INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 14, 71–81 (1992)

ITERATIVE METHODS FOR STABILIZED MIXED VELOCITY–PRESSURE FINITE ELEMENTS

JOHN ATANGA AND DAVID SILVESTER

Department of Mathematics, University of Manchester Institute of Science and Technology,
PO Box 88, Manchester M60 1QD, U.K.

SUMMARY

This paper is concerned with iterative techniques for the solution of the linear system of equations arising from a finite element approximation of an elliptic partial differential equation by a mixed method. Three types of iterative algorithms are investigated. Applications to the Stokes equations are discussed and the results of numerical experiments given.

KEY WORDS Stabilized finite element Stokes equation

1. INTRODUCTION

Direct solution methods for large sparse systems, especially those arising from finite element discretizations of three-dimensional problems, can exhibit high computational costs. The development of iterative strategies to reduce these costs has been an active research area for many years (see e.g. References 1–3) and the area remains a vigorous branch of numerical analysis.

The discretized system corresponding to a conventional mixed finite element formulation can be written as the partitioned matrix equation (formally described in Reference 4)

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{pmatrix}\begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix}=\begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}. \tag{1}$$

Here we will restrict our attention to the Stokes equations (see e.g. Reference 5), in which case $\mathbf{A}$ is a symmetric positive definite $N \times N$ matrix, $\mathbf{B}$ is $M \times N$ matrix corresponding to the coupling term and is of full rank if a stable mixed method is used,[4] $\mathbf{0}$ is the $M \times M$ zero matrix, the vectors $\mathbf{f}$ and $\mathbf{g}$ result from inhomogeneous data, and $\mathbf{u}$ and $\mathbf{p}$ denote vectors of length N and M corresponding to nodal velocity and pressure values respectively. Of crucial interest here is the zero block on the diagonal, which is a source of indefiniteness and which makes iterative solution of the system (1) difficult.

Low-order conforming mixed approximations of incompressible flows, such as those obtained with the bilinear velocity–constant pressure element (Q_1–P_0), are well known to be unstable in pressure while providing reasonable results for the velocity. On the other hand, computer implementation aspects curtail the use of higher-order approximations. These observations have led to the introduction of stable low-order mixed finite element methods.

0271–2091/92/010071–11$05.50

Received September 1990
Revised March 1991

A recent development of mixed finite element discretizations of the Stokes (and Navier–Stokes) equations with low-order methods has been the use of regularization, which enhances the possibility of using iterative solution methods for these problems.[6–9] Here we will consider a symmetric regularization of the discrete Stokes problem[9]

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & -\beta\mathbf{C} \end{pmatrix}\begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix}=\begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}, \tag{2}$$

where $\mathbf{C}$ is a symmetric positive semidefinite matrix such that the 'stabilization condition'

$$\mathbf{B}^{\mathrm{T}}\mathbf{p}=0 \quad \Rightarrow \quad \mathbf{p}^{\mathrm{T}}\mathbf{C}\mathbf{p}\neq 0 \tag{3}$$

is satisfied and $\beta>0$ is the stabilization parameter.

We can justify the use of the term 'regularization' by considering the eigenvalue distribution of the systems (1) and (2). In particular, constructing the congruence transformation

$$\begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & -\beta\mathbf{C} \end{pmatrix}=\begin{pmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{B} & \mathbf{I} \end{pmatrix}\begin{pmatrix} \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & -\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}-\beta\mathbf{C} \end{pmatrix}\begin{pmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{0} & \mathbf{I} \end{pmatrix}$$

and applying Sylvester's law of inertia, it is clear that the coefficient matrix in (2) is always non-singular, whereas (1) will have zero eigenvalues whenever the matrix $\mathbf{B}$ is rank-deficient (giving rise to instability associated with spurious pressure modes). In simple terms, both (1) and (2) have N strictly positive eigenvalues, but (1) can have zero eigenvalues which are transformed into strictly negative eigenvalues by the stabilization condition (3).

Our objective in this work is to present a comparative survey of the three general strategies for solving the stabilized system (2). In Section 2 we derive an iterative scheme for the so-called primal problem (i.e. for the velocity field). Our technique is a variant of the well-known Uzawa method (see e.g. Reference 10) applied to (2). Section 3 discusses a standard reformulation of (2) as a symmetric positive definite system for the dual variable (i.e. the pressure field). The basic conjugate gradient method is then applied to this reformulated system, as proposed by Verfürth[11] and developed by Bramble and Pasciak[12] and also Bank *et al.*[13] We show that both the primal and dual strategies implicitly involve a decoupling of the Stokes system into the component equations of momentum conservation and incompressibility. This decoupling implies that the resulting iterative schemes have an embedded inner–outer iteration structure. In Section 4 we consider applying a preconditioned conjugate gradient method directly to the stabilized system (2), i.e. we investigate the feasibility of using a much simpler (single-level) iteration for both primal and dual variables simultaneously. The numerical performance of the three techniques is then assessed on a test problem in Section 5 and some conclusions are drawn. Finally, in Section 6 the impact of this work within the more general framework of Navier–Stokes flow solvers is assessed.

2. AN ITERATED PENALTY ALGORITHM

A discrete penalty formulation corresponding to the system (2) can be written in matrix form as

$$\mathbf{A}\mathbf{u}+\mathbf{B}^{\mathrm{T}}\mathbf{p}=\mathbf{f}, \tag{4}$$

$$\mathbf{B}\mathbf{u}-(\varepsilon\mathbf{M}+\beta\mathbf{C})\mathbf{p}=\mathbf{g}, \tag{5}$$

where $\mathbf{M}$ is the pressure mass matrix and ε is a small penalty parameter. The matrix system (4), (5) is a perturbation of (2); thus the solution of (4), (5) corresponds to that of a neighbouring problem and not to that of (2). A simple way of preserving consistency is to perform the iteration

$$\mathbf{A}\mathbf{u}^{i+1}+\mathbf{B}^{\mathrm{T}}\mathbf{p}^{i+1}=\mathbf{f}, \tag{6}$$

$$\mathbf{B}\mathbf{u}^{i+1}-(\varepsilon\mathbf{M}+\beta\mathbf{C})\mathbf{p}^{i+1}=-\varepsilon\mathbf{M}\mathbf{p}^{i}+\mathbf{g}, \tag{7}$$

for $i=1, 2, \ldots$, with $\mathbf{p}^0=0$.

Here, since $\mathbf{M}$ is symmetric positive definite and $\mathbf{C}$ is symmetric positive semidefinite, the matrix $\varepsilon\mathbf{M}+\beta\mathbf{C}$ is symmetric positive definite (hence non-singular), so $\mathbf{p}^{i+1}$ can be eliminated from the system:

$$\left(\mathbf{A}+\frac{1}{\varepsilon}\mathbf{B}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{B}\right)\mathbf{u}^{i+1}=\mathbf{f}+\frac{1}{\varepsilon}\mathbf{B}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{g}-\mathbf{B}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{M}\mathbf{p}^{i}, \tag{8}$$

with

$$\mathbf{p}^{i+1}=\mathbf{D}^{-1}\left(\mathbf{M}\mathbf{p}^{i}+\frac{1}{\varepsilon}(\mathbf{B}\mathbf{u}^{i+1}-\mathbf{g})\right), \tag{9}$$

where $\mathbf{D}=\mathbf{M}+(\beta/\varepsilon)\mathbf{C}$ and $(1/\varepsilon)\mathbf{B}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{B}$ is the 'stabilized penalty matrix'. We now prove the convergence of the iteration scheme defined by (6), (7).

Theorem 1

For all $\mathbf{p}^0\in\mathbb{R}^M$ and $\varepsilon>0$ the sequence $(\mathbf{u}^{i+1}, \mathbf{p}^{i+1})$ defined by (6), (7) converges to the solution of (2).

Proof. Suppose $(\mathbf{u}, \mathbf{p})$ is a solution of (2); then

$$\mathbf{A}\mathbf{u}+\mathbf{B}^{\mathrm{T}}\mathbf{p}=\mathbf{f}, \tag{10}$$

$$\mathbf{B}\mathbf{u}-\beta\mathbf{C}\mathbf{p}-\varepsilon\mathbf{M}\mathbf{p}=\mathbf{g}-\varepsilon\mathbf{M}\mathbf{p}. \tag{11}$$

We introduce $\bar{\mathbf{u}}^i=\mathbf{u}^i-\mathbf{u}$ and $\bar{\mathbf{p}}^i=\mathbf{M}^{1/2}(\mathbf{p}^i-\mathbf{p})$. Subtracting (10) from (6) and (11) from (7), we obtain

$$\mathbf{A}\bar{\mathbf{u}}^{i+1}+\mathbf{B}^{\mathrm{T}}\mathbf{M}^{-1/2}\bar{\mathbf{p}}^{i+1}=0, \tag{12}$$

$$\mathbf{B}\bar{\mathbf{u}}^{i+1}-\beta\mathbf{C}\mathbf{M}^{-1/2}\bar{\mathbf{p}}^{i+1}-\varepsilon\mathbf{M}^{1/2}\bar{\mathbf{p}}^{i+1}=-\varepsilon\mathbf{M}^{1/2}\bar{\mathbf{p}}^{i}. \tag{13}$$

Substituting for $\bar{\mathbf{u}}^{i+1}$ from (12) into (13) gives

$$\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}\mathbf{M}^{-1/2}\bar{\mathbf{p}}^{i+1}+\beta\mathbf{C}\mathbf{M}^{-1/2}\bar{\mathbf{p}}^{i+1}+\varepsilon\mathbf{M}^{1/2}\bar{\mathbf{p}}^{i+1}=\varepsilon\mathbf{M}^{1/2}\bar{\mathbf{p}}^{i}, \tag{14}$$

leading to the iteration for the pressure error

$$\left(\frac{1}{\varepsilon}\mathbf{M}^{-1/2}(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}+\beta\mathbf{C})\mathbf{M}^{-1/2}+\mathbf{I}\right)\bar{\mathbf{p}}^{i+1}=\bar{\mathbf{p}}^{i}. \tag{15}$$

Now, because of the stabilization condition (3), the matrix $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}+\beta\mathbf{C}$ is positive definite and symmetric, hence the matrix $\mathbf{M}^{-1/2}(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}+\beta\mathbf{C})\mathbf{M}^{-1/2}$ is positive definite and symmetric and so must have strictly positive eigenvalues. Writing the iteration in the form

$$\bar{\mathbf{p}}^{i+1}=\mathbf{K}\bar{\mathbf{p}}^{i}, \tag{16}$$

it is clear that the convergence behaviour of (15) is determined by the spectral radius of $\mathbf{K}$, which is given by

$$\rho(\mathbf{K})=\frac{1}{(1/\varepsilon)\lambda^{*}+1}, \tag{17}$$

where λ^* is the minimum eigenvalue of the matrix $\mathbf{M}^{-1/2}(\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}+\beta\mathbf{C})\mathbf{M}^{-1/2}$. Hence, since $\varepsilon>0$, (16) will always converge, i.e. $\bar{\mathbf{p}}^i\rightarrow 0$ as $i\rightarrow\infty$. Since $\mathbf{M}^{1/2}$ is non-singular, this implies that $\mathbf{p}^i\rightarrow\mathbf{p}$. Further, from (12), $\bar{\mathbf{p}}^{i+1}=0$ implies $\bar{\mathbf{u}}^{i+1}=0$, so we also have that $\mathbf{u}^i\rightarrow\mathbf{u}$ as $i\rightarrow\infty$. □

Concerning the sensitivity of the scheme with respect to the penalty parameter ε, we have that for ε sufficiently small the 'contraction factor' $\rho(\mathbf{K})$ is of the form

$$\rho(\mathbf{K})=\frac{\varepsilon}{\lambda^*}\left(1+\frac{\varepsilon}{\lambda^*}\right)^{-1}\approx\frac{\varepsilon}{\lambda^*}$$

so that ε behaves like an acceleration parameter: the smaller the value of ε, the faster is the convergence of the algorithm. The weak point of the algorithm is obviously the fact that an 'inner system' (8) has to be solved at each iteration. Unfortunately, the coefficient matrix $\mathbf{A}+(1/\varepsilon)\mathbf{B}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{B}$ becomes progressively more ill-conditioned as ε decreases, making iterative solution of the 'inner system' progressively more difficult. See Reference 5 for a fuller discussion of this issue.

Concerning the sensitivity of the scheme with respect to the stabilization parameter β for a fixed ε, our experiments reported later show that if β is in the range $1-10^2$ there is no significant change in the behaviour of the iteration. However, as $\beta\rightarrow 0$ the number of iterations required increases (because $\lambda^*\rightarrow 0$ as $\beta\rightarrow 0$).

Our implementation of the iterated penalty method is as defined in Algorithm One. Concentrating on the convergence of the inner iteration (i.e. the calculation of $\mathbf{u}^{i+1}$), the condition number of the coefficient matrix $\mathbf{A}+(1/\varepsilon)\mathbf{B}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{B}$ can be shown to be $O(1/\varepsilon)$ and so the inner system becomes progressively ill-conditioned as the 'acceleration parameter' is reduced. However, the matrix is always symmetric positive definite, so for a moderate choice ($\varepsilon=10^{-4}$, say) the conjugate gradient method is a realistic choice for the inner solution procedure.

Clearly, the convergence of the outer iteration depends on the convergence of the inner iteration at each step. The two obvious strategies are having a fixed tolerance for the inner iteration or performing a fixed number of inner iterations. We will adopt the first approach,

```
set $p^0 = 0$
solve
$(A + \frac{1}{\epsilon}B^t D^{-1} B)u^1 = f - \frac{1}{\epsilon}B^t D^{-1} g$
$p^1 = \frac{1}{\epsilon}D^{-1}(Bu^1 - g)$
for $i = 1,2....$
    solve
    $(A + \frac{1}{\epsilon}B^t D^{-1} B)u^{i+1} = f + \frac{1}{\epsilon}B^t D^{-1} g - B^t D^{-1} M p^i$
    update the pressure
    $p^{i+1} = D^{-1}(Mp^i + \frac{1}{\epsilon}(Bu^{i+1} - g))$
    update the residual
    $r^{i+1} = g + \beta C p^{i+1} - Bu^{i+1}$
endfor
```

Algorithm One

converging both levels of iteration to the same accuracy using the current outer iterate as the initial guess for the inner iteration at each stage.

In the next section we discuss a method which does not involve a penalty parameter.

3. A TWO-LEVEL CONJUGATE GRADIENT ALGORITHM

Applying block Gaussian elimination to (2) and multiplying the last block row by -1, we obtain

$$\begin{pmatrix} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}} \\ \mathbf{0} & \beta\mathbf{C}+\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}} \end{pmatrix}\begin{pmatrix} \mathbf{u} \\ \mathbf{p} \end{pmatrix}=\begin{pmatrix} \mathbf{A}^{-1}\mathbf{f} \\ \mathbf{B}\mathbf{A}^{-1}\mathbf{f}-\mathbf{g} \end{pmatrix}. \tag{18}$$

The stabilized pressure equation can then be written as

$$(\beta\mathbf{C}+\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}})\mathbf{p}=\mathbf{B}\mathbf{A}^{-1}\mathbf{f}-\mathbf{g} \tag{19}$$

and the velocity is recovered via

$$\mathbf{A}\mathbf{u}=\mathbf{f}-\mathbf{B}^{\mathrm{T}}\mathbf{p}. \tag{20}$$

The pressure matrix $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}$ is symmetric and for quasi-uniform grids and using a stable mixed method has a condition number which is independent of the grid parameter h.[11] In our case the stabilized pressure matrix $\beta\mathbf{C}+\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}$ is also symmetric and furthermore the stability condition (3) ensures that it is positive definite. Numerical experiments show that it is also 'perfectly' conditioned in the sense that for a given β the condition number of the pressure matrix tends to a constant as the grid is uniformly refined. Consequently, we can solve (19) efficiently using the basic conjugate gradient method. This leads to Algorithm Two.

Considering the convergence of Algorithm Two, the computation of the step length is crucial. This is associated with the inner linear system

$$\mathbf{A}\mathbf{s}_{\mathrm{v}}^{i}=\mathbf{B}^{\mathrm{T}}\mathbf{s}_{\mathrm{p}}^{i}, \tag{21}$$

and clearly the algorithm will break down only if

$$(\mathbf{s}_{\mathrm{v}}^{i},\ \mathbf{A}\mathbf{s}_{\mathrm{v}}^{i})+\beta(\mathbf{s}_{\mathrm{p}}^{i},\ \mathbf{C}\mathbf{s}_{\mathrm{p}}^{i})=0, \tag{22}$$

i.e. $\mathbf{s}_{\mathrm{v}}^{i}=0$ and $(\mathbf{s}_{\mathrm{p}}^{i}, \mathbf{C}\mathbf{s}_{\mathrm{p}}^{i})=0$. However, the velocity search direction $\mathbf{s}_{\mathrm{v}}^{i}=0$ only if $\mathbf{B}^{\mathrm{T}}\mathbf{s}_{\mathrm{p}}^{i}=0$ and by the stabilization condition (3)

$$\mathbf{B}^{\mathrm{T}}\mathbf{s}_{\mathrm{p}}^{i}=0 \quad\Rightarrow\quad (\mathbf{s}_{\mathrm{p}}^{i}, \mathbf{C}\mathbf{s}_{\mathrm{p}}^{i})\neq 0,$$

so the algorithm cannot break down prematurely. Further, since the coefficient matrix is symmetric, the algorithm must terminate in a finite number of iterations and hence always converges (assuming exact arithmetic).

For the inner system (21) the matrix $\mathbf{A}$ is symmetric and positive definite so the conjugate gradient method can be applied, thus again defining a two-level iteration. As with Algorithm One, our strategy will be to converge the inner iteration to the same accuracy required for the outer iteration.

As mentioned earlier, the stabilized pressure matrix $\beta\mathbf{C}+\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^{\mathrm{T}}$ is extremely well-conditioned so that in practice the number of outer iterations becomes independent of h as the grid is successively refined. One immediate consequence of this is that for an arbitrary initial grid, if a uniform refinement strategy is adopted, then preconditioning the system (19) will only be of limited benefit (the preconditioned system will have a condition number independent of h, possibly with a smaller constant). Nonetheless, Cahouet and Chabard[14] advocate explicitly constructing the matrix $\mathbf{B}\tilde{\mathbf{A}}^{-1}\mathbf{B}^{\mathrm{T}}$, where $\tilde{\mathbf{A}}$ is an approximation to $\mathbf{A}$, in order to precondition

```
set $p^0 = 0$
$Au^0 = f$
$r_p{}^0 = Bu^0 - g$
$i = 0$
loop:until residual is sufficiently small
    $i = i + 1$
    if $i = 1$
        $s_p{}^1 = r_p{}^0$
    else
        $\delta = \frac{(r_p{}^{i-1}, r_p{}^{i-1})}{(r_p{}^{i-2}, r_p{}^{i-2})}$
        calculate the new pressure search direction
        $s_p{}^i = r_p{}^{i-1} + \delta s_p{}^{i-1}$
    endif
    solve for the new velocity search direction
    $As_v{}^i = B^t s_p{}^i$
    calculate the step length
    $\alpha = \frac{(r_p{}^{i-1}, r_p{}^{i-1})}{((s_v{}^i, As_v{}^i) + \beta(s_p{}^i, Cs_p{}^i))}$
    update the pressure, the velocity and the residual
    $p^i = p^{i-1} + \alpha s_p{}^i$
    $u^i = u^{i-1} - \alpha s_v{}^i$
    $r_p{}^i = r_p{}^{i-1} - \alpha(Bs_v{}^i - \beta Cs_p{}^i)$
endloop
```

Algorithm Two

(19) in the case of a stable mixed approximation, and Vincent and Boyer[15] have extended this idea to a stabilized mixed approximation. Whilst the value of this is debatable in the case of regular grids, especially if the construction of the preconditioner is expensive, when using a non-quasi-uniform sequence of grids the use of a preconditioner of the form discussed in Reference 14 or 15 appears to be vital if Algorithm Two is to perform reasonably.

Regarding the sensitivity of Algorithm Two to the magnitude of the parameter β in the stabilized pressure equation (19), we show in Section 5 that in practice the rate of convergence is fairly insensitive to the magnitude of β (at least in the range 10^{-2}–10^2). This gives the method an inherent robustness, since there are no parameters which require careful tuning, and is in contrast to Algorithm One where the penalty parameter needs to be chosen carefully.

Our only doubt concerning the optimality of the dual-variable iteration is that the complexity associated with having two-level iteration scheme might be unnecessary. This is explored further in the next section.

4. A ONE-LEVEL PRECONDITIONED CONJUGATE GRADIENT ALGORITHM

An interesting alternative to the use of a two-level iteration scheme (which is necessary for the primal or dual variable formulation) is to apply a (preconditioned) iterative scheme directly to the stabilized system (2) with coefficient matrix

$$\mathbf{K}=\begin{pmatrix}\mathbf{A} & \mathbf{B}^{\mathrm{T}}\\ \mathbf{B} & -\beta\mathbf{C}\end{pmatrix}. \tag{23}$$

Assuming that **K** is preconditioned by a symmetric positive definite matrix **M**, two possible schemes are the conjugate residual algorithm and the conjugate gradient algorithm. With such a preconditioner **M** the preconditioned system $\mathbf{M}^{-1/2}\mathbf{K}\mathbf{M}^{-1/2}$ is symmetric so the conjugate residual algorithm (see e.g. the robust ORTHODIR implementation of Jea and Young[16] (p. 403)) will converge, giving a monotonic reduction in residuals in the $\mathbf{M}^{-1}$ norm. Numerical experiments with such preconditioned conjugate residual methods show that they invariably converge but that the speed of convergence depends crucially on the stabilization (in particular on the magnitude of the parameter β). This suggests the possibility of 'tuning' the choice of β so as to optimize the rate of convergence. A theoretical analysis of the convergence of the conjugate *residual algorithm* in the case of the diagonal preconditioner

$$\mathbf{M}=\begin{pmatrix}\operatorname{diag}(\mathbf{A}) & 0\\ 0 & \beta\operatorname{diag}(\mathbf{C})\end{pmatrix}=\begin{pmatrix}\mathbf{M}_{\mathrm{A}} & 0\\ 0 & \mathbf{M}_{\mathrm{C}}\end{pmatrix} \tag{24}$$

is presented in Reference 17.

Here we would like to consider applying the standard preconditioned conjugate gradient algorithm as defined by Golub and Van Loan[18] (p. 374) to (2) with coefficient matrix (23) and preconditioning matrix (24). This leads to Algorithm Three. However, in contrast to Algorithm Two, the stabilization condition (3) does not provide a guarantee that the step length in Algorithm Three is bounded, so in theory the conjugate gradient method may fail to converge. Our results in the next section are testimony to the fact that the method does not break down in practice. In fact the method displays very similar convergence characteristics to the conjugate residual algorithm, although of course the nice property of the monotonic reduction in residuals is lost.

Referring to other possible preconditioning schemes, the presence of the negative entries on the diagonal of **K** means that some preconditioners (for example ICCG) are not directly applicable. Some results using a conjugate gradient algorithm with an indefinite diagonal preconditioner are presented in Reference 9. Another feasible choice would be to use an element-by-element preconditioning method.[19] More sophisticated preconditioners lead back to the dual formulation (19); see References 12 and 13 for details in the stable mixed approximation case. We emphasize here that the major advantage of this direct approach is the fact that there is only one level of iteration and therefore implementation is particularly simple, especially with parallel processing in mind. This advantage will be lost if the preconditioner is too sophisticated.

5. NUMERICAL RESULTS

In this section we compare the numerical performance of the three algorithms described in the previous sections for solving the classical two-dimensional lid-driven cavity problem. Only half the solution domain was modelled, by exploiting symmetry, and the discretization was by uniformly refined grids of $\mathbf{Q}_1$–$\mathbf{P}_0$ square elements using the so-called local stabilization procedure of Silvester and Kechkar.[9] See Reference 9 for a full description of this problem. The com-

```
set $x_u{}^0 = 0$
set $x_p{}^0 = 0$
$r_u{}^0 = f$
$r_p{}^0 = g$
$i = 0$
loop:until residual is sufficiently small
    $z_u{}^i = M_A{}^{-1} r_u{}^i$
    $z_p{}^i = M_C{}^{-1} r_p{}^i$
    $i = i + 1$
    if $i = 1$
        $p_u{}^1 = z_u{}^0$
        $p_p{}^1 = z_p{}^0$
    else
        $\delta = \frac{(r_u{}^{i-1}, z_u{}^{i-1}) + (r_p{}^{i-1}, z_p{}^{i-1})}{(r_u{}^{i-2}, z_u{}^{i-2}) + (r_p{}^{i-2}, z_p{}^{i-2})}$
        update the search direction
        $p_u{}^i = z_u{}^{i-1} + \delta p_u{}^{i-1}$
        $p_p{}^i = z_p{}^{i-1} + \delta p_p{}^{i-1}$
    endif
    $w_u{}^i = A p_u{}^i + B^t p_p{}^i$
    $w_p{}^i = B p_u{}^i - \beta C p_p{}^i$
    update the step length
    $\alpha = \frac{(r_u{}^{i-1}, z_u{}^{i-1}) + (r_p{}^{i-1}, z_p{}^{i-1})}{(w_u{}^i, p_u{}^i) + (w_p{}^i, p_p{}^i)}$
    update the solution and the residual
    $x_u{}^i = x_u{}^{i-1} + \alpha p_u{}^i$
    $x_p{}^i = x_p{}^{i-1} + \alpha p_p{}^i$
    $r_u{}^i = r_u{}^{i-1} - \alpha w_u{}^i$
    $r_p{}^i = r_p{}^{i-1} - \alpha w_p{}^i$
endloop
```

Algorithm Three

putations were done using Pro-MATLAB on an Ardent Titan computer. In all cases the tolerance for convergence was a reduction of 10^{-6} in the L_2-norm of the residual.

An important feature of the local stabilization approach is the fact that solution accuracy is preserved in the limit of an arbitrarily large stabilization parameter. This is in contrast to the so-called global stabilization methods,[6-8] where the accuracy deteriorates as $\beta \rightarrow \infty$. As a consequence of this, using the local stabilization approach it is feasible to tune the stabilization parameter to achieve the best rate of convergence for each of the three algorithms above. Thus in Tables I–III we present iteration counts for three values of the stabilization parameter in the range 10^{-2}–10^2 in the knowledge that the numerical solutions are of comparable accuracy in each case.

For the solution of the inner systems in Algorithms One and Two we used the basic one-step Jacobi preconditioned conjugate gradient as described by Adams.[20] The current estimate of the solution at each outer iteration was used as the initial guess for the inner iteration; consequently the number of inner iterations required to reach the tolerance steadily decreased as the outer iteration converged. In Tables I and II both the number of outer iterations and the total number of inner iterations performed in each case are presented.

Using Algorithm One with local stabilization and a macroelement numbering of the pressure degrees of freedom, the matrix $\mathbf{D}$ turns out to be block diagonal. The matrix $\mathbf{D}^{-1}$ can therefore be stored as a single inverted block, thus reducing the storage required and simplifying the calculation of the 'stabilized penalty matrix' $(1/\varepsilon)\mathbf{B}^{\mathrm{T}}\mathbf{D}^{-1}\mathbf{B}$.

Considering the results in Table I, it is clear that there is no great change in the number of inner iterations when increasing the stabilization parameter from $\beta=1$ to $\beta=10^2$. For smaller values of the stabilization parameter (e.g. $\beta=10^{-2}$) more iterations are required, and in the limit of $\beta\rightarrow 0$ convergence becomes arbitrarily slow.

The projected Uzawa method of Fortin and Boivin[21] is very similar to the iterated penalty method above. The principle difference is that in Reference 21 the coefficient matrix of the inner iteration is essentially $\mathbf{A}+(1/\varepsilon)\mathbf{B}^{\mathrm{T}}\mathbf{B}$. Comparing our iterated penalty method results with those presented by Fortin and Boivin,[21] the superiority of our approach is obvious. In Reference 21 there is a 'residual instability' which seems to preclude convergence for $\varepsilon\leqslant 10^{-2}$. Our method allows us to choose a much smaller value for ε without running into difficulty.

Looking at Table II, an important point to note is that for all values of β the number of outer iterations is tending to a constant as the grid is refined, indicating that the condition of the

Table I. Number of iterations required by Algorithm One ($\varepsilon = 10^{-4}$)

	$\beta = 10^{-2}$		$\beta = 1$		$\beta = 10^2$	
Grid	Outer	Inner	Outer	Inner	Outer	Inner
2×4	5	39	4	30	3	20
4×8	6	244	4	156	4	126
8×16	10	746	6	481	6	459
16×32	21	1967	9	1165	9	1197

Table II. Number of iterations required by Algorithm Two

	$\beta = 10^{-2}$		$\beta = 1$		$\beta = 10^2$	
Grid	Outer	Inner	Outer	Inner	Outer	Inner
2×4	8	32	8	32	8	32
4×8	14	173	15	187	19	240
8×16	14	336	18	424	22	540
16×32	14	575	19	750	25	1070

Table III. Number of iterations required by Algorithm Three

Grid	$\beta = 10^{-2}$	$\beta = 1$	$\beta = 10^2$
2×4	15	15	14
4×8	74	63	66
8×16	209	177	162
16×32	529	426	322

stabilized pressure matrix is indeed independent of the mesh parameter h. Clearly, the number of outer iterations tends to a different constant for different values of β, with more iterations required as β gets larger. This reflects the fact that whilst the stabilized pressure matrix has a condition number independent of h for all values of β, the 'critical' condition number varies with β. Numerical experiments show that a value of $\beta = O(10^{-1})$ is the best choice in this case.

It is immediately obvious from Table III that for Algorithm Three an increase in β from 10^{-2} reduces the number of iterations required. The value of $\beta = O(10^2)$ turns out to be optimal in this case, because for larger values of β, i.e. $\beta = O(10^k)$ with $k > 2$, the numer of iterations can be seen to increase monotonically with k.

In conclusion, comparing the total iteration counts in the tables above, it is clear that both the two-level and one-level conjugate gradient algorithms are generally superior to the iterated penalty method. The two-level scheme is probably best overall in view of the inherent robustness of the iteration. However, if the stabilization parameter is appropriately chosen (i.e. $O(10^2)$), then the one-level scheme is potentially the most efficient of the three methods considered and appears to be perfectly robust in practice.

6. IMPACT ON NAVIER-STOKES SOLVERS

From the outset it is stressed that the algorithms described herein are only directly applicable when the matrix $\mathbf{A}$ in (1) is symmetric and positive definite. A conventional discretization of the steady state Navier-Stokes equations leads to a non-linear algebraic system of the form (1) where the coefficients of $\mathbf{A}$ depend on the velocity field $\mathbf{u}$. A standard non-linear iterative solver (e.g. the Newton-Raphson scheme) then leads to a linearized system of the form (1), except that $\mathbf{A}$ is *non-symmetric* (although it may still be positive definite). Whilst it is straightforward to extend all three algorithms described herein to the case of an unsymmetric matrix $\mathbf{A}$, it is not an approach that we would advocate.

When solving a time-dependent Navier-Stokes problem, there are several ways of constructing a solution process so that one or more generalized Stokes systems (each with a symmetric $\mathbf{A}$) must be solved at each time step. One of the most efficient is the 'transport diffusion algorithm' which is described in Reference 22 and has been proven to be very effective in the case of flow at *high* Reynolds numbers. Other solution algorithms for the time-dependent and steady state Navier-Stokes equations are built upon the fact that efficient Stokes solvers are available; for example, the least-squares methods described in Reference 23 (Chap. VII). Working within such a 'least-squares framework', the efficiency of the underlying Stokes solver is of crucial importance.

REFERENCES

1. L. A. Hageman and D. M. Young, *Applied Iterative Methods*, Academic, New York, 1981.
2. R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.

3. D. M. Young, *Iterative Solution of Large Linear Systems*, Academic, New York, 1971.
4. F. Thomasset, *Implementation of Finite Element Methods for Navier–Stokes Equations, Springer Series in Computational Physics*, Springer, Berlin, 1981.
5. M. D. Gunzburger, *Finite Element Methods for Viscous Incompressible Flows*, Academic, London, 1989.
6. F. Brezzi and J. Pitkäranta, 'On the stabilization of finite element approximations of the Stokes problem', in W. Hackbusch (ed.), *Efficient Solutions of Elliptic Systems, Notes on Numerical Fluid Mechanics, Vol. 10*, Vieweg, Braunschweig, 1984, pp. 11–19.
7. T. J. R. Hughes and L. P. Franca, 'A new finite element formulation for CFD: VII. The Stokes problem with various well-posed boundary conditions: symmetric formulations that converge for all velocity/pressure spaces', *Comput. Methods Appl. Mech. Eng.*, **65**, 85–96 (1987).
8. F. Brezzi and J. Douglas, 'Stabilized mixed methods for the Stokes problem', *Numer. Math.*, **53**, 225–235 (1988).
9. D. J. Silvester and N. Kechkar, 'Stabilised bilinear–constant velocity–pressure finite elements for the conjugate gradient solution of the Stokes problem', *Comput. Methods Appl. Mech. Eng.*, **79**, 71–86 (1990).
10. M. Fortin and R. Glowinski, *Augmented Lagrangian Methods: Application to the Numerical Solution of Boundary-value Problems*, North-Holland, Amsterdam, 1983.
11. R. Verfürth, 'A combined conjugate gradient–multigrid algorithm for the numerical solution of the Stokes problem', *IMA J. Numer. Anal.*, **4**, 441–455 (1984).
12. J. Bramble and J. Pasciak, 'A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems', *Math. Comput.*, **50**, 1–17 (1988).
13. R. E. Bank, B. D. Welfert and H. Yserentant, 'A class of iterative methods for solving saddle point problems', *Numer. Math.*, **56**, 645–666 (1990).
14. 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).
15. C. Vincent and R. Boyer, to appear in *Int. j. numer. methods fluids.*
16. K. Jea and D. Young, 'On the simplification of generalized conjugate gradient methods for nonsymmetrizable linear systems', *Linear Algebra Appl.*, **52/53**, 399–417 (1983).
17. A. J. Wathen and D. J. Silvester, submitted to *SIAM J. Numer. Anal.*
18. G. H. Golub and C. F. Van Loan, *Matrix Computations*, Johns Hopkins University Press, Baltimore, MD, 1983.
19. T. J. R. Hughes, R. M. Ferencz and J. O. Hallquist, 'Large-scale vectorized implicit calculations in solid mechanics on a CRAY X-MP/48 utilizing EBE preconditioned conjugate gradients', *Comput. Methods Appl. Mech. Eng.*, **61**, 215–248 (1987).
20. L. Adams, '*M*-step preconditioned conjugate gradient methods', *SIAM J. Sci. Stat. Comput.*, **6**, 452–463 (1985).
21. M. Fortin and S. Boivin, 'Iterative stabilization of the bilinear velocity–constant pressure element', *Int. j. numer. methods fluids*, **10**, 125–140 (1990).
22. O. Pironneau, 'On the transport diffusion algorithm and its application to the Navier–Stokes equations', *Numer. Math.*, **38**, 309–332 (1982).
23. R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer, New York, 1984.