INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 19, 67-83 (1994)

ITERATIVE SOLUTION TECHNIQUES FOR THE STOKES AND NAVIER–STOKES EQUATIONS

ALISON RAMAGE* AND ANDREW J. WATHEN
School of Mathematics, University of Bristol, Bristol BS8 1TW, U.K.

SUMMARY

The linear system arising from a Lagrange–Galerkin mixed finite element approximation of the Navier–Stokes and continuity equations is symmetric indefinite and has the same block structure as a system arising from a mixed finite element discretization of a Stokes problem. This paper considers the iterative solution of such a system, comparing the performance of the one-level preconditioned conjugate residual method for indefinite matrices with that of a more traditional two-level pressure correction approach. Asymptotic estimates for the amount of work involved in each method are given together with the results of related numerical experiments.

KEY WORDS Mixed finite elements Navier–Stokes Conjugate gradients

1. INTRODUCTION

The question of how to solve linear systems efficiently plays a very large part in many problems in fluid dynamics, not least in the solution of the Navier–Stokes and continuity equations for incompressible viscous fluid flow. The partial differential equations are

$$u_t + u \cdot \nabla u + \nabla p - \frac{1}{Re} \nabla^2 u = 0, \tag{1}$$

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

where u is velocity, p is pressure and Re is the Reynolds number of the flow.[1] Applying a standard non-linear iterative solver (e.g. the Newton–Raphson method) reduces the problem to one of solving a series of linear systems, but the coefficient matrices involved are in general non-symmetric. Iterative methods are certainly available for such non-symmetric systems (see e.g. Reference 2), but at present theoretical knowledge and analysis of their symmetric counterparts are superior. We therefore adopt a different approach and take advantage of the symmetry which arises naturally in a Lagrange–Galerkin discretization.

Applying the Lagrange–Galerkin method to (1) and (2) in the context of a mixed finite element approximation removes the non-linearity introduced by the advection term and gives rise to a symmetric and indefinite linear system of the form

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ \mathbf{0} \end{bmatrix} \tag{3}$$

* Present address: Department of Mathematics, University of Strathclyde, Glasgow G1 1XH, U.K.

CCC 0271–2091/94/130019–17

Received 29 January 1993
Revised 29 December 1993

at each time step. Here $\mathbf{A}$ is $N_U \times N_U$ and symmetric positive definite, $\mathbf{B}$ is $N_P \times N_U$, $\mathbf{0}$ is the $N_P \times N_P$ zero matrix and the vectors $\mathbf{u}$ and $\mathbf{p}$ contain the N_U nodal velocity and N_P nodal pressure values respectively. This process is described in more detail in Section 2. System (3) may be solved in a number of ways, but since it will often in practice be large and sparse, an iterative rather than a direct approach is usually more practical. In fact, the coefficient matrix in (3) has the same structure as a system arising from the Stokes equations for creeping flow[3] and so various iterative methods which have been proposed for Stokes problems are applicable. Here we compare two basic types: Section 3 discusses the traditional pressure correction approach (which decouples the pressure and velocity equations to form two related symmetric positive definite systems), while Section 4 considers the preconditioned conjugate residual method for tackling the fully coupled indefinite system. Asymptotic estimates for the amount of work involved in each method are calculated in Section 5 and the performance of each algorithm in practice is investigated using numerical experiments in Section 6.

2. THE LAGRANGE–GALERKIN METHOD

The Lagrange–Galerkin method is a numerical technique used for solving advection-dominated diffusion problems for incompressible viscous flow. It combines a Galerkin finite element approach with the method of characteristics, using a relatively straightforward finite difference approximation of the Lagrangian material derivative along particle trajectories. The method was developed in a series of papers over a number of years; a full description of its various formulations and properties can be found in e.g. References 4–7.

Here we discuss only the method's application to (1) and (2). Representing the convective derivative by $\mathrm{D}/\mathrm{D}t = \partial/\partial t + u \cdot \nabla$, the weak forms of these equations are

$$\left(\frac{\mathrm{D}u}{\mathrm{D}t}, v\right) - (p, \nabla \cdot v) + \frac{1}{Re}(\nabla u, \nabla v) = 0, \tag{4}$$

$$-(\nabla \cdot u, q) = 0, \tag{5}$$

where v and q are the appropriate velocity and pressure test functions respectively.[1]

If $\phi_j(x)$ and $\psi_j(x)$ are basis functions spanning the appropriate finite-dimensional subspace, a discrete approximation to these equations can be obtained by writing

$$u^{n+1} = \sum_{j=1}^{N_U} U_j \phi_j, \qquad p^{n+1} = \sum_{j=1}^{N_P} P_j \psi_j, \tag{6}$$

where N_U and N_P are the numbers of velocity and pressure unknowns respectively. A Lagrange–Galerkin discretization replaces the convective derivative in (4) by the matrix term

$$\frac{\mathbf{Mu} - \mathbf{b}}{\delta t}, \tag{7}$$

where $\mathbf{M}$ is the finite element velocity mass matrix defined below, δt is the time step and $\mathbf{u}$ contains the N_U unknown coefficients U_j. Evaluating the vector $\mathbf{b}$ involves tracking back along characteristics to evaluate the solution at the previous time level, i.e. it is calculated using $\mathbf{u}^n$. This is usually the most expensive part of the Lagrange–Galerkin procedure. The remaining terms are discretized by the usual Galerkin mixed finite element approximation to give the solution at the $(n+1)$th time level.

The complete coupled system of (4) and (5) is then represented in matrix terms by

$$\hat{\mathbf{A}}\mathbf{v}_1 = \mathbf{v}_2 \tag{8}$$

or equivalently

$$\begin{bmatrix} \mathbf{A} & \mathbf{B}^{\mathrm{T}} \\ \mathbf{B} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{u}^{n+1} \\ \mathbf{p}^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{b}/\delta t \\ \mathbf{0} \end{bmatrix}. \tag{9}$$

The matrix **A** has diagonal blocks which are linear combinations of the finite element mass (**M**) and stiffness (**K**) matrices, i.e. $M/\delta t + (1/Re)\mathbf{K}$, where

$$\mathbf{M} = \{m_{ij}\}, \qquad m_{ij} = \int_{\Omega} \phi_i \cdot \phi_j \delta\Omega, \tag{10}$$

$$\mathbf{K} = \{k_{ij}\}, \qquad k_{ij} = \int_{\Omega} \nabla\phi_i \cdot \nabla\phi_j \delta\Omega, \tag{11}$$

and is thus symmetric and positive definite. Note that the matrix **B** is defined by

$$\mathbf{B} = \{b_{ij}\}, \qquad b_{ij} = -(\nabla \cdot \phi_i, \psi_j) \tag{12}$$

and is of full rank under the assumption that an LBB-stable pair of finite elements is used.[1] In addition, the symmetric indefinite system (9) has the same form as a system arising from the normal mixed finite element approximation of the Stokes equations for slow incompressible viscous flow.

3. A PRESSURE CORRECTION APPROACH

One of the most popular methods for solving a Stokes-like system is a pressure correction (PC) approach. This involves decoupling the variables to obtain a symmetric positive definite system to solve for pressure. Consider the pair of equations

$$\mathbf{Au} = \mathbf{b} - \mathbf{B}^T\mathbf{p}, \tag{13}$$

$$\mathbf{Bu} = \mathbf{0} \tag{14}$$

obtained from (9). Since **A** is symmetric positive definite, (13) implies

$$\mathbf{Bu} + \mathbf{BA}^{-1}\mathbf{B}^T\mathbf{p} = \mathbf{BA}^{-1}\mathbf{b}. \tag{15}$$

It is clear that if a pressure $\mathbf{p}^*$ can be found such that

$$\mathbf{BA}^{-1}\mathbf{B}^T\mathbf{p}^* = \mathbf{BA}^{-1}\mathbf{b}, \tag{16}$$

then $\mathbf{Bu}^* = \mathbf{0}$ and the continuity equation is satisfied, where $\mathbf{u}^*$ is the related velocity obtained from solving (13). Hence solving (9) is equivalent to solving (13) and (16). Using an LBB-stable finite element method, the matrix $\mathbf{BA}^{-1}\mathbf{B}^T$ is symmetric positive definite.[1]

Iterative solution of symmetric positive definite systems is a much researched topic and one obvious method to use is the preconditioned conjugate gradient algorithm, one of a family of Krylov subspace methods.[8,9] The idea of applying the conjugate gradient method to the reformulated system (16) was first analysed in the context of solving Stokes systems by Verfürth[10] and is applicable to the Lagrange–Galerkin Navier–Stokes system under discussion here. An

(1) $p_0 = 0$ • *initialise pressure vector*

(2) $Au_0 = b$ • *solve for velocity*

(3) $r_0 = Bu_0$ • *calculate initial residual*

(4) $i = 1$ • *initialise iteration count*

begin loop • *loop until residual is small*

if

(5) $i = 1$

then

(6) $s_i = r_0$ • *set pressure search direction*

else

(7) $\delta = \frac{(r_{i-1}, r_{i-1})}{(r_{i-2}, r_{i-2})}$

(8) $s_i = r_{i-1} + \delta s_{i-1}$ • *set pressure search direction*

endif

(9) $Av_i = B^T s_i$ • *solve for velocity search direction*

(10) $\alpha = \frac{(r_{i-1}, r_{i-1})}{(v_i, Av_i)}$ • *set step length*

(11) $p_i = p_{i-1} + \alpha s_i$ • *update pressure*

(12) $u_i = u_{i-1} - \alpha v_i$ • *update velocity*

(13) $r_i = r_{i-1} - \alpha B v_i$ • *update residual*

(14) $i = i + 1$ • *update iteration count*

end loop

Figure 1. Pressure correction algorithm

example of an application of this technique to equations (13) and (16) (hence to system (9)) is detailed in Figure 1. The coefficient matrix $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ is never actually constructed: all that is required are subroutines to multiply $\mathbf{B}$ and $\mathbf{B}^T$ by a vector and a way of efficiently solving the linear systems with $\mathbf{A}$ as coefficient matrix in steps (2) and (9).

Verfürth[10] used a multigrid method for solving the inner velocity systems, but his idea has been developed in many subsequent papers by a variety of authors using different inner solvers. Efficient solution of these inner systems is very important for the overall effective performance of the algorithm. Bramble and Pasciak[11] and Bank *et al.*[12] use different algorithms based on hierarchical methods. Cahouet and Chabard[13] and Atanga and Silvester[14] both use a second set of conjugate gradient iterations for the inner system, the former with modified incomplete Cholesky factorization as a preconditioner and the latter with diagonal scaling. Asymptotic estimates of the work involved in using both these methods (PCMIC and PCDS respectively) are given in Section 5. An implementation of the latter algorithm is used in the numerical experiments of Section 6.

4. A CONJUGATE RESIDUAL APPROACH

As an alternative to the two-level approach above, in this section we consider an iterative method which can be applied directly to the coupled indefinite system (8). The preconditioned conjugate

(1) $r_0 = \hat{b} - \hat{A}x_0$ • *calculate initial residual*

(2) $p_0 = \hat{M}^{-1}r_0$ • *initialise search vector*

(3) $\alpha_0 = 0$

(4) $i = 1$ • *initialise iteration count*

begin loop • *loop until residual is small*

(5) $\alpha_i = \frac{(r_{i-1}, \hat{M}^{-1}\hat{A}p_{i-1})}{(\hat{A}p_{i-1}, \hat{M}^{-1}\hat{A}p_{i-1})}$ • *step length*

(6) $x_i = x_{i-1} + \alpha_i p_{i-1}$ • *update solution*

(7) $r_i = r_{i-1} - \alpha_i \hat{A}p_{i-1}$ • *update residual*

if $|\alpha_i| \leq \epsilon$ then •*ORTHODIR step*

(8) $\gamma_i = \frac{(\hat{M}^{-1}\hat{A}p_{i-1}, \hat{A}\hat{M}^{-1}\hat{A}p_{i-1})}{(\hat{A}p_{i-1}, \hat{M}^{-1}\hat{A}p_{i-1})}$

(9) $c_i = \begin{cases} 1, & |\alpha_{i-1}| \leq \epsilon \\ \frac{-1}{\alpha_{i-1}}, & |\alpha_{i-1}| > \epsilon \end{cases}$

(10) $\sigma_i = c_i \frac{(\hat{M}^{-1}\hat{A}p_{i-1}, \hat{A}p_{i-1})}{(\hat{A}p_{i-2}, \hat{M}^{-1}\hat{A}p_{i-2})}$

(11) $p_i = \hat{M}^{-1}\hat{A}p_{i-1} - \gamma_i p_{i-1} - \sigma_i p_{i-2}$

else •*ORTHOMIN step*

(12) $s_i = \hat{M}^{-1}r_i$

(13) $\beta_i = \frac{(\hat{M}^{-1}\hat{A}p_{i-1}, \hat{A}\hat{M}^{-1}r_i)}{(\hat{A}p_{i-1}, \hat{M}^{-1}\hat{A}p_{i-1})}$

(14) $p_i = s_i - \beta_i p_{i-1}$

endif

(15) $i = i + 1$ • *update iteration count*

end loop

Figure 2. Preconditioned conjugate residual algorithm

residual (CR) method is another member of the Krylov subspace family and can be used for any symmetric indefinite system (see e.g. References 8 and 15). The method works by minimizing the L_2-norm of the residual at each step (see Appendix for a more detailed analysis of its convergence rate). Here the exact algorithm used is a hybrid of the fast ORTHOMIN and robust ORTHODIR forms with preconditioner $\hat{\mathbf{M}}$:[8] the details are given in Figure 2.

For the algorithm implemented in the numerical experiments of Section 6 we again use a very simple preconditioner, namely diagonal scaling.[16] The precise form of the preconditioning matrix is

$$\hat{\mathbf{M}} = \begin{bmatrix} \mathbf{D_A} & \mathbf{0} \\ \mathbf{0} & \mathbf{D_P} \end{bmatrix}, \tag{17}$$

where $\mathbf{D_A}$ is the diagonal of $\mathbf{A}$ and $\mathbf{D_P}$ is the diagonal of the pressure mass matrix $\mathbf{P}$. This is of course only one of a number of possible preconditioners. For example, $\mathbf{D_A}$ could easily be replaced by a modified incomplete factorization, equivalent to applying MICCG to the inner

system of PC.[17] In particular, element-based preconditioners (e.g. element-by-element methods[18]) may be efficient. Note, however, that applying any preconditioner to the top left diagonal block of $\hat{\mathbf{A}}$ will have an effect similar to applying the same preconditioner to the inner velocity system of PC. For example, if an optimal solver (such as multigrid) is used with the PC approach so that the solution time scales linearly with the problem size, then using the same method as a preconditioner in place of the block $\mathbf{D}_A$ in (17) will render the convergence of CR independent of the grid size. That is, the work in CR solution will also grow linearly with the problem size.[19] However, the PC inner system must be solved to high accuracy to preserve conjugacy and hence speed of convergence in the outer conjugate gradient iteration. A distinct advantage of CR is that the preconditioner used does not have to be an exact solve. For example, Silvester and Wathen[19] note that PC is *not* convergent when only one multigrid V-cycle is used for the inner solve, whereas CR converges at a rate independent of grid size when one multigrid V-cycle is used as the upper left block preconditioner. In their computations four V-cycles were required to obtain PC convergence. In the numerical experiments of Section 6 we have not taken advantage of this: we have used the smallest possible number of inner conjugate gradient iterations to give convergence by choosing an inner convergence tolerance just one order of magnitude smaller than the outer convergence tolerance.

5. ASYMPTOTIC WORK ESTIMATES

In this section we construct some asymptotic estimates of the work involved in implementing the above solvers. Consider a regular finite element grid with V vertices in each dimension a distance h apart. If this lattice is partitioned into nine-node rectangles in two dimensions, using biquadratic velocity and bilinear pressure finite element basis functions (see e.g Reference 1, p. 34) gives a total of $N_U = 2V^2$ velocity and $N_P \simeq \frac{1}{4}V^2$ pressure unknowns. Using the same basis functions in three dimensions (with 27 node brick elements) gives $N_U = 3V^3$ and $N_P \simeq \frac{1}{8}V^3$. In both cases we denote the total number of unknowns (and hence the dimension of the coefficient matrix in (8)) by $N = N_U + N_P$.

To obtain work estimates for the above iterative methods, we must consider two separate factors, namely the amount of work per iteration and the number of iterations required to obtain convergence. The former is straightforward to compute, since both the conjugate gradient and conjugate residual methods require $O(D)$ flops per iteration, where D is the dimension of the coefficient matrix. The latter factor is, however, more complicated to ascertain, depending not only on the method itself but also on the eigenvalue distribution of the preconditioned coefficient matrix. Below we derive a general bound on the number of iterations required before examining the methods used in the above sections separately.

Let $\hat{\mathbf{x}}$ be the exact solution of the preconditioned linear system and represent the kth error vector by $\|\mathbf{e}_k\| = \|\mathbf{x}_k - \hat{\mathbf{x}}\|$. An estimate of the number of iterations required to achieve

$$\|\mathbf{e}_k\| \leq \varepsilon \|\mathbf{e}_0\| \quad \forall \mathbf{e}_0 \in \mathbb{R}^N, \tag{18}$$

for a particular tolerance ϵ can be obtained by defining the average reduction factor per iteration to be (see Reference 20, p. 62)

$$\sigma = \left(\frac{\|\mathbf{e}_k\|}{\|\mathbf{e}_0\|} \right)^{1/k}. \tag{19}$$

Since the norm in which error reduction is monotonic varies from method to method, it is convenient to have such an estimate of iterative convergence which is independent of the specific

norm used. Some examples of how the reduction factor is calculated for a particular method are given in the Appendix. For the moment we assume that, for large k, σ has the general form

$$\alpha^{1/k}(1 - \beta h^p), \tag{20}$$

where α, β and p are real positive constants and h is the mesh spacing parameter ($h \simeq 1/V$). From (18) and (19), for iterative convergence we require

$$\alpha(1 - \beta h^p)^k \leqslant \varepsilon, \tag{21}$$

which implies

$$k \geqslant \frac{\ln \varepsilon - \ln \alpha}{\ln(1 - \beta h^p)}. \tag{22}$$

Since

$$\frac{\ln \varepsilon - \ln \alpha}{\ln(1 - \beta h^p)} \simeq \frac{\ln \varepsilon - \ln \alpha}{-\beta h^p(1 + \beta h^p/2)} \tag{23}$$

$$\simeq -\frac{\ln \varepsilon - \ln \alpha}{\beta h^p}, \tag{24}$$

a general bound on the number of iterations required to achieve convergence to within tolerance ϵ is therefore given by

$$k > \frac{\ln \alpha - \ln \varepsilon}{\beta h^p}. \tag{25}$$

Each iterative method described above must now be considered separately. The results are derived in the following subsections and summarized in Table I. Note that c_i will be used to represent a real positive constant.

Pressure correction

Both the inner and outer iterations in Section 3 involve an application of the conjugate gradient method. For the outer pressure iteration the eigenvalues of the coefficient matrix $\mathbf{B}\mathbf{A}^{-1}\mathbf{B}^T$ in (8) are independent of the mesh parameter h as $h \to 0$,[13] so the number of iterations required for conjugate gradient convergence is also independent of h (see Appendix). For the inner velocity iteration the coefficient matrix has diagonal blocks which are linear combinations of $\mathbf{M}$ and $\mathbf{K}$ as defined in Section 3. Calculating the reduction factor, which is affected by the choice of preconditioner, gives us the means of constructing overall asymptotic work estimates in both two and three dimensions.

PCDS. The eigenvalues of the diagonally scaled velocity matrix in (13) are dominated by those of $\mathbf{K}$ and are therefore contained in an interval $(c_1 h^2, c_2)$.[21] For large k this gives a reduction factor of

$$\sigma = 2^{1/k}(1 - 2h) \tag{26}$$

(see Appendix), hence bound (25) gives

$$k > \frac{\ln 2 - \ln \varepsilon}{2h}. \tag{27}$$

The outer iteration involves a constant number of iterations in both 2D and 3D and so requires cN_P flops. From (27) each one of these iterations requires $O(N_U^{1/2})$ inner iterations in two dimensions and $O(N_U^{1/3})$ inner iterations in three dimensions (since $V \simeq h^{-1}$). Thus the flop count estimates are $c_3(N_P + c_4 N_U^{3/2})$ and $c_5(N_P + c_6 N_U^{4/3})$ respectively. Taking $N_U \simeq 8N_P$ in two dimensions and $N_U \simeq 24N_P$ in three dimensions (and omitting the unknown constants in each case) gives overall estimates of

$$\text{2D:}\quad (1 + 8^{3/2})N_P^{3/2}, \qquad \text{3D:}\quad (1 + 24^{4/3})N_P^{4/3}. \tag{28}$$

In particular, note that we have dropped the constant c_3 in these estimates, which is the number of outer iterations required for convergence.

PCMIC. Applying MIC factorization to the finite element stiffness matrix leads to a preconditioned matrix whose eigenvalues lie in an interval $[1, c_7 h^{-1})$,[22] giving an estimated iteration count of

$$k > \frac{\ln 2 - \ln \varepsilon}{2h^{1/2}}. \tag{29}$$

Again the outer iteration will involve cN_P flops and so the total flop count estimates are $c_8(N_P + c_9 N_U^{5/4})$ and $c_{10}(N_P + c_{11} N_U^{7/6})$ in 2D and 3D respectively. In terms of N_P these become

$$\text{2D:}\quad (1 + 8^{5/4})N_P^{5/4}, \qquad \text{3D:}\quad (1 + 24^{7/6})N_P^{7/6}. \tag{30}$$

Preconditioned conjugate residuals

If the preconditioned conjugate residual method is applied to a general symmetric indefinite linear system, results analogous to those stated above for the symmetric positive definite case are straightforward to obtain. Since they depend heavily on the distribution of the eigenvalues of the underlying coefficient matrix, we consider two separate cases:

(i) making the standard theoretical assumption that the eigenvalues are symmetric about the origin
(ii) taking advantage of knowledge of the eigenvalue spectrum of the coefficient matrix from the Lagrange–Galerkin Navier–Stokes (or equivalently Stokes) system.

CRDS(SYM). If the eigenvalues of the diagonally scaled indefinite matrix are symmetric about the origin and contained in $(-c_{13}, -c_{12}h^2) \cup (c_{12}h^2, c_{13})$, the average reduction factor for large k is given by

$$\sigma = 2^{1/k}(1 - 2h^2) \tag{31}$$

(see Appendix) and so (25) becomes

$$k > \frac{\ln 2 - \ln \varepsilon}{2h^2}. \tag{32}$$

Since there are $O(N)$ flops per iteration, the overall flop count is given by $c_{14}N^2$ in two dimensions and $c_{15}N^{5/3}$ in three dimensions. To obtain estimates which are easy to compare, we again express these figures in terms of the total number of pressure unknowns, N_P. Since $N = N_U + N_P = 9N_P$ in two dimensions and $N = 25N_P$ in three dimensions, the estimates become

$$\text{2D:}\quad 9^2N_P^2, \qquad \text{3D:}\quad 25^{5/3}N_P^{5/3}. \tag{33}$$

CRMIC(SYM). Again assuming that the eigenvalues are symmetric about the origin, after scaling with modified incomplete Cholesky factorization, they will lie in the union of intervals $(-c_{16}h^{-1}, -1] \cup [1, c_{16}h^{-1})$, giving an iteration bound

$$k > \frac{\ln 2 - \ln \varepsilon}{2h}. \tag{34}$$

The overall flop count is thus $c_{17}N^{3/2}$ in two dimensions and $c_{18}N^{4/3}$ in three dimensions or

$$\text{2D:}\quad 9^{3/2}N_P^{3/2}, \qquad \text{3D:}\quad 25^{4/3}N_P^{4/3}. \tag{35}$$

CRDS(LG). In the particular case of Section 4 the preconditioned conjugate residual method is applied to problem (8) with preconditioner $\mathbf{M}$. The eigenvalues of the diagonally preconditioned matrix are dominated by those of the associated Stokes problem and are therefore not symmetric about the origin, but instead lie in the union of intervals $(-c_{22}, -c_{21}h) \cup (c_{19}h^2, c_{20})$.[16] Finding an expression for the reduction factor is more difficult owing to this lack of symmetry, but it can be shown that, for large k,[23]

$$\sigma = 2^{1/k}(1 - 2h^{3/2}). \tag{36}$$

This leads to a bound on the number of iterations required to achieve convergence of

$$k > \frac{\ln 2 - \ln \varepsilon}{2h^{3/2}}. \tag{37}$$

Hence we require $c_{23}N^{3/4}$ iterations in two dimensions and $c_{24}N^{1/2}$ iterations in three dimensions, giving overall asymptotic work estimates of

$$\text{2D:}\quad 9^{7/4}N_P^{7/4}, \qquad \text{3D:}\quad 25^{3/2}N_P^{3/2}. \tag{38}$$

CRMIC(LG). When modified incomplete Cholesky preconditioning has been used in the top left block of $\mathbf{M}$, the eigenvalues of the Stokes matrix lie in $(-c_{27}h^{-1/2}, -c_{26}) \cup [1, c_{25}h^{-1})$.[17] The contraction factor can again be used to give an iteration bound of

$$k > \frac{\ln 2 - \ln \varepsilon}{2h^{3/4}}. \tag{39}$$

This implies that for convergence we require $c_{28}N^{3/8}$ iterations in two dimensions and $c_{29}N^{1/4}$ iterations in three dimensions. In terms of N_P, overall work estimates are

$$\text{2D:}\quad 9^{11/8}N_P^{11/8}, \qquad \text{3D:}\quad 25^{5/4}N_P^{5/4}. \tag{40}$$

A summary of these results is given in Table I. Again all unknown constants have been discarded, while those explicitly calculated in the above process have been included to retain as much information as possible. The results are shown graphically in Figures 3 and 4. Although

Table I. Summary of asymptotic work estimates

Method	2D	2D
PCDS	$24N_P^{3/2}$	$70N_P^{4/3}$
PCMIC	$14N_P^{5/4}$	$42N_P^{7/6}$
CRDS(SYM)	$81N_P^{2}$	$214N_P^{5/3}$
CRMIC(SYM)	$27N_P^{3/2}$	$73N_P^{4/3}$
CRDS(LG)	$47N_P^{7/4}$	$125N_P^{3/2}$
CRMIC(LG)	$21N_P^{11/8}$	$56N_P^{5/4}$

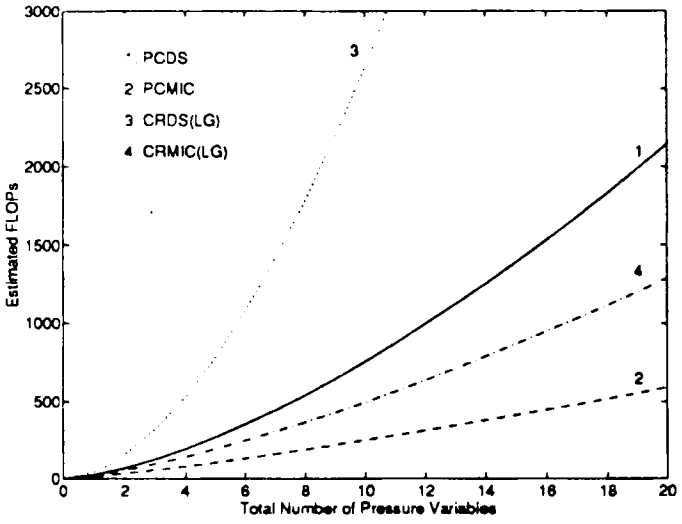

Figure 3. Asymptotic work estimates in 2D

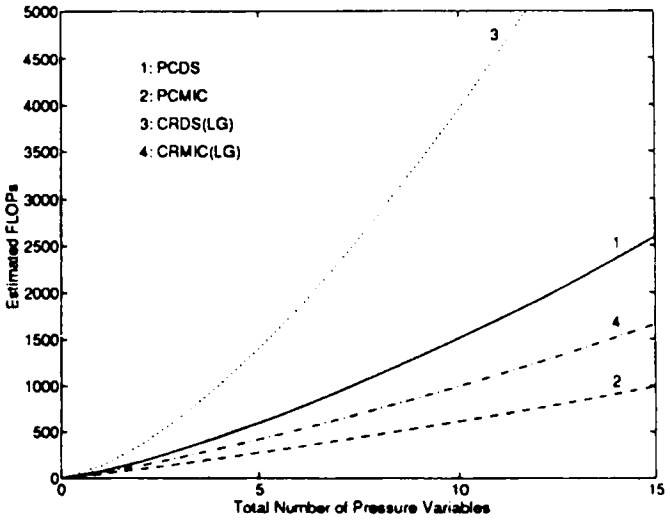

Figure 4. Asymptotic work estimates in 3D

such figures are only approximate, asymptotic work estimates are often used as a guide to the prospective relative efficiency of methods. Except for those which correspond to the number of outer PC iterations, the unknown constants hidden in these estimates are determined by implementation and architectural features which may play an important role in determining how a method will actually perform in practice. Modified incomplete Cholesky factorization is clearly an advantageous procedure to use, but of course this is only one example of possible preconditioning. For the purposes of this comparison we note only that any preconditioner applied to the inner pressure correction system will improve performance in a similar way if used in the top left diagonal block of the conjugate residual preconditioner $\mathbf{M}$.

A similar comment applies to the memory requirement for each method. Both PC and CR require $O(N)$ storage in order to represent the problem, while a further $7N$ for PC and $9N$ for CR are needed for the few auxiliary vectors required in the solution process. They will have essentially identical memory requirements for any preconditioner: all common preconditioners require also only $O(N)$ storage. This is by contrast with direct methods which have typical requirements of $O(N^{3/2})$ locations in two dimensions and $O(N^{5/3})$ locations in three dimensions in order to store the matrix fill that arises from elimination (see Reference 21, p. 388).

6. NUMERICAL EXPERIMENTS

In this section we compare the theoretical results above with a number of numerical experiments. These involve solving the Navier–Stokes and continuity equations (1) and (2) on two- and three-dimensional finite element grids. The domain chosen is the unit square or unit cube: the grids can thus be designed to minimize the effect on the solution procedure of grid irregularities. In two dimensions we consider a series of regular finite element grids of $n \times n$ eight-node square serendipity elements. The boundary conditions are no flow on the top and bottom walls,

$$u_1 = 0{\cdot}4y(1 - y), \qquad u_2 = 0 \tag{41}$$

at the inlet and full traction at the outlet. In three dimensions an equivalent partitioning would cause an overdetermination of the pressure unknowns in the corners of the domain. To avoid this complication, an extra layer of elements has been introduced around the edge of each three-dimensional grid, leaving the 'core' of the grid as an $n \times n \times n$ block of 20-node cuboid elements. The boundary conditions are analogous to the two-dimensional case: no flow on the top, bottom, front and back walls,

$$u_1 = u_2 = 0, \qquad u_3 = xy(1 - x)(1 - y) \tag{42}$$

at the inlet and full traction at the outlet. In all cases the Reynolds number is 10·0 and the Lagrange–Galerkin time step δt is 1·0 (although the results presented here are for one time step only). The experiments were carried out in FORTRAN on a SUN SPARC2 workstation within the framework of Nuclear Electric plc.'s finite-element-based CFD code FEAT.[24] This is a general finite element code which uses bilinear pressure and biquadratic velocity serendipity elements (see Reference 1, p. 34) to model various incompressible fluid flow situations.

The numerical results are given in Tables II–IV. In Tables II and IV, the top five rows contain some specifications of the grids, where N_E is the total number of elements. The times are in CPU seconds and are for the solve process only; they do not include the time taken to set up the finite element matrices. In each case the preconditioner used is diagonal scaling. This means that throughout this section the acronym PC corresponds to the method PCDS in the previous section, while CR refers to CRDS(LG). This is because the eigenvalues of the Lagrange–Galerkin

Table II. Numerical results in two dimensions

	n	2	4	8	16	32	64
	N	25	105	433	1761	7105	28545
	N_U	16	80	352	1472	6016	24320
	N_P	9	25	81	289	1089	4225
	N_E	4	16	64	256	1024	4096
PC	Outer iterations	7	14	19	21	22	22
$\varepsilon_O = 10^{-3}$	Inner iterations (average)	7	14	27	51	101	195
$\varepsilon_I = 10^{-4}$	Solve time	0·08	0·81	11·21	107·70	986·12	7542·65
CR	Iterations	14	37	101	209	460	1055
$\varepsilon = 10^{-3}$	Solve time	0·05	0·24	3·44	31·06	288·61	2704·57

Table III. 2D results with fixed number of inner iterations

	n	8	16	32	64
PC	Outer iterations	19	21	29	32
$\varepsilon_O = 10^{-3}$	Inner iterations	20	20	20	20
	Solve time	8·69	43·03	251·37	1138·39
	DPs of accuracy	2	1	0	0

Table IV. Numerical results in three dimensions

	n	2	4	8	10
	N	278	1476	9128	16830
	N_U	228	1272	8112	15060
	N_P	50	204	1016	1770
	N_E	24	128	768	1400
PC	Outer iterations	35	46	26	26
$\varepsilon_O = 10^{-3}$	Inner iterations (average)	27	27	39	47
$\varepsilon_I = 10^{-4}$	Solve time	46·22	470·19	2942·52	6749·24
CR	Iterations	121	264	399	474
$\varepsilon = 10^{-3}$	Solve time	7·45	128·00	1506·90	3400·12

system we are actually solving in practice are not symmetric about the origin but rather lie in precisely the intervals discussed above.

One important question is how to choose the convergence tolerance for each iterative method. Here each algorithm is considered converged when

$$\frac{\|\mathbf{r}_k\|}{\|\mathbf{r}_0\|} \leqslant \varepsilon \tag{43}$$

for a particular tolerance ε. For CR, $\varepsilon = 10^{-3}$. In the pressure correction case both an outer (ε_O) and an inner (ε_I) value must be set. Here we have chosen $\varepsilon_O = 10^{-3}$ and $\varepsilon_I = 10^{-4}$, reflecting the fact that we expect to have to solve the inner iteration more accurately in order to retain convergence in the outer loop (e.g. Reference 25). In comparison with an 'exact' solution (computed via an uncompetitive direct method), these tolerances ensure that both CR and PC

give solutions which, when normalized with respect to the largest velocity and pressure values, are correct to three decimal places. If ε_I is adjusted to 10^{-3}, this is no longer true: PC gives solutions correct to only two decimal places. Furthermore, accuracy is also lost by choosing a fixed number of iterations for the inner velocity system. For the four largest grids the PC method was rerun restricting the maximum number of possible inner iterations at each step to 20 and the results are listed in Table III. Note that the number of outer iterations required rises as strict inner convergence is lost. In addition, and of more fundamental concern, the overall accuracy of the solution is affected. For this specific example the solution is accurate to two decimal places in the 8×8 case and only one decimal place in the 16×16 case. For the larger grids no decimal places of accuracy were achieved. Hence, although the asymptotic work estimates calculated for PC in Section 5 would be much improved by fixing the number of inner iterations, it is clear that the size of this fixed number would still have to grow in some way with the size of the problem in order to retain accuracy of the solution. Indeed, for larger problems it is possible that the tolerance ratio $\varepsilon_O/\varepsilon_I = 10$ would be insufficient to obtain accurate solutions. Unlike in the case of nested stationary iterations (such as relaxation), here increasing the inner solution accuracy as the outer solution converges is not likely to be successful.

Although we are far from the asymptotic limit with such sequences of problems, it is possible to see some trends emerging. The number of outer iterations in PC does indeed seem to tend to a constant as the grid size increases, more noticeably in the two-dimensional case. For all problems, however, the one-level indefinite CR solver outperforms the more traditional two-level pressure correction approach. This is surprising in view of the analysis in the previous section.

Figures 5 and 6 show log–log plots of solve time against the number of pressure unknowns, N_P in the two- and three-dimensional cases respectively. The slopes of the lines for PC and CR can be seen to be very similar. Estimating the slopes numerically indicates that in our practical calculations both methods behave approximately like $cN_P^{3/2}$ in both two and three dimensions. The better overall performance of CR stems from the lower value of the constant c: in two dimensions the constant is approximately 0·027 for PC and 0·009 for CR and in three dimensions

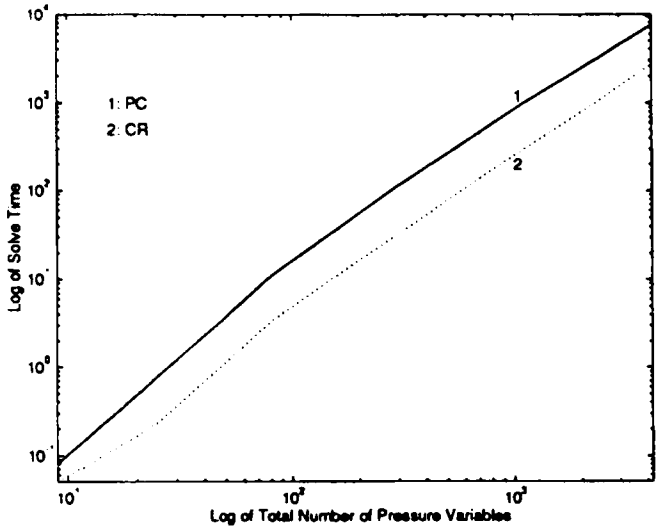

Figure 5. Solve time versus N_P in 2D

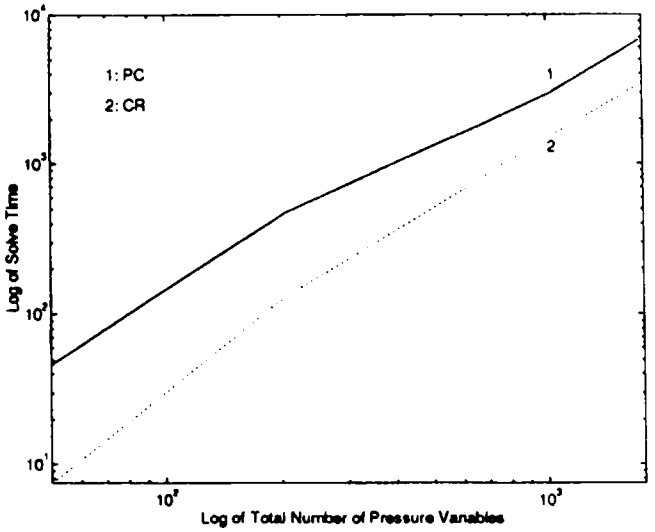

Figure 6. Solve time versus N_P in 3D

$c \simeq 0{\cdot}09$ for PC and 0·045 for CR. It is difficult to extrapolate from these results, since these constants are affected by many issues of implementation and machine architecture. In particular, the actual number of outer PC iterations (which contributes to the constant for this method), though independent of the mesh size for all stable element pairs, will be different for different problems.[16] It is, however, clear that there is at least a range of problems for which the often neglected CR method can be very competitive in practice.

7. CONCLUSIONS

The preconditioned conjugate residual method is an attractive way of solving large Stokes-like linear systems. Here we have applied it to the full Navier–Stokes and continuity equations by using a Lagrange–Galerkin finite element formulation. Although the asymptotic work estimates of Section 5 suggest that the conjugate residual method is theoretically less efficient than a traditional two-level pressure correction approach, this is not confirmed by the numerical experiments of Section 6. These show that in practice the growth rate of the amount of work involved in both methods is similar, with a lower constant in the CR case leading to faster solution times. In addition, these experiments show that care must be taken if an attempt is made to improve the performance of the pressure correction method by using a constant number of steps for the inner iteration, since accuracy of the solution will be lost. This is an important and unexpected point: prescribing a fixed number of inner iterations limits the level of accuracy which can possibly be achieved.

ACKNOWLEDGEMENT

This work was supported by SERC and Nuclear Electric plc. under grant GR/G11309.

APPENDIX

Consider applying the conjugate gradient algorithm to a system

$$\mathbf{A}\mathbf{x} = \mathbf{b}, \tag{44}$$

where the preconditioned coefficient matrix $\mathbf{A}$ is symmetric and positive definite. To derive an error bound, define the A-norm of a vector $\mathbf{v}$ by

$$\|\mathbf{v}\|_{\mathbf{A}} = (\mathbf{v}^{\mathrm{T}}\mathbf{A}\mathbf{v})^{1/2}. \tag{45}$$

The conjugate gradient algorithm generates a sequence of iterates

$$\mathbf{x}_k \in S = \mathbf{x}_0 + \operatorname{span}\{\mathbf{r}_0, \mathbf{A}\mathbf{r}_0, \mathbf{A}^2\mathbf{r}_0, \ldots, \mathbf{A}^{k-1}\mathbf{r}_0\}, \tag{46}$$

where $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$ is the initial residual. The kth residual can thus be written as

$$\mathbf{r}_k = \hat{P}_k(\mathbf{A})\mathbf{r}_0, \tag{47}$$

where $\hat{P}_k(\mathbf{A})$ is a polynomial of degree k with constant term one, i.e. $\hat{P}_k(\mathbf{A}) \in \Pi_k^1$. Since the conjugate gradient method constructs $\mathbf{x}_k$ specifically to minimize a certain functional over all $\mathbf{x} \in S$, $\hat{P}_k(\mathbf{A})$ is in fact the polynomial which minimizes $\mathbf{r}_k$ over all polynomials in Π_k^1; hence

$$\|\mathbf{e}_k\|_{\mathbf{A}} = \min_{P \in \Pi_k^1} \|P(\mathbf{A})\mathbf{r}_0\|_{\mathbf{A}^{-1}}. \tag{48}$$

Expanding the residual in terms of orthonormal eigenvectors of $\mathbf{A}$ leads to an error estimate independent of $\mathbf{r}_0$, namely

$$\|\mathbf{e}_k\|_{\mathbf{A}} \leqslant \min_{P \in \Pi_k^1} \max_i |P(\lambda_i)| \, \|\mathbf{e}_0\|_{\mathbf{A}}, \tag{49}$$

where $\{\lambda_i, i = 1, \ldots, N\}$ are the eigenvalues of $\mathbf{A}$. Full details of this analysis can be found in Reference 21.

It is immediately clear from (49) that the rate of conjugate gradient convergence depends crucially on the distribution of the eigenvalues of $\mathbf{A}$. Details of the exact nature of this dependence are given in References 21 and 26–28. A simple bound can, however, be derived based on the extreme eigenvalues, a and b say. Using this interval, the appropriate polynomial in (49) is a shifted Tchebyshev polynomial which leads to the estimate

$$\|\mathbf{e}_k\|_{\mathbf{A}} \leqslant 2\left(\frac{\sqrt{R}-1}{\sqrt{R}+1}\right)^k \|\mathbf{e}_0\|_{\mathbf{A}}, \tag{50}$$

where $R = b/a$. The average reduction factor as defined by (19) is thus (see Reference 29, p. 187).

$$\sigma = 2^{1/k}\left(\frac{\sqrt{R}-1}{\sqrt{R}+1}\right) \simeq 2^{1/k}\left(1 - \frac{2}{\sqrt{R}}\right). \tag{51}$$

Note that in the symmetric positive definite case the ratio R is the condition number of the matrix. If this condition number is independent of h, so is the resulting reduction factor. For the velocity matrix of PC in Section 5 the eigenvalues lie in the interval $(c_1 h^2, c_2)$ for some real constants c_1 and c_2, so $R = O(h^{-2})$ and for large k we can write

$$\sigma = 2^{1/k}(1 - 2h). \tag{52}$$

Analysis of the preconditioned conjugate residual method can be carried out in a similar way. In this case, if we assume that A is symmetric and indefinite with a spectrum which is symmetric about the origin and contained in

$$(-b, -a) \cup (a, b) \tag{53}$$

for some $a, b > 0$, we can again use shifted Tchebyshev polynomials to obtain a result analogous to (50), namely

$$\| \mathbf{e}_k \|_{\mathbf{A}^2} \leqslant 2\left(\frac{R-1}{R+1}\right)^k \| \mathbf{e}_0 \|_{\mathbf{A}^2}. \tag{54}$$

The related reduction factor is thus

$$\sigma = 2^{1/k}\left(1 - \frac{2}{R}\right), \tag{55}$$

and in the specific case where the eigenvalues lie in $(-c_4, -c_3h^2) \cup (c_3h^2, c_4)$,

$$\sigma = 2^{1/k}(1 - 2h^2). \tag{56}$$

When the spectrum is not symmetric about the origin, the corresponding analysis is considerably more intricate.[23]

REFERENCES

1. M. D. Gunzburger, *Finite Element Methods for Viscous Incompressible Flows*, Academic, London, 1989.
2. P. Chin, E. F. D'Azevedo, P. A. Forsyth and W.-P. Tang, 'Preconditioned conjugate gradient methods for the incompressible Navier–Stokes equations', *Int. j. numer. methods fluids*, **15**, 273–295 (1992).
3. O. Pironneau, *Finite Element Methods for Fluids*, Masson, Paris, 1989.
4. K. W. Morton, 'Lagrange–Galerkin and characteristic–Galerkin methods and their application', *Oxford University Rep. 90/13*, 1990.
5. K. W. Morton and A. Priestley, 'On characteristic and Lagrange–Galerkin methods', D. F. Griffiths and G. A. Watson (eds), *Pitman Research Notes in Mathematics Series*, Longman, London, 1986.
6. K. W. Morton, A. Priestley and E. E. Süli, 'Stability of the Lagrange–Galerkin method with non-exact integration', *RAIRO M² AN*, **22**, 625–653 (1988).
7. E. E. Süli, 'Convergence and nonlinear stability of the Lagrange–Galerkin method for the Navier–Stokes equations', *Numer. Math.*, **53**, 459–483 (1988).
8. S. F. Ashby, T. A. Manteuffel and P. E. Saylor, 'A taxonomy for conjugate gradient methods', *SIAM J. Numer. Anal.*, **27**, 1542–1568 (1990).
9. P. Concus, G. H. Golub and D. P. O'Leary, 'A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations', J. Bunch and D. J. Rose (eds), *Sparse Matrix Computations*, Academic, New York, 1976.
10. R. Verfürth, 'A combined conjugate gradient–multigrid algorithm for the numerical solution of the Stokes problem', *IMA J. Numer. Anal.*, **4**, 441–455 (1984).
11. J. Bramble and J. Pasciak, 'A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems', *Math. Comput.*, **50**, 1–17 (1988).
12. 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).
13. 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).
14. J. Atanga and D. Silvester, 'Iterative methods for stabilized mixed velocity–pressure finite elements', *Int. j. numer. method fluids*, **14**, 71–81 (1992).
15. D. M. Young and K. C. Jea, 'Generalized conjugate gradient acceleration of nonsymmetrizable iterative methods', *Linear Algebra Appl.*, **34**, 159–194 (1980).
16. A. J. Wathen and D. J. Silvester, 'Fast iterative solution of stabilised Stokes systems. Part I: Using simple diagonal preconditioners', *SIAM J. Numer. Anal.*, **30**, 630–649 (1993).
17. A. J. Wathen and D. J. Silvester, 'The convergence of Krylov-space methods for the Stokes problem in computational fluid dynamics', *Proc. Copper Mountain Conf. on Iterative Methods*, SIAM, Philadelphia, PA, 1990.

18. T. J. R. Hughes, I. Levit and J. Winget, 'An element-by-element solution algorithm for problems of structural and solid mechanics', *Comput. Methods Appl. Mech. Eng.*, **36**, 241–254 (1983).
19. D. J. Silvester and A. J. Wathen, 'Fast iterative solution of stabilised Stokes systems. Part II: Using general block preconditioners', *SIAM J. Numer. Anal.* (1994).
20. R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.
21. O. Axelsson and V. A. Barker, *Finite Element Solution of Boundary Value Problems: Theory and Computation*, Academic, New York, 1984.
22. I. Gustafsson, 'A class of first order factorisation methods', *BIT*, **18**, 142–156 (1978).
23. A. J. Wathen, B. Fischer and D. J. Silvester, 'On convergence of conjugate residuals for the Stokes problem', *Numer. Math.*, submitted.
24. S. Hickmott, M. Rabbitt and R. M. Smith, *FEAT Reference Guide*, Nuclear Electric plc., Berkeley Nuclear Laboratories, Berkeley, 1993.
25. G. H. Golub and M. Overton, 'The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems', *Numer. Math.*, **53**, 571–594 (1988).
26. A. Greenbaum, 'Comparison of splittings used with the conjugate gradient algorithm', *Numer. Math.*, **33**, 181–193 (1979).
27. A. Ramage, 'Eigenvalue clustering and conjugate gradient convergence for elliptic partial differential equations', R. Beauwens and P. de Groen (eds), *Iterative Methods in Linear Algebra*, Elsevier/North-Holland, Amsterdam, 1992, pp. 593–601.
28. A. Van der Sluis and H. A. Van der Vorst, 'The rate of convergence of conjugate gradients', *Numer. Math.*, **48**, 543–560 (1986).
29. D. G. Luenberger, *Introduction to Linear and Nonlinear Programming*, Addison-Wesley, Reading, MA, 1973.