INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2004; **45**:1249–1268 (DOI: 10.1002/fld.734)

Pressure correction methods based on Krylov subspace conception for solving incompressible Navier–Stokes problems

Weiming Liu*,†, Mike Ashworth and David Emerson

Computational Science and Engineering Department, CCLRC Daresbury Laboratory, Warrington, WA4 4AD, U.K.

SUMMARY

Pressure correction concept is widely used to solve incompressible Navier–Stokes problems numerically. Based on Krylov subspace methods, we introduce several new pressure correction algorithms. Compared with the traditional pressure correction methods, they do not need to solve the pressure Poisson equation, which appears to reduce the computational cost. The preconditioning technique links the pressure correction methods based on Krylov iterations and with the pressure Poisson equation. In order to investigate the convergence performance of the new methods, we carried out various numerical experiments. Moreover we also discuss some ways on computational cost. Finally, these pressure correction methods are applied to solve the three-dimensional lid-driven cavity flows. © Crown Copyright 2004. Reproduced with permission of Her Majesty's Stationery Office. Published by John Wiley & Sons, Ltd.

KEY WORDS: pressure correction; Krylov subspace; incompressible flow; iterative method

1. INTRODUCTION

Numerical solutions of incompressible Navier–Stokes equations are a complex and significant topic. A lot of researchers contributed to this subject. Their works enrich our comprehension on this problem and provide us with a foundation to develop more effective algorithms. This paper reports authors' some attempt and experience to solve them by Krylov subspace methods. Its aim is to increase our knowledge and to get a better grasp of this problem.

Discretization of incompressible Navier–Stokes equations leads to a so-called saddle point system of equations. If the linear system of pressure is segregated from the saddle point system, we would find that the system matrix of the pressure equation is defined in an implicit formulation. Hence it is impossible to individually solve the linear equation of pressure independent of the momentum equations. In traditional pressure correction methods, the system matrixes of the momentum equations are approximated into simple forms. Then using the

*Correspondence to: Weiming Liu, Computational Science and Engineering Department, CCLRC Daresbury Laboratory, Warrington, WA4 4AD, U.K.
†E-mail: wliu@dl.ac.uk

Published online 26 May 2004

Received 27 December 2002
Revised 7 March 2004

simplified matrixes, we may construct the approximate pressure Poisson equation. The solution of the pressure Poisson equation is used as a correction to the previous pressure iteration. Repeating this correction step associated with solving the momentum equations, we may obtain a convergent solution of the saddle point system. However, the pressure correction procedure through solution of the pressure Poisson equation may be substituted by Krylov iterations. In this paper we study the four pressure correction methods which are derived based on Krylov subspace methods. The pressure correction methods of this type do not need to solve the discrete Poisson equation. They correct the pressure using the information of the residual as Krylov method. Under this framework, the traditional pressure correction algorithms with solving pressure Poisson equation, such as SIMPLE method, may be viewed as preconditioning techniques.

Now briefly outline the organization of this paper. Next section introduces discrete Navier–Stokes equations. Then in Section 3, we introduce four stationary and non-stationary Krylov subspace iteration or pressure correction methods for the saddle point problem yielded by discrete Navier–Stokes equations. The last part of this section addresses the preconditioning techniques to those iterative methods. Section 4 presents and discusses the results of numerical experiment on them, and also reports some ways on the computational cost. Section 5 shows an application which is used as testing global solution methods for incompressible Navier–Stokes problems. Eventually, Conclusions are given in Section 6.

2. DISCRETE NAVIER–STOKES EQUATIONS

Consider finite element solution of incompressible, time-dependent Navier–Stokes problems. Navier–Stokes equations are defined in the primitive variable formulation by

$$\frac{\partial u}{\partial t} + (u \cdot \nabla)u - v\Delta u + \dot{\nabla} p = f \quad \text{in } \Omega \tag{1}$$

$$\operatorname{div}(u) = 0 \quad \text{in } \Omega \tag{2}$$

$$u(x,0) = I(x) \quad \text{with} \quad \operatorname{div}(I) = 0 \quad \text{in } \Omega \tag{3}$$

Here u and p denote the unknowns, flow velocities and pressure, respectively. v is the viscosity of the fluid and f the known body force. Ω ($\subset R^d$, $d = 2,3$) denotes an open bound flow domain with a sufficiently smooth boundary Γ. It still requires appropriate boundary conditions to sufficiently define a solution of problem (1)–(3). For example, in finite element method, they are often given as

$$u(x,t) = b(x) \quad \text{on } \Gamma_1 \tag{4}$$

$$v\frac{\partial u}{\partial n} - \vec{n} p = 0 \quad \text{on } \Gamma_2 \tag{5}$$

Here $\Gamma = \Gamma_1 \cup \Gamma_2$, $\vec{\mathbf{n}}$ is the unit vector of the outward normal at Γ_2.

After temporal and spatial discretization and linearization of the above problems (1)–(5), one may obtain the following discrete saddle point system

$$\begin{pmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ g \end{pmatrix} \tag{6}$$

Here A is an $n \times n$ non-singular matrix which includes three parts. The first one is the consistent mass matrix that is symmetrical and positive semidefinite. The second is the non-symmetrical convection matrix, and the last the diffusion matrix with symmetrical and positive definite form. Clearly, the matrix A becomes symmetric positive definite for Stokes problem. B^{T} denotes an $n \times m$ matrix. It has full row rank for a stable element space in the LBB sense, and so system (6) is uniquely solvable. For an unstable element space in the LBB sense, the rank of B is less than m, and system (6) has more solutions, which are called spurious pressure modes. It requires a kind of stabilization technique to exclude these spurious pressure modes. The stabilized linear system of Equation (6) has unique solution and it may be generally expressed as

$$\begin{pmatrix} A & B^{\mathrm{T}} \\ B & -\chi C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ 0 \end{pmatrix} \tag{7}$$

Here the $m \times m$ matrix C is symmetrical and positive semidefinite [1] and χ denotes a small stabilized parameter. Unfortunately, some elements used popularly in practice do not satisfy Babuska–Brezzi condition, consequently the stabilized techniques are frequently employed in application. Lots efficient stabilized methods have been proposed for last two decades. The readers interested in them may see, for example, the literatures [2].

In principle, one can use direct Gaussian elimination type solvers to solve the linear system (6) or (7). But with increasing of size and physical complexity of calculated problems, the cost of the direct type solvers becomes expensive, both in terms of storage and CPU time. Consequently, most of the methods adopted in application are iterative algorithms. The iterative approaches may be classified into two different schemes according to the treatment of the unknowns, u and p. One is full-coupled schemes, in which it requires to store all the entries of the matrix in system (6) or (7). The other is the so-called segregated approaches, that is, the solutions of the linear system of the equation

$$Au + B^{\mathrm{T}} p = f \tag{8}$$

for u and

$$(BA^{-1}B^{\mathrm{T}} + \chi C)p = BA^{-1} f \tag{9}$$

for p are sought for in segregated formulation. Compared with the full-coupled scheme, the segregated algorithm only stores the matrix A and C of the linear system (7) during computation. As a result, it only needs about half memory of the full-coupled scheme. It is particularly significant for large-scale three-dimensional computation. The numerical algorithms in this paper belong to the methodology of segregated approaches.

For the sake of the completeness, we simply address some details implementing the discrete system (7). Navier–Stokes problem (1)–(5) is discretized on $Q_1 - P_0$ element space. Namely, in two-dimensional flows, the bilinear interpolation function is used for velocity components and pressure is taken into constant on each element. To three-dimensional cases, the trilinear interpolation function for flow velocities and pressure is constant on each element again. Unfortunately, the element $Q_1 - P_0$ does not satisfy Babuska–Brezzi condition. Therefore the discrete system (6) results in the checkerboard pressure mode in some cases. To filter the spurious pressure modes, we adopted the jump function method proposed by Hughes and Franca [3]. There are several linearization methods to handle the non-linear convection terms, such as, Picard approach or Newton–Raphson approach. Considering a large convergence radius of the Picard method, we use this method in this work. Since the numerical methods designed in this paper are for solving general incompressible flow problems, the chosen scheme for the temporal discretization is two-order implicit. The implicit scheme permits a larger time step and has more stable properties, compared with explicit schemes.

Now simply summarize the global iteration process as follows:

Begin Computation
- *Do Time Step Integration*
  - *Update last time step results*
- *Do Non-linear System Iteration*
  - *Update flow velocities and rebuild discrete saddle point system*
- *Do Solution of Linear Saddle Point System*
  - *Solving discrete saddle point system* (6) *or* (7) *in segregated methods*
- *End Do Solution of Linear Saddle Point System*
- *End Do Non-linear System Iteration*
- *End Do Time Step Integration*

Terminate Computation

3. PRESSURE CORRECTION METHODS BASED ON KRYLOV SUBSPACE ITERATION

In iterative solutions of the discrete system (8) and (9) in segregated formulation, actually, the iterative algorithm for the discrete pressure equation (9) is our concern centre. When using Krylov subspace iteration to the discrete pressure equation system (9), we need to calculate its residual for next iteration. The system matrix of the discrete pressure equation, however, is implicitly defined. Hence, the residual calculation requires to solve the momentum equation through taking the former iterative pressure as the known in advance. Then, the residual of the discrete pressure equation (9) may be decided by

$$r_k = BA^{-1}f - (BA^{-1}B^{\mathrm{T}} + \chi C)p_k = Bu_k - \chi C p_k \tag{10}$$

Moreover, the averaged convergence rate is defined as

$$\left(\frac{\|r_k\|}{\|r_0\|}\right)^{1/k} = \left(\frac{\|Bu_k - \chi C p_k\|}{\|Bu_0 - \chi C p_0\|}\right)^{1/k} \tag{11}$$

Having had the calculating residual formulation (10), we introduce four Krylov iterations for the discrete pressure equation (9) as follow. The former two of them are stationary iterations, and the last two are non-stationary. Note that the main procedures in the four algorithms are the same: (a) solving the momentum equation (8) under a given pressure (initial approximation or preceding iteration); (b) correcting the pressure when residual does not meet the convergence requirement; (c) returning to (a) until convergence. Consequently, when obtaining the solution of the discrete pressure equation, we also acquire the solution of the momentum equation (8). In the literatures the methods of these types are called pressure correction method.

3.1. Uzawa algorithms

The first scheme for saddle point system (8) and (9) used in this work is the classic Uzawa algorithm. It is derived using the least residual method and may be expressed as

Uzawa Algorithm

Give an initial approximation p_0 *of* p

for $k=1$ *until convergence, do*

$$\text{Solve } Au_k = f - B^{\mathrm{T}} p_k \tag{12}$$

$$\text{Calculate } p_{k+1} = p_k + \gamma(Bu_k - \chi C p_k) \tag{13}$$

end do

Here $\gamma > 0$ is a constant in iterative process. The Uzawa method only requires to solve the linear system (12), while the pressure is directly corrected through (13). Hence it is very simple to implement it. This extreme simplicity is the attractive property of Uzawa algorithm. A large number of practical numerical computation shows that Uzawa method is robust to solve saddle point problem though its rate of convergence is slow. Because of its robustness, it is frequently adopted nowadays in different forms, in particular, to non-linear system such as the discrete Navier–Stokes equations mentioned in last section.

3.2. Pressure correction on second-order Richardson iteration

In order to improve the convergence behaviour of Uzawa method, Liu and Xu [4] proposed a second-order Richardson algorithm for the saddle point system (8) and (9) by linear combination of Uzawa iterations. It is written as

Second-order Richardson Algorithm

Give an initial approximation p_0 *of* p

Solve u_0 *and* r_0

$Au_0 + B^{\mathrm{T}} p_0 = f$

$r_0 = Bu_0 - \chi C p_0$

Calculate η_0 *and* p_0

$\eta_0 = r_0/d$

$p_1 = p_0 + \eta_0$

for $k=1$ *until convergence, do*

Solve $Au_k + B^{\mathrm{T}} p_k = f$

Calculate $r_k = Bu_k - \chi C p_k$

$\eta_k = \alpha r_k + \beta \eta_{k-1}$

$p_{k+1} = p_k + \eta_k$

end do

Here α and β are decided by

$$\alpha = \frac{1}{d}\frac{2}{1+\sqrt{1-(c/d)^2}} \tag{14}$$

$$\beta = \frac{2}{1+\sqrt{1-(c/d)^2}} - 1 \tag{15}$$

The constants d and c in the above formulations determine ellipse in the complex plane centred at d with foci at $d-c$ and $d+c$, which includes all the eigenvalues of $(BA^{-1}B^{\mathrm{T}} + \chi C)$. The constants d and c also satisfy the conditions $d>0$, and $d^2>c^2$. Such the determination of the constants, d and c, pledges convergence of the above algorithm. The optimal choice of d and c that yields the maximum rate of convergence should make the maximum eigenvalue σ_m of $(BA^{-1}B^{\mathrm{T}} + \chi C)$ lie on the ellipse produced by d and c. There are some schemes to determine the optimal values of d and c, for example, the procedures proposed by Manteuffel [5] and Hagemen and Young [6].

The second-order Richardson algorithm maintains the simplicity and robustness of the original Uzawa method. It requires almost no additional cost of computation, in terms of storage or CPU time. Numerical tests to the saddle point problems yielded from the Stokes equations which were carried out on the different grids showed that this algorithm speeds up convergence of Uzawa algorithm and provides more favourite convergence properties [4].

3.3. Pressure correction on conjugate gradient method

Theoretically, we may apply nearly all non-stationary Krylov methods to solve the discrete pressure equation (9). However, when choosing the Krylov iterative algorithms, we should carefully consider their properties. The solution of the saddle point system is only an inner iteration of the non-linear iteration. In practical applications, we often do not require that it is particularly accurate but the iterative processing had best have the smooth convergence behaviour. Based on this consideration, we opt for two non-stationary methods: conjugate gradient method (CG) and bi-conjugate gradient stabilized method (BiCGSTAB) iterative algorithms (such choices are based on our experience. One can also test other Krylov subspace methods. So it could not be said that CG and BiCGSTAB methods are the best of them). When the standard CG iteration is applied to the discrete pressure equation (9), one obtains

CG Algorithm

Given initial guess p_0 *and convergence criterion* ε

$Au_0 = f - Bp_0$

$r_0 = Bu_0 - \chi C p_0$

if $\|r_0\| \leqslant \varepsilon$ *stop*

$w_0 = r_0$

$Az_0 = Br_0$

for $n \geqslant 0$

$\rho_n = \dfrac{r_n^{\mathrm{T}} r_n}{r_n^{\mathrm{T}} B^{\mathrm{T}} z_n}$

$p_{n+1} = p_n - \rho_n w_n$

$u_{n+1} = u_n - \rho_n z_n$

$$r_{n+1} = r_n - \rho_n(B^{\mathrm{T}}z_n - \chi C w_n)$$
if $\|r_{n+1}\| \leqslant \varepsilon$ *stop*
$$\lambda_n = \frac{r_{n+1}^{\mathrm{T}} r_{n+1}}{r_n^{\mathrm{T}} r_n}$$
$$w_{n+1} = r_n + \lambda_n w_n$$
$$Az_{n+1} = Bw_{n+1}$$

The conjugate gradient algorithm is only fit to the case when the matrix A is symmetrical and positive definite. That is, in the discrete Navier–Stokes equations, we have to treat the convection terms explicitly. In next subsection we introduce the BiCGSTAB algorithm which is fit to more general case.

3.4. Pressure correction on BiCGSTAB

The BiCGSTAB iteration for the discrete pressure equation (9) may be derived from its standard implementation as follows:

BiCGSTAB Algorithm

Given initial guess p_0 *and convergence criterion* ε
$$Au_0 = f - Bp_0$$
$$r_0 = Bu_0 - \chi C p_0$$
if $\|r_0\| \leqslant \varepsilon$ *stop*
$$\tilde{r} = r_0$$
for $n \geqslant 1$
$$\rho_{n-1} = \tilde{r}^{\mathrm{T}} r_{n-1}$$
if $\rho_{n-1} = 0$ *method fails*
if $n = 1$
$$q_n = r_{n-1}$$
else
$$\beta_{n-1} = (\rho_{n-1}/\rho_{n-2})(\alpha_{n-1}/\omega_{n-1})$$
$$q_n = r_{n-1} + \beta_{n-1}(q_{n-1} - \omega_{n-1} v_{n-1})$$
end if
$$Az_n = -Bq_n$$
$$v_n = (B^{\mathrm{T}} z_n - \chi C q_n)$$
$$\alpha_n = \frac{\rho_{n-1}}{\tilde{r}^{\mathrm{T}} v_n}$$
$$s = r_{n-1} - \alpha_n v_n$$
if $\|s\| \leqslant \varepsilon$ *stop*
$$Ay_n = -Bs$$
$$t = (B^{\mathrm{T}} y_n - \chi C s)$$
$$\omega_n = \frac{t^{\mathrm{T}} s}{t^{\mathrm{T}} t}$$
$$p_n = p_{n-1} - \alpha_n q_n - \omega_n s$$
$$u_n = u_{n-1} - \alpha_n z_n - \omega_n y_n$$
$$r_n = s - \omega_n t$$
if $\|r_n\| \leqslant \varepsilon$ *stop*

The above BiCGSTAB algorithm, like Uzawa and second-order Richardson methods, does not require that the matrix A in the momentum equation (8) is symmetrical or positive definite. Therefore the matrix A may include the non-symmetrical convection matrix. But when used to Stokes problems, the BiCGSTAB method theoretically has the similar convergence properties with the conjugate gradient method. Moreover, note that the BiCGSTAB algorithm includes the twice pressure correction in every iteration, so it requires two times computational cost over the other ones.

3.5. Preconditioning techniques

All Krylov iterations for a linear system $Sx = f$ should satisfy the relation

$$x_m - x = P_m(S)(x_0 - x) \tag{16}$$

for some polynomial of degree m, with $P_m(0) = 1$. Different iterative algorithm produces its own matrix polynomial and so leads to different properties of convergence, although every algorithm tries to choose such polynomial that the error between iterative and exact solutions descends the most speedily in some sense. The polynomial corresponding to Uzawa iteration, for example, may be expressed in this form $P_m(x) = (1 - \gamma x)^m$. For the other iterations, the corresponding polynomials are more complex. According to relation (16), we have

$$||x_m - x|| = \max_{\lambda_k \in \rho(S)} |P_m(\lambda_k)| \, ||x_0 - x|| \tag{17}$$

Here $\lambda_1, \lambda_2, \ldots, \lambda_n$ are eigenvalues of the system matrix. Consequently, the convergence rates of iterative methods depend on the spectral distribution of the system matrix.

A preconditioner is a matrix which transforms system matrix into more favourable spectral properties. The transformed system has the same solution as the original equation, but the distribution of its spectral may be more favourable so as to gain a faster convergence rate. An excellent preconditioner should be a good approximate to original system matrix and would not bring much extra computing cost. In this work we tested two preconditioners to the discrete pressure equation (9) which are

$$M = (B\bar{A}^{-1}B^{\mathrm{T}} + C) \tag{18}$$

and

$$M = \mathrm{diag}(B\bar{A}^{-1}B^{\mathrm{T}} + C) \tag{19}$$

where the matrix $\bar{A}$ is close to the A matrix in some manner. A possible choice is the diagonal entries of the A matrix. In this work, however, we choose $\bar{A}_{ij} = \sum_j |A_{ij}|$. The preconditioner, $B\bar{A}^{-1}B^{\mathrm{T}}$, is a discrete approximation of the continuum Poisson operator on the finite element space. But it is not constructured directly from discretizing the continuum Poisson equation, rather by multiplying the two matrices B and B^{T}. In multiplication of B and B^{T}, we firstly multiply them in an element, and then assemble them into the global matrix.

The transformed system with the preconditioners, M, may be written as

$$M^{-1}(BA^{-1}B^{\mathrm{T}} + C)p = M^{-1}BA^{-1}f \tag{20}$$

Hence one can easily obtain the preconditioned algorithms mentioned in the preceding subsections. In particular, the preconditioned Uzawa algorithm is all the same as the pressure correction algorithm in SIMPLE-like method in the literature [7] which is written as follows:

Preconditioned Uzawa Algorithm
Give an initial approximation p_0 *of* p
for $k=1$ *until convergence, do*
Solve $Au_k = f - B^{\mathrm{T}} p_k$
Calculate $p_{k+1} = p_k + \gamma M^{-1}(Bu_k - Cp_k)$
end do

The numerical tests show that these preconditioners may to some extent improve the convergence rate of Uzawa iterations, but we did not find their significance to the other algorithms, considering the computational cost to construct them.

4. CONVERGENCE PERFORMANCE

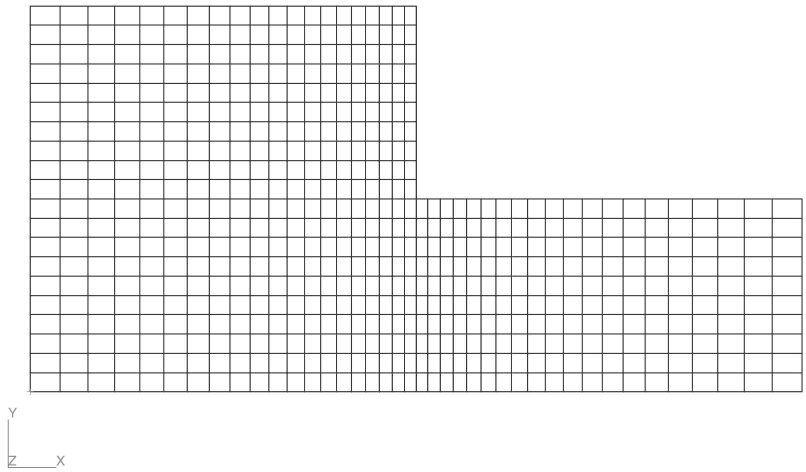
As we know, the convergence rates of the four methods depend on the spectral distribution of the system matrix $BA^{-1}B^{\mathrm{T}} + C$. Theoretically, Uzawa method gives the slowest convergence rate, then second-order Richardson algorithm. CG and BiCGSTAB methods should be the fastest methods. In this paper we do not attempt to mention the theoretical results too much, but present some typical numerical experiments. Besides the convergence rates, the numerical experiments still tell us the information on their convergence histories.

4.1. General comparison of four algorithms

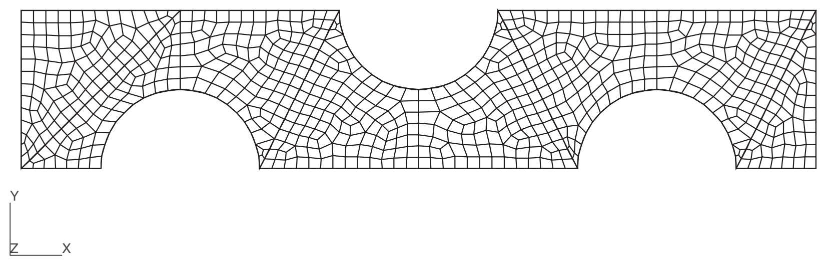
To investigate the effect of computational grids on the convergence properties of the algorithms, two kinds of grids are chosen: the regular and irregular. Each grid is generated into coarse and fine ones. The grid-density in the fine grid is some five times of the coarse. Figure 1 shows the two coarse grids, on which the upper and down boundaries are no-slip walls, and the left sides are inlets. The flow Reynolds numbers are 50 and 100, respectively, which is defined on the characteristic length of the inlet widths. Because CG method is only fit to Stoke problem, we actually remove the convection terms from Navier–Stokes equations when testing it. To Uzawa and second Richardson methods, although there are some procedures to decide the optimal parameters in them, they in the following tests are obtained by multiple computations.

Figures 2–5 show residual convergence processes of the discrete pressure equation for the four method, on which we may make the following observations.

Uzawa method presents the slowest convergence rate against the other three methods. Yet, its convergence behaviour is indeed smooth, and seems not to be much affected by mesh structure and density of elements. Second-order Richardson algorithm substantially improves the convergence performance of Uzawa method. As was expected, however, it cannot provide us with the same convergence rate as CG or BiCGSTAB method. Besides the convergence rate, it also has the other same advantages as Uzawa method.



(a)



(b)

Figure 1. Coarse computational grids: (a) regular grid; and (b) irregular grid.

CG method gives the best convergence behaviour of the four methods. This may have the benefit of removing the convection terms from Navier–Stokes equations, which leads to the matrix A in the momentum equation (8) symmetrical and positive definite. It appears to be somewhat unfair to compare it with the other three methods. But it is definite that it is an adequate algorithm for Stokes problems. The convergence rate of bi-conjugate gradient method is superior to Uzawa and second-order Richardson methods, and yet Figure 4 shows that its convergence performance seems to be much influenced by element size in regular mesh.

4.2. *Influence of mesh size*

In order to further investigate the influence of mesh sizes on the convergence performance, we calculate the two-dimensional flows around a square in a channel, in which the flow Reynolds number on the square width is 100. For this test, the computational meshes are quite regular. This is because the influence of mesh size on convergence property is the largest in regular mesh. We generated three meshes and they consist of about 10 000 elements, 5000 elements and 3000 elements, respectively. Figures 6–8 present the convergence histories of BiCGSTAB, second-order Richardson and Uzawa methods on these three grids. It may be found that the refinement of grid would lead to reduce the convergence rates to the three methods. But the

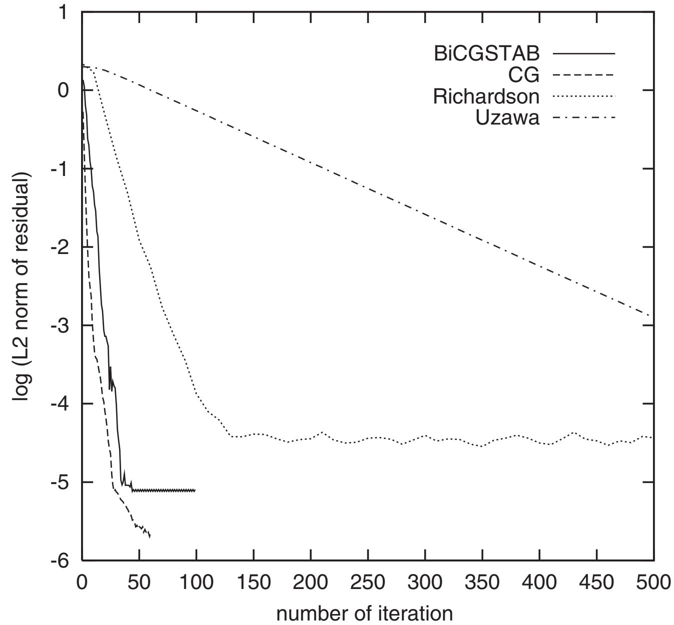


Figure 2. Convergence processes of velocity divergence residual for finer irregular mesh. Note: (1) When the residual reduces under 10e-5, stop the iteration in CG method tests. (2) The pressure correction after about iteration 50 in BiCGSTAB method is too small to be able to change the velocity field in machine accuracy. This case occurs also in the following tests.

influence to BiCGSTAB method is particularly marked. And also, more regular and finer grid makes the convergence process of BiCGSTAB method more irregular. It is fatal weakness of BiCGSTAB method to solve the non-linear system produced by Navier–Stokes problem.

4.3. Coefficients in Uzawa and second-order Richardson methods

The optimal coefficients γ in Uzawa algorithm and α as well as β in second-order Richardson method are decided by the eigenvalue distribution of the system matrix $BA^{-1}B^{\mathrm{T}}+C$. Though there are some procedures to determine these coefficients when system matrix is explicitly given, they are still quite difficult to be applied to the discrete pressure equation with the implicitly given system matrix. A practical method is to decide them through some test computations. Fortunately, Uzawa and second-order Richardson methods are not sensitive when these coefficients vary around their optimal values. This may be confirmed by Figure 9.

The coefficient α in second-order Richardson method, in general, is set as same as the coefficient γ in Uzawa. According to formulation (15), the coefficient β has the values from zero to one. As usual, the more it is, the faster the convergence rate. But too largely setting it could lead that the residual cannot descend to lower level. Figure 10 is typical example that the coefficient β affects the convergence behaviour.

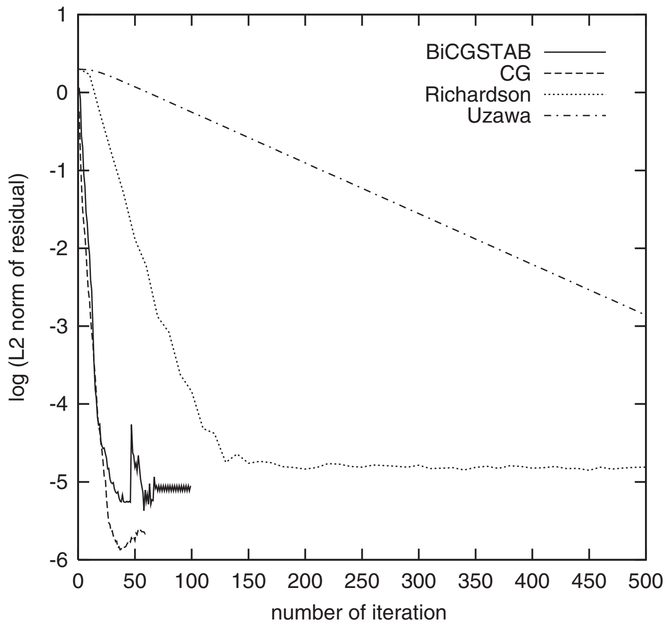


Figure 3. Convergence processes of velocity divergence residual for coarse irregular mesh.

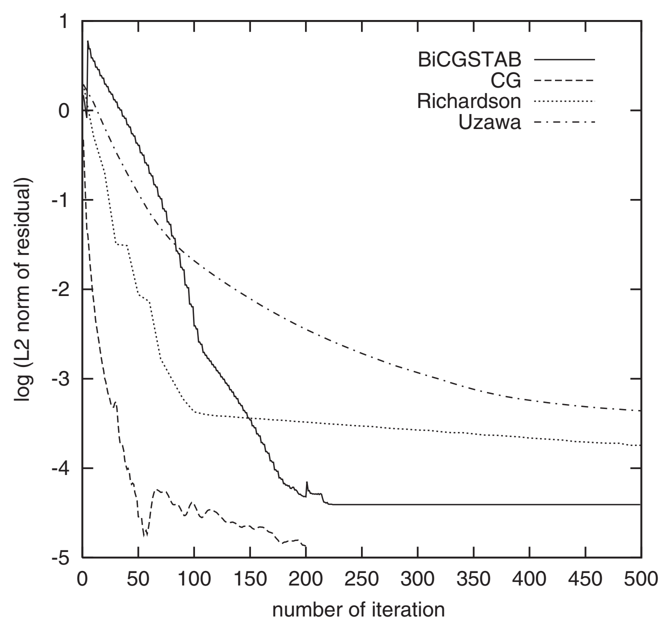


Figure 4. Convergence processes of velocity divergence residual for finer regular mesh.

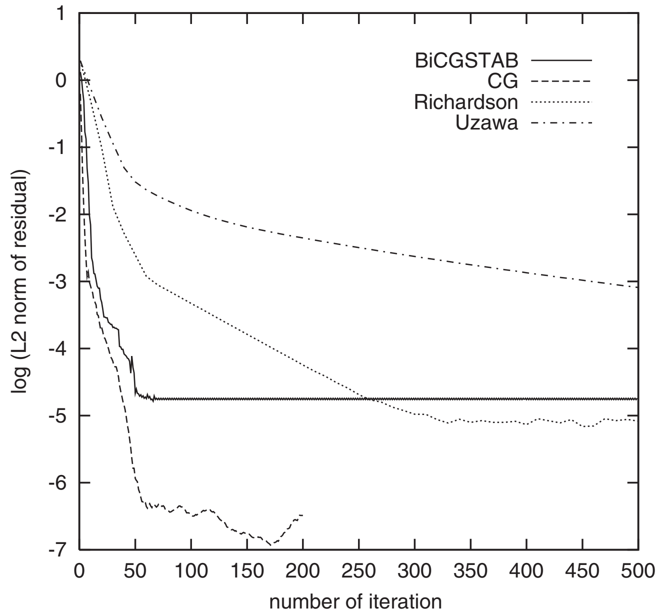


Figure 5. Convergence processes of velocity divergence residual for coarse regular mesh.

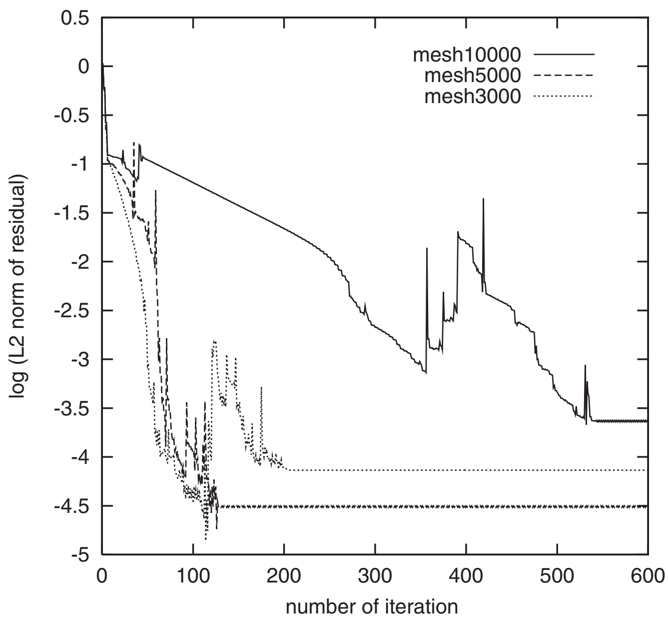


Figure 6. Influence of mesh size to BiCGSTAB method's convergence behaviour.

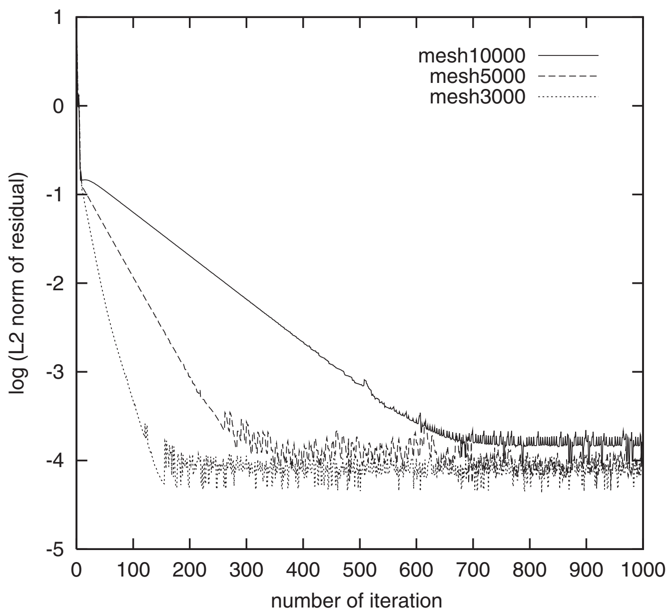


Figure 7. Influence of mesh size to second-order Richardson method's convergence behaviour.

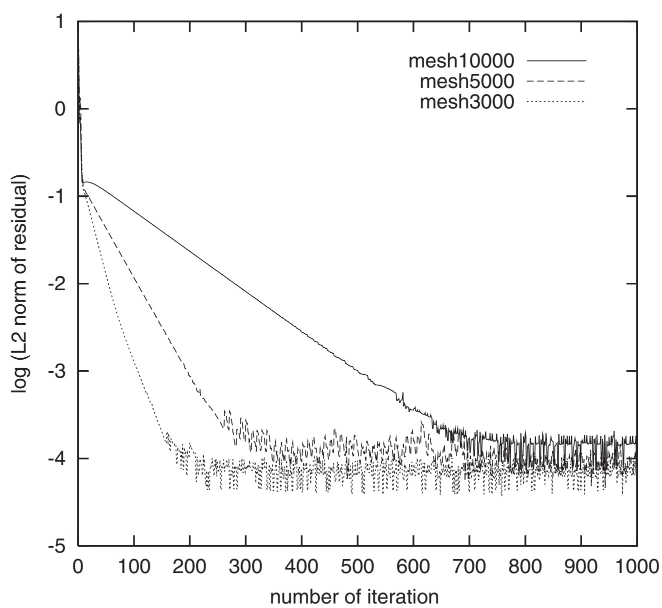


Figure 8. Influence of mesh size to Uzawa method's convergence behaviour.

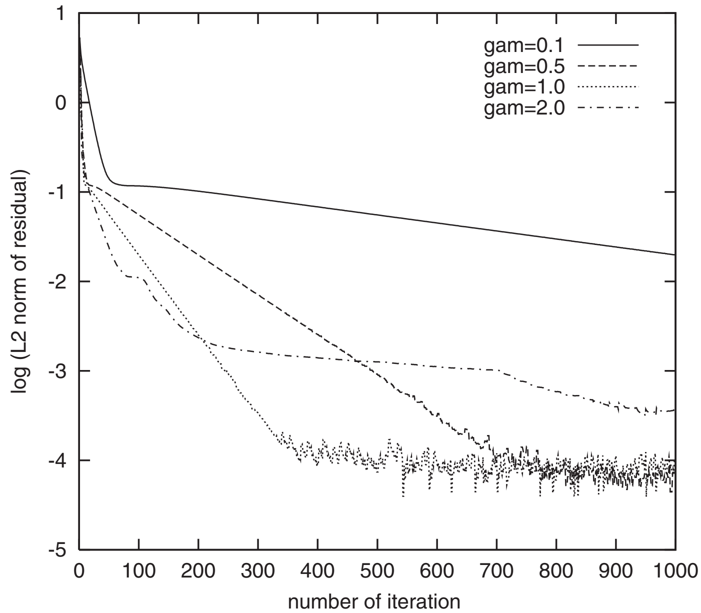


Figure 9. Influence of parameter γ to Uzawa method's convergence behaviour.

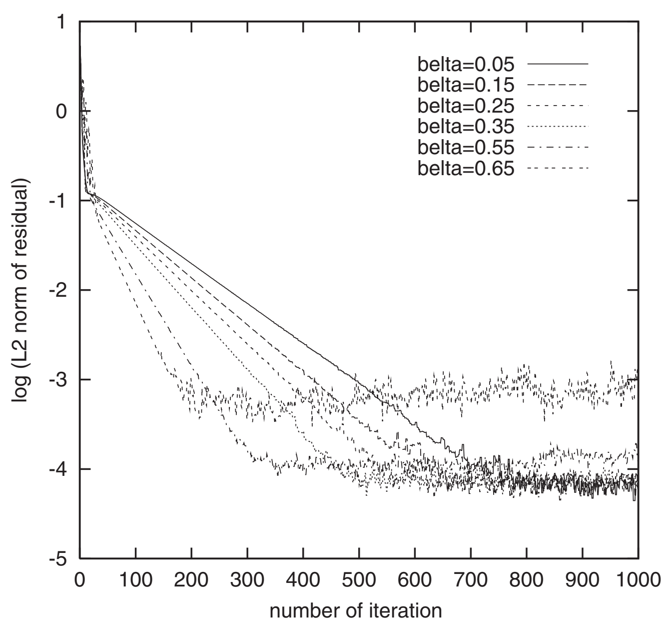


Figure 10. Influence of parameter β to second-order Richardson method's convergence behaviour.

4.4. Remark on computing cost

Now let us take a look at the computing cost of the four algorithms. Before referring to it, let us particularly note the numerical solutions of the momentum equations, $Au_k = f - B^{\mathrm{T}} p_k$ and $Az_k = g$ in the above iterations. In fact they are the major cost during iterations, because in each pressure correction we have to exactly solve these momentum equations. Based on the solution of the momentum equation, we do correction of the pressure for the next iteration. Actually, the difference among the four methods is only in determining the coefficients with which to correct the pressure. And also, the pressure correction only needs very little computing operation. As usual, all the processes of the pressure correction occupy less than one percent of solving the momentum equations. Therefore, the iteration numbers actually represent the computational cost when taking the same computing time for solving the momentum equations.

However, we would still like to complement some comments to the computing cost of the four algorithms. As is known, the residual is obtained not directly with multiplying the system matrix by the solution vector, but using formulation (10). Thus the error to solve the momentum equation inevitably transfers to the residual. To non-stationary Krylov methods, the coefficients are decided by the residuals of iterations. A large error of the residual may lead to wrong estimate them, and so damages numerical stability. In order to avoid this case, the solutions of the momentum equation (8) have to be exact. To do so is quite expensive in some cases. Another factor that affects the computing cost of solution of the momentum equations is the rough convergence behaviour of BiCGSTAB method. The rough convergence processes mean large variation of the corrected pressure. When solving the momentum equations by iterative methods, the previous solution is often taken as an initial iteration value for next iteration. Clearly, the smaller the pressure correction, the less the iteration for solving momentum equations. The stationary iterations, such as, Uzawa or second-order Richardson algorithms, however, are not subjected to this restriction, and their convergence behaviours are always quite smooth as long as the constants of the algorithms are taken appropriately. Therefore they save computing time from solving the momentum equation. Such robust numerical stability is particularly helpful to the ill-conditioned system matrix.

5. APPLICATION

In this section, we apply the three pressure correction methods mentioned in Section 3 to compute the three-dimension lid-driven cavity flows: Uzawa, second-order Richardson and BiCGSTAB methods. The global iterative process is described in Section 2. To choose the lid-driven cavity flows is because it was recently calculated by Sheu and Tsai [8] and may provide us with a comparable data quantitatively. In Sheu and Tsai's work, the geometry of the cavity has a depth-to-width aspect ratio of 1:1 and a span-to-width aspect ratio of 1:1. Their computations at Reynolds number $Re = 400$ are based on two meshes which have the resolutions of 41^3 and 51^3 nodes in three co-ordinate directions, respectively, and distribute the nodes in non-uniform along the edges of the cavity. In our computations, the coarse of the above meshes is utilized and the flow Reynolds number is the same as one in Sheu and Tsai's work. When the residual of the pressure equation (10) and the variation of the iterative flow velocities are less than 10^{-4}, we think the convergent solution is reached and terminate the

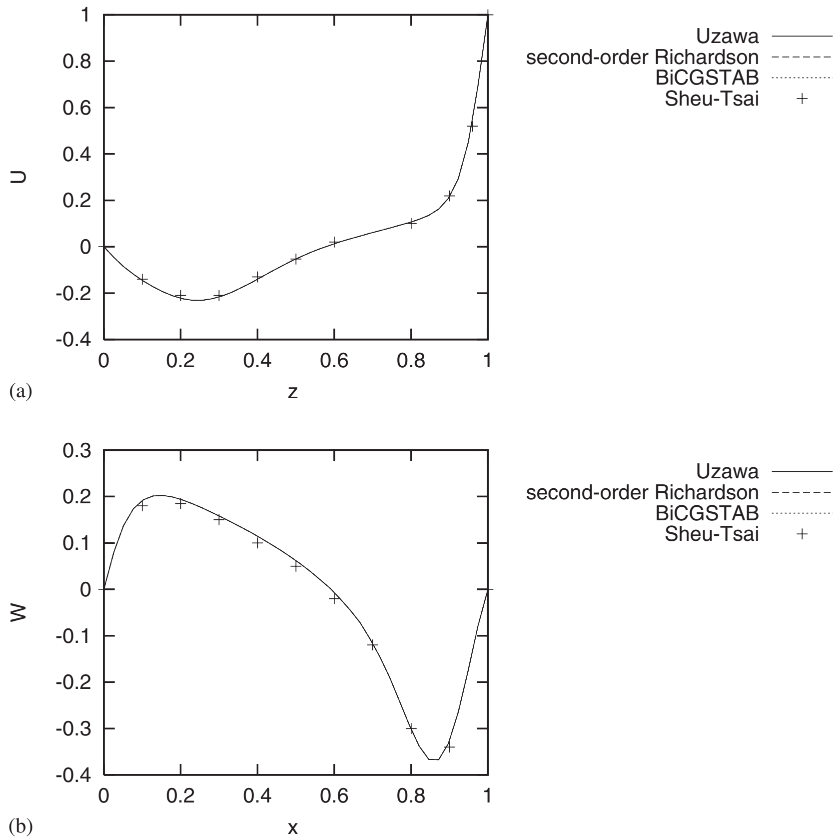


Figure 11. Comparison of U and W profiles at centre-lines of symmetry.

computation. The convergent solutions obtained by the three methods, actually, are identical, and they are compared with the Sheu and Tsai's computations in Figure 11, which shows the flow velocity profiles along the vertical and horizontal centrelines on the mid-plane of the cavity. Note that the velocity profiles of the three methods in Figure 11 are overlapped together means the solutions from the three methods are identical and they also agree with the computations in Sheu and Tsai' work.

In order to further compare our computations with Sheu and Tsai's work, we plot the limiting streamlines on five solid walls on Figure 12. They are all agreeable with the Figure 4 in Sheu and Tsai's paper.

Now we give the details of the computations as follow (Table I): The performance of the global iteration depends not only on the solution method of the saddle point problem, but also on the linearization procedure and other factors. In general, a faster solver for the saddle point problem would lead to better performance of the global iteration. Nevertheless, this argument is not always correct to a solver for non-linear system of equation. Sometime, a more accurate solution of the linear system in the inner loop could produce too large change to the preceding iteration of the non-linear system in the outer loop, and bring the negative

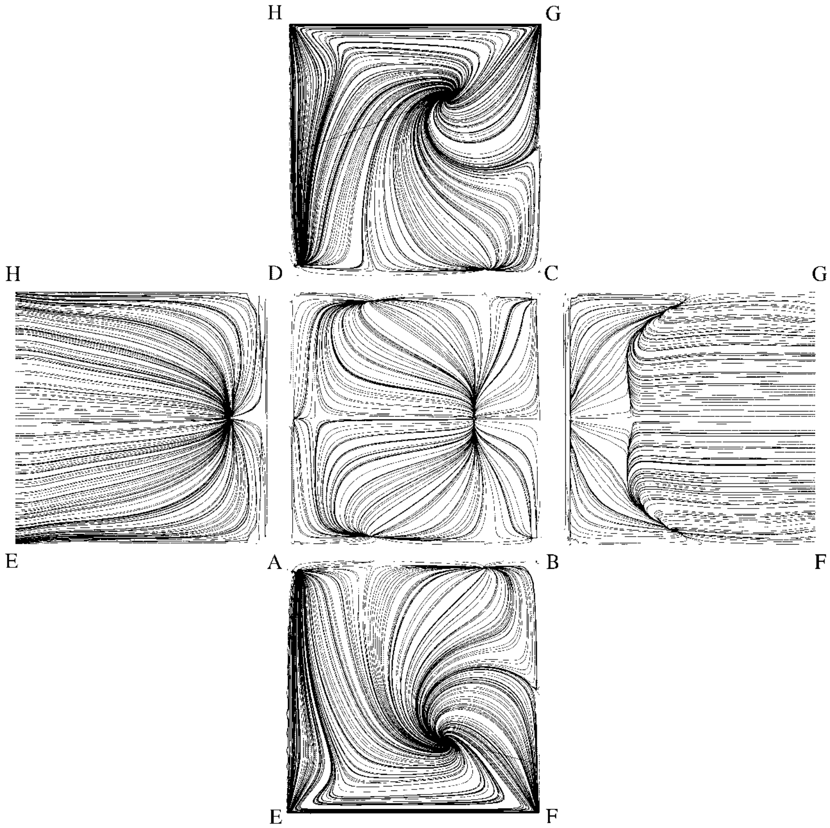


Figure 12. Limiting streamlines on five solid walls (the tags are identical with Sheu and Tsai's paper).

Table I.

Methods	Iteration number	Computing time (s)
Uzawa ($\gamma = 0.02$)	120	6983
Second-order Richardson ($\alpha = 0.02, \beta = 0.25$)	109	6549
BiCGSTAB	68	6102

effect to the global iterative process. Hence it is significant work to study what scheme to solve the non-linear system to match the algorithm for the saddle point system mentioned in Section 3, so as to yield optimal combination. However such the topic is beyond the scale of this paper. The numerical results presented in this section are just to show a successful example to use Krylov method in incompressible Navier–Stokes solver. It could not be the most optimal combination. We still add some comments on BiCGSTAB method. As mentioned in Section 4.4, in practical computations, BiCGSTAB method requires more accurate solutions of the momentum equations in each pressure correction. Consequently every iteration of BiCGSTAB method actually needs more computational time.

The above numerical test appears to suggest that BiCGSTAB method is the best one. In fact, fairly comparing the pressure correction methods in this paper is not simple. It is subject to the mesh sizes, Reynolds numbers and tested flows. Especially, for high Reynolds number flows, solving methods of the non-linear system become important. The more irregular convergence performance of BiCGSTAB algorithm may lead to poor convergence behaviour of non-linear iterations. Therefore, we do not try to recommend readers to abandon Uzawa method and just adopt BiCGSTAB method. But when Uzawa method works badly, second-order Richardson and BiCGSTAB methods may be an option. Particularly note that there are two parameters in second-order Richardson iteration. Yet, Uzawa iteration has only one parameter. In practical applications of second-order Richardson iteration, the first parameter may be taken as the same as the one of Uzawa iteration. The second parameter, due to be quantified in zero to one, is easily be determined. The convergence performance of second-order Richardson iteration is always better than Uzawa algorithm, when the coefficients in the algorithms given appropriately.

6. CONCLUSIONS

In this paper we present a new approach to construct the pressure correction method for solving incompressible Navier–Stokes problems, based on Krylov subspace concept. As examples, we study four kinds of the pressure correction methods under this framework: Uzawa, second-order Richardson, CG and BiCGSTAB methods. Their convergence properties and behaviour are investigated by the numerical experiments. Certainly the pressure correction methods based on Krylov subspace iteration should not be confined in the four methods. From the point of view of convergence rates, the non-stationary iterative methods, CG and BiCGSTAB, are superior to Uzawa and second-order Richardson methods. However, the convergent behaviour of BiCGSTAB iterations in some cases becomes irregular. It is particularly useless for BiCGSTAB method to be used to solve the non-linear system of discrete Navier–Stokes problems. Preconditioning concept is a bridge to link the methods mentioned in this paper and the traditional pressure correction methods with solving pressure Poisson equation. However, we still need further work to determine how to combine these two types of the methods, so as to gain more rapidity of convergence rates.

REFERENCES

1. Silvester D, Kechkar N. Stabilized bilinear-constant velocity-pressure finite elements for the conjugate gradient solution of the Stokes problem. *Computer Methods in Applied Mechanics and Engineering* 1990; **79**:71–86.
2. Franca LP, Frey SL. Stabilized finite element methods: II the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1992; **99**:209–233.
3. Hughes TJR, Franca LP. A new finite element formulation for CFD: VII. The Stokes problem with various well-posed boundary conditions: symmetrical formulation that converge for all velocity/pressure space. *Computer Methods in Applied Mechanics and Engineering* 1987; **65**:85–96.
4. Liu W, Xu S. A new improved Uzawa method for finite element solution of Stokes problem. *Computational Mechanics* 2001; **27**:305–310.
5. Manteuffel TA. Adaptive procedure for estimating parameters for the non-symmetric Tchebychev iteration. *Numeriche Mathematik* 1978; **31**:183–208.
6. Hagemen LA, Young DM. *Applied Iterative Methods*. Academic Press: New York, London, 1981.
7. Haroutunian V, Engelman MS, Hasbani I. Segregated finite element algorithms for the numerical solution of large-scale incompressible flow problems. *International Journal for Numerical Methods in Fluids* 1993; **17**:323–348.

8. Sheu T, Tsai S. Flow topology in a steady three-dimensional lid-driven cavity. *Computers and Fluids* 2002; **31**:911–934.
9. Arrow KJ, Hurwicz L, Uzawa H. *Studies in Non-linear Programming*. Stanford University Press, Stanford: 1958.
10. Bank RE, Welfert BD, Yserentant H. A class of iterative methods for saddle point problems. *Numeriche Mathematik* 1990; **56**:645–666.
11. Barrett R *et al*. Templates for the solution of linear systems: building blocks for iterative methods. www.netkib.org/templates, 1994.
12. Benim AC, Zinser W. A segregated formulation of Navier–Stokes equations with finite elements. *Computer Methods in Applied Mechanics and Engineering* 1986; **57**:223–237.
13. Brezzi F, Fortin M. *Mixed and Hybrid Finite Element Methods*. Springer: Berlin, 1991.
14. Cahouet J, Chabard JP. Some fast 3D finite element solvers for the generalized Stokes problem. *International Journal for Numerical Methods in Fluids* 1988; **8**:869–895.
15. Elman HC. Multigrid and Krylov subspace methods for the discrete Stokes equations. *International Journal for Numerical Methods in Fluids* 1996; **22**:755–770.
16. Elman HC, Golub GH. Inexact and preconditioned Uzawa algorithms for saddle point problems. *SIAM Journal on Numerical Analysis* 1994; **31**:1645–1661.
17. Elman H, Silvester D. Fast nonsymmetric iterations and preconditioning for Navier–Stokes equations. *SIAM Journal on Scientific Computing* 1996; **17**:33–46.
18. Fortin M, Glowinski R. *Augmented Lagrangial Methods*. North-Holland: Amsterdam, 1983.
19. Fortin M, Boivin S. Iterative stabilization of the bilinear velocity-constant pressure element. *International Journal for Numerical Methods in Fluids* 1990; **10**:125–140.
20. Glowinski R, Pironneau O. Finite element methods for Navier–Stokes equations. *Annual Review of Fluid Mechanics* 1992; **24**:167–204.
21. Golub GH, Wathen A. An iteration for indefinite systems and its application to the Navier–Stokes Equations. *SIAM Journal on Scientific Computing* 1998; **19**:530–539.
22. Humphrey JAC, Iacovides H, Launder BE. Some numerical experiments on developing laminar flow in circular-sectioned bends. *Journal of Fluid Mechanics* 1985; **154**:357–375.
23. Manteuffel TA. The Tchebychev iteration for non-symmetric linear systems. *Numeriche Mathematik* 1977; **28**:307–327.
24. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere: Washington, DC, 1980.
25. Saad Y. Krylov subspace methods for solving large unsymmetrical linear system. *Mathematical Computation* 1981; **37**:105–126.
26. Shaw CT. Using a segregated finite element scheme to solve the incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Fluids* 1991; **12**:81–92.
27. Silvester D, Wathen A. Fast iterative solution of stabilized Stokes systems, part two: using general block preconditioners. *SIAM Journal on Numerical Analysis* 1994; **31**:1352–1367.
28. Varga RS. *Matrix Iterative Analysis*. Prentice-Hall: Englewood Cliffs, NJ, 1962.
29. Verfuerth R. A combined conjugate gradient-multigrid algorithm for the numerical solution of the Stokes problem. *IMA Journal on Numerical Analysis* 1984; **4**:441–455.
30. Wathen A, Silvester D. Fast iterative solution of stabilized Stokes systems, part one: using simple diagonal preconditioners. *SIAM Journal on Numerical Analysis*1993; **30**:630–649.
31. Wittum G. Multi-grid methods for Stokes and Navier–Stokes equations. *Numeriche Mathematik* 1989; **55**:543–563.
32. Zhou RQN. Preconditioned finite element algorithms for 3D Stokes flows. *International Journal for Numerical Methods in Fluids* 1993; **17**:667–685.
33. Zienkiewicz OC, Vilotte JP, Toyoshima S, Nakazawa S. Iterative method for constrained and mixed approximation, an inexpensive improvement of F.E.M. performance. *Computer Methods in Applied Mechanics and Engineering* 1985; **51**:3–29.