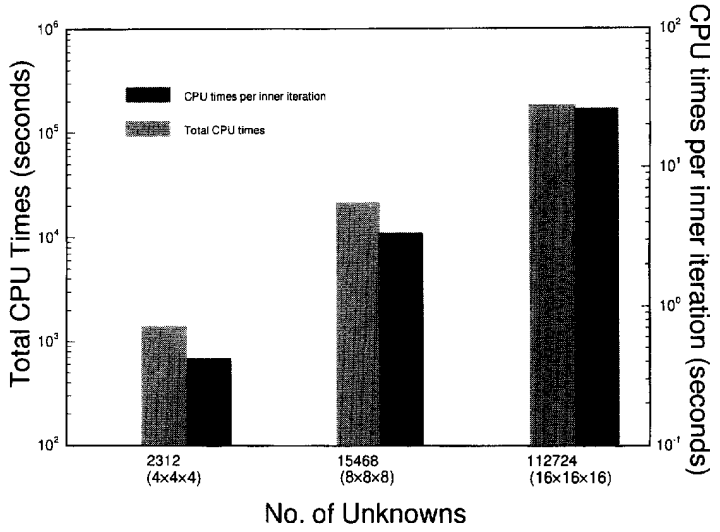


Fig. 7. Plots of CPU times in total and per inner iteration against the number of unknowns in using the BICGSTAB iterative solver.

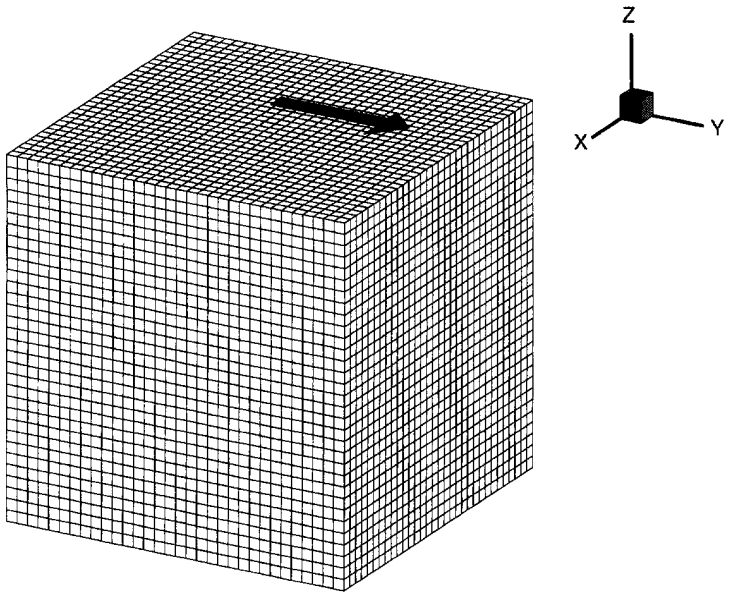


Fig. 8. Configuration of the lid-driven cubical cavity.

4.2. Lid-driven cubical cavity flow problem

A benchmark test, configured as a lid-driven cavity problem, is now considered. The configuration of interest is illustrated in Fig. 8, where a unit velocity is given on the top surface of the cubical cavity. Being simple in geometry while exhibiting complex flow behavior, this problem is an ideal

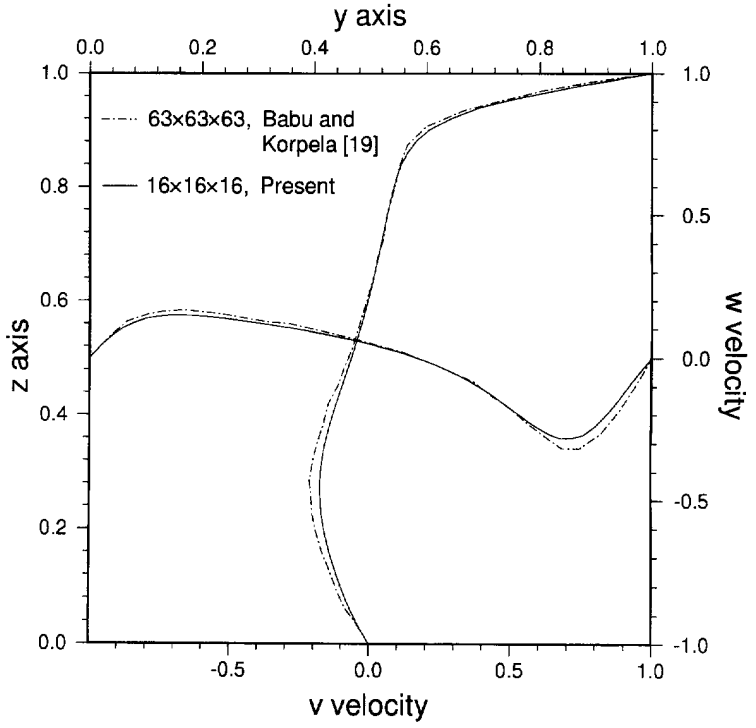


Fig. 9. Comparison of velocity profiles along the vertical and horizontal centerlines on the symmetric plane, $x = 0.4$, at $Re = 400$.

setting to benchmark the developed computer code. On the basis of $Re = 400$, calculations were performed in a cubic domain, containing $16 \times 16 \times 16$ tri-quadratic elements, shown in Fig. 8.

Since the investigated Reynolds number was much less than 2000, the steady-state analysis was appropriate. We first validated the computed solutions by comparing them with those in [19]. As seen in Fig. 9, close agreement is demonstrated and supports the conclusion that the present approach can accurately predict incompressible flow behavior. Also, we plot computed velocity vectors at different planes in Fig. 10 for illustrating the secondary flow patterns. For completeness, we have plotted the error reduction sequences for both inner and outer iteration loops in Fig. 11. In this test, an order of error decrease is prescribed in the inner iteration. A manifestation of the less restrictive inner convergence tolerance is the very regular outer iteration where two orders of residuals are reduced. The inclusion of a stringent tolerance is surely at the cost of a much irregular inner iteration processes. In BICGSTAB iterative solvers, the CPU times consumed is problem-dependent. It implies that the physical complications will determine the overhead of the outer iterations. To illustrate this aspect, we plot in Fig. 12 the CPU times for both analytic and lid-driven cavity problems.

5. Conclusions

The objective of the present work was to construct a stabilized Galerkin finite element model in the physical domain which comprises quadratic elements. The analysis was intended to simulate

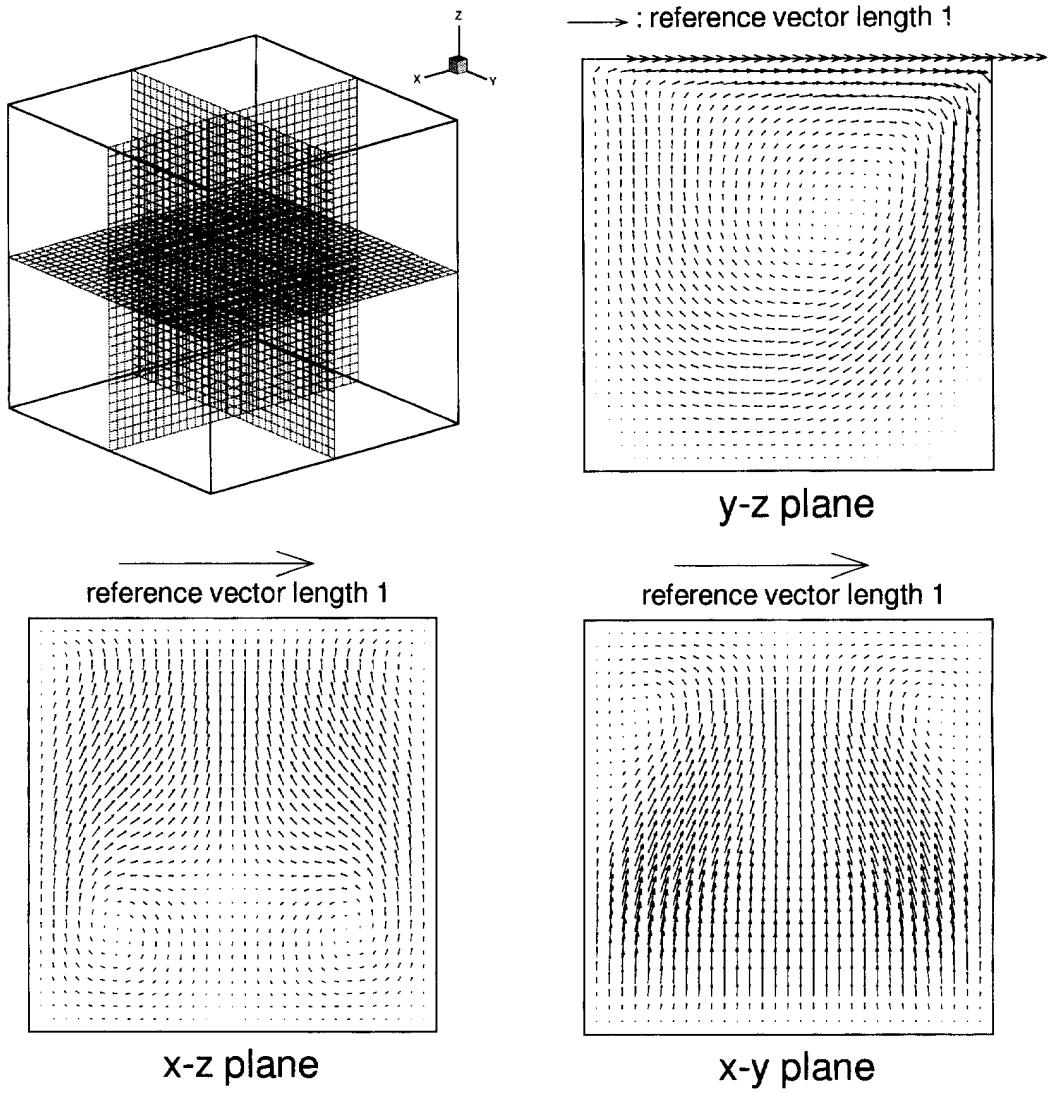


Fig. 10. Velocity vector plots at different planes.

three-dimensional incompressible Navier–Stokes flows. In situations where the convection effect was significant, extended stability was particularly needed along the flow direction. The streamline upwind feature has been clearly demonstrated in the formulation. The BICGSTAB solver has been implemented in the present mixed formulation to alleviate difficulties both in indefiniteness and asymmetry. As a basis for comparison, we have also considered a direct frontal solver which is the counterpart of BICGSTAB. Conclusions regarding the solvers are summarized as follows:

1. Usually, the direct frontal solver is more cost-effective and reliable than the employed BICGSTAB iterative solver when small size problems are considered. In contrast, the iterative BICGSTAB solver is more practical than its counterpart when the problem size is large. Also,

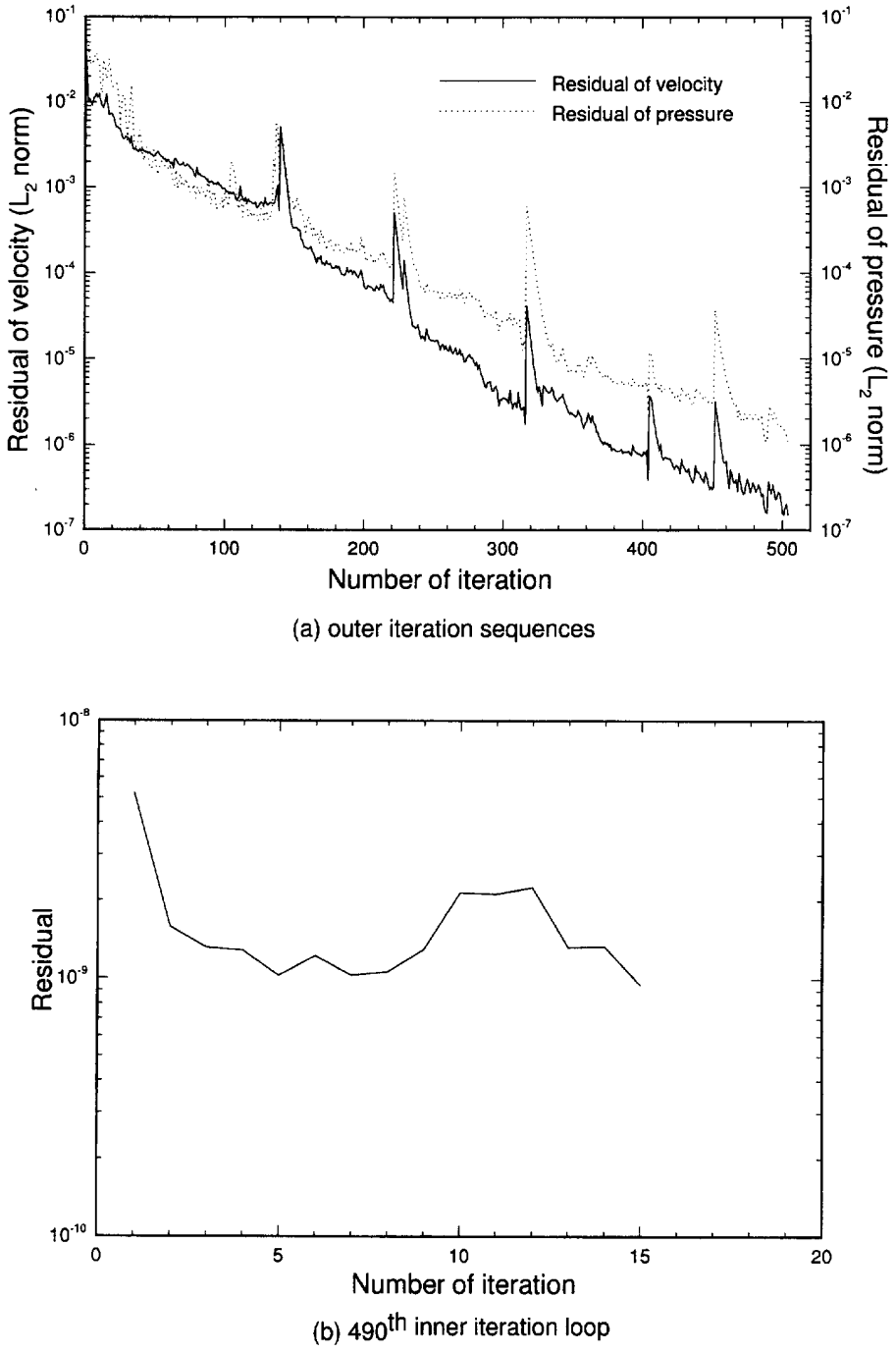


Fig. 11. Illustration of L_2 error reduction for the cubical lid-driven cavity flow problem at $Re = 400$: (a) outer iteration sequences; (b) 490th inner iteration loop.

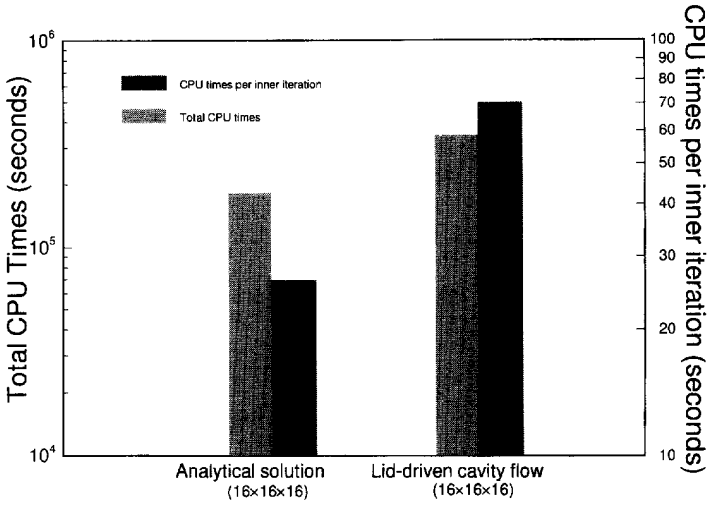


Fig. 12. Total CPU times and CPU times per inner iteration via two test problems for iterative BICGSTAB solver.

the quality of the BICGSTAB iterative solution is comparable with that of the frontal direct solution as long as the specified tolerance for the inner iteration loop is small enough.

2. The "near breakdown" phenomenon can be practically avoided by using a relinearized procedure. While near breakdown of solutions may occur in the course of inner iterations, outer iterations are little affected.

3. With a decrease by a few orders of the error norm in each inner iteration, the BICGSTAB solver can render a comparable solution by using a direct solution solver. The CPU time can also be reduced by using this strategy.

References

[1] V. Babu and S.A. Korpela, Numerical solution of the incompressible three-dimensional Navier–Stokes equations, *Comput. Fluids* **23** (1994) 675–691.

[2] Babuška, Error bounds for finite element methods, *Numer. Math.* **16** (1971) 322–333.

[3] F. Brezzi and J. Douglas, Stabilized mixed methods for the Stokes problem, *Numer. Math.* **53** (1988) 225–235.

[4] A.J. Chorin, A numerical method for solving incompressible viscous flow problems, *J. Comput. Phys.* **2** (1967) 12–26.

[5] J. Donea, Generalized Galerkin methods for convection dominated transport phenomena, *ASME Appl. Mech. Rev.* **44** (1991) 205–214.

[6] R. Fletcher, Conjugate gradient methods for indefinite systems, Lecture Notes in Mathematics **506** (1976) 73–89.

[7] R.W. Freund, A transpose-free quasi-minimum residual algorithm for non-Hermitian linear systems, *SIAM J. Sci. Comput.* **14** (1993) 470–482.

[8] R. Freund and N. Nachtigal, QMR: a quasi-minimal residual method for non-Hermitian linear systems, *Numer. Math.* **60** (1991) 315–339.

[9] P.M. Gresho and R.L. Sani, On pressure boundary conditions for the incompressible Navier–Stokes equations, *Int. J. Numer. Methods Fluids* **7** (1987) 1111–1145.

[10] M.D. Gunzburger, *Finite Element Methods for Viscous Incompressible Flows, A Guide to Theory, Practice, and Algorithms* (Academic Press, London, 1989).

[11] M. Hestenes and E. Stiefel, Methods of conjugate gradients for solving linear systems, *J. Research NBS* **49** (1952) 409–436.
[12] T.J.R. Hughes, W.K. Liu and A. Brooks, Finite element analysis of incompressible viscous fluid by the penalty function formulation, *J. Comput. Phys.* **30** (1979) 1–60.
[13] O. Ladyzhenskaya, *The Mathematical Theory of Viscous Incompressible Flow* (Gordon and Breach, New York, 1969).
[14] N. Nachtigal, S. Reddy and L. Trefethen, How fast are nonsymmetric matrix iterations? *SIAM J. Matrix Anal. Appl.* **13** (1992) 778–795.
[15] S.V. Patankar, *Numerical Heat Transfer and Fluid Flow* (Hemisphere, Washington, DC, 1980).
[16] Y. Saad and M.H. Schultz, GMRES: a generalized minimum residual algorithm for solving nonsymmetric linear system, *SIAM J. Sci. Statist. Comput.* **7** (1986) 856–569.
[17] T.W.H. Sheu and M.M.T. Wang, A comparison study on multivariant and univariant finite elements for three dimensional incompressible viscous flows, *Int. J. Numer. Methods Fluids* **71** (1995) 683–696.
[18] P. Sonneveld, CGS, a fast Lanczos-type solver for nonsymmetric systems, *SIAM J. Sci. Statist. Comput.* **10** (1989) 36–52.
[19] H.A. Van der Vorst, BI-CGSTAB: a fast and smoothly converging variant of BI-CG for the solution of nonsymmetric linear systems, *SIAM J. Sci. Statist. Comput.* **13** (1992) 631–644.