NUMERICAL LINEAR ALGEBRA WITH APPLICATIONS
Numer. Linear Algebra Appl. (2010)
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/nla.693

Block-triangular preconditioners for PDE-constrained optimization

Tyrone Rees[1] and Martin Stoll[2,*,†]

[1]*Oxford University Mathematical Institute, 24-29 St Giles', Oxford OX1 3LB, U.K.*
[2]*Oxford Centre for Collaborative Applied Mathematics, Mathematical Institute, 24-29 St Giles', Oxford OX1 3LB, U.K.*

SUMMARY

In this paper we investigate the possibility of using a block-triangular preconditioner for saddle point problems arising in PDE-constrained optimization. In particular, we focus on a conjugate gradient-type method introduced by Bramble and Pasciak that uses self-adjointness of the preconditioned system in a non-standard inner product. We show when the Chebyshev semi-iteration is used as a preconditioner for the relevant matrix blocks involving the finite element mass matrix that the main drawback of the Bramble–Pasciak method—the appropriate scaling of the preconditioners—is easily overcome. We present an eigenvalue analysis for the block-triangular preconditioners that gives convergence bounds in the non-standard inner product and illustrates their competitiveness on a number of computed examples. Copyright © 2010 John Wiley & Sons, Ltd.

Received 24 March 2009; Revised 13 November 2009; Accepted 22 November 2009

KEY WORDS: saddle point problems; linear systems; Krylov subspaces; preconditioning; PDE-constrained optimization

1. INTRODUCTION

At the heart of many applications lies the solution of a linear system in saddle point form

$$\underbrace{\begin{bmatrix} A & B^{\mathrm{T}} \\ B & -C \end{bmatrix}}_{\mathcal{A}} x = b \tag{1}$$

*Correspondence to: Martin Stoll, Oxford Centre for Collaborative Applied Mathematics, Mathematical Institute, 24-29 St Giles', Oxford OX1 3LB, U.K.
†E-mail: martin.stoll@maths.ox.ac.uk

Contract/grant sponsor: King Abdullah University of Science and Technology (KAUST); contract/grant number: KUK-C1-013-04

where $A \in \mathbb{R}^{n,n}$, $B \in \mathbb{R}^{m,n}$. The properties of the blocks of $\mathcal{A}$ vary with the underlying application—for example, when looking at saddle point problems arising in the mixed finite element treatment of Stokes flow problems the matrix A is symmetric and positive definite and the matrix C positive semi-definite—often $C = 0$. A comprehensive survey describing methods for solving saddle point problems—along with a list of applications that lead to such systems—is given by Benzi *et al.* in [1]. In this paper we focus on the situation where the matrix $\mathcal{A}$ has a symmetric and positive-definite block A and $C = 0$, and in particular on an important class of saddle point problems arising in optimization problems with PDE constraints. It is well-known that under the assumption that $B \in \mathbb{R}^{m,n}$ is of full rank the invertibility of $\mathcal{A}$ is guaranteed. The resulting system is symmetric and indefinite and hence MINRES [2] is applicable as an iterative solution algorithm. Unfortunately, the system matrix $\mathcal{A}$ does not allow for an iterative process to be effectively applied without using a preconditioner. Depending on the iterative method of choice one needs to design the preconditioner $\mathcal{P}$ in such a way that the preconditioned matrix $\widehat{\mathcal{A}} = \mathcal{P}^{-1}\mathcal{A}$ satisfies any required properties needed for the numerical scheme—for MINRES this means that the preconditioner has to be symmetric and positive definite in order to preserve symmetry in the preconditioned system. In this paper we consider using MINRES with a block-diagonal preconditioner,

$$\mathcal{P} = \begin{bmatrix} A_0 & 0 \\ 0 & S_0 \end{bmatrix}. \tag{2}$$

We will discuss preconditioners of this type constructed for problems from PDE-constrained optimization in more detail in Section 4.

The most popular Krylov subspace iterative method for symmetric matrices is the conjugate gradient (CG) method proposed by Hestenes and Stiefel in [3]. This method can, in its basic form, only be applied to systems $\mathcal{A}$ that are symmetric and positive definite, and in addition needs the preconditioner $\mathcal{P}$ also to be symmetric [4]. Various modifications to CG were proposed to make it applicable for systems of the form (1) (see [5] for a comprehensive survey). One of the most popular variants is the so-called Bramble–Pasciak CG method. This method uses a block-triangular preconditioner and a non-standard inner product in which the preconditioned matrix is symmetric and, under certain conditions, positive definite. In Section 3, we discuss this method in more detail. The main emphasis in Section 3 is to show that the Bramble–Pasciak CG method is well suited for saddle point problems arising in PDE-constrained optimization—we introduce these problems in Section 2.

Finally, in Section 5 we compare numerical results of MINRES with block-diagonal preconditioner to the Bramble–Pasciak CG with block-triangular preconditioner.

2. THE OPTIMAL CONTROL PROBLEM

One of the application areas that leads to linear systems in saddle point form is the field of PDE-constrained optimization (see [6, 7] for introductions to the area). The main aim here is to minimize a functional $J(y, u)$ subject to a PDE as the constraint. In our case, the functional to be minimized over a domain $\Omega \in \mathbb{R}^d$ with $d = 2, 3$ is given by

$$J(y, u) := \frac{1}{2}\|y - \bar{y}\|^2_{L^2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L^2(\Omega)}. \tag{3}$$

DOI: 10.1002/nla

The state y and the control u are linked via a partial differential equation—we consider the Poisson equation here

$$-\triangle y = f + u \quad \text{in } \Omega, \tag{4}$$

where f and $\bar{y}$ are some given functions, and β is a regularization constant. Note that although our theory and computations as presented here are based on this equation the ideas presented here can be applied to any elliptic PDE. In some applications, the control can be bounded by so-called *box constraints*

$$u_a(x) \leqslant u(x) \leqslant u_b(x) \quad \text{a.e in } \Omega. \tag{5}$$

Note that we will neglect the box constraints for the remainder of this paper and only talk about the unconstrained problem. Nevertheless, we want to emphasize that the techniques presented in this paper are useful when solving problems with box constraints (see [8]). We can summarize the setup in the following optimization problem:

$$\begin{aligned} \min \frac{1}{2}\|y-\bar{y}\|^2_{L^2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L^2(\Omega)} \quad & \text{s.t.} \\ -\triangle y = f + u \quad & \text{in } \Omega \\ y = \bar{y} \quad & \text{on } \Gamma, \end{aligned} \tag{6}$$

where Γ represents the boundary of Ω. The discretization of the optimization problem (6) can be done via the finite element method [9]. As a result we obtain a discrete optimization problem given by

$$\begin{aligned} \min \frac{1}{2}(y_h - \bar{y}_h)^{\mathrm{T}} M (y_h - \bar{y}_h) + \frac{\beta}{2} {u_h}^{\mathrm{T}} M u_h \quad & \text{s.t.} \\ K y_h = M f_h + M u_h & \end{aligned} \tag{7}$$

where $M \in \mathbb{R}^{n,n}$ is a mass matrix and $K \in \mathbb{R}^{n,n}$ represents the stiffness matrix of the Poisson equation. With a Lagrange multiplier approach, we obtain the following Lagrange function

$$L(y_h, u_h, \lambda_h) = \frac{1}{2}(y_h - \bar{y}_h)^{\mathrm{T}} M (y_h - \bar{y}_h) + \frac{\beta}{2} {u_h}^{\mathrm{T}} M u_h - {\lambda_h}^{\mathrm{T}} (K y_h - M f_h - M u_h). \tag{8}$$

Using the standard techniques for (8), we see that the optimal solution $({y_h}^*, {u_h}^*, {\lambda_h}^*)$ of the system (7) satisfies the linear system

$$\underbrace{\begin{bmatrix} M & 0 & -K^{\mathrm{T}} \\ 0 & \beta M & M \\ -K & M & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} {y_h}^* \\ {u_h}^* \\ {\lambda_h}^* \end{bmatrix} = \begin{bmatrix} M\bar{y}_h \\ 0 \\ -M f_h \end{bmatrix} \tag{9}$$

where the saddle point matrix $\mathcal{A}$ is now a symmetric matrix with a 3×3 block structure. The system (7) needs to be solved efficiently and in the remainder of this paper we discuss methods that are well suited for this purpose. For more details on the derivation of (9), we refer to [10]. Note that for notational convenience we will switch between the 3×3 and 2×2 block structure of

DOI: 10.1002/nla

the saddle point matrix in (1). Typically, we will introduce general statements in the 2×2 block form and properties specific to the matrix from the optimal control problem will be explained in the 3×3 block structure.

3. BRAMBLE–PASCIAK CG

The Bramble–Pasciak CG method was introduced in [11] as a solver for saddle point problems which arise, for example, in the simulation of fluid flow problems or Lagrange multiplier approaches. The method is widely used in the finite element community [12–21] (Langer and Kunh; unpublished). We first introduce the method based on the 2×2 saddle point structure for $\mathcal{A}$ (cf. (1)). The Bramble–Pasciak CG is based on the application of a block-triangular preconditioner of the form

$$\mathcal{P}=\begin{bmatrix} A_0 & 0 \\ B & -S_0 \end{bmatrix}, \tag{10}$$

where A_0 is an approximation to the $(1,1)$-block of $\mathcal{A}$ and S_0 approximates the corresponding Schur-complement $BA^{-1}B^{\mathrm{T}}$. The preconditioning blocks A_0 and S_0 are generally chosen such that a good eigenvalue distribution is achieved. In practice, a good clustering of the eigenvalues will typically ensure fast convergence for the problem considered here (see [10, 22]). The application of the block-triangular preconditioner is only slightly more expensive than the block-diagonal preconditioner (2) as it requires one more multiplication with B [11, 23, 24]. When applying the left preconditioner $\mathcal{P}$ to the system matrix $\mathcal{A}$ the preconditioned matrix $\widehat{\mathcal{A}}=\mathcal{P}^{-1}\mathcal{A}$ will be non-symmetric. On first sight the question that arises is whether such a preconditioner might be useful in practice as the symmetric system is transformed into a non-symmetric one. Amazingly, Bramble and Pasciak showed that the matrix $\widehat{\mathcal{A}}$ is symmetric and also positive definite in $\langle u, v\rangle_{\mathcal{H}}=u^{\mathrm{T}}\mathcal{H}v$ with

$$\mathcal{H}=\begin{bmatrix} A-A_0 & 0 \\ 0 & S_0 \end{bmatrix} \tag{11}$$

whenever $\mathcal{H}$ defines an inner product. To see that the preconditioned matrix $\widehat{\mathcal{A}}$ is symmetric and positive definite in $\mathcal{H}$ one has to show that

$$\langle\widehat{\mathcal{A}}x, y\rangle_{\mathcal{H}}=\langle x, \widehat{\mathcal{A}}y\rangle_{\mathcal{H}} \Leftrightarrow \widehat{\mathcal{A}}^{\mathrm{T}}\mathcal{H}=\mathcal{H}\widehat{\mathcal{A}} \quad \forall x, y\in\mathbb{R}^n$$

holds and by using a splitting technique we can prove the positivity condition

$$\langle\widehat{\mathcal{A}}x, x\rangle_{\mathcal{H}}>0 \Leftrightarrow \mathcal{H}\widehat{\mathcal{A}}>0. \tag{12}$$

The details can be found in [11, 14, 23, 25, 26]. Note that it is also possible to consider right preconditioning but as the difference is typically marginal we employ left preconditioning and refer to [26–28] for more information. The implementation of a CG method based on this is given in Algorithm 1 (see [23] for details).

One of the questions that arises when implementing the Bramble–Pasciak CG is whether we can use preconditioners A_0 and S_0 where these matrices are never given explicitly. This is the case, for example, with multigrid-based preconditioners, where we apply a linear process that could be written as the inverse of a matrix, but we do not generally have that matrix. It was shown in [11] that even in this case the method is applicable as only the action of the inverse of

DOI: 10.1002/nla

Given $x_0 = 0$, set $r_0 = \mathcal{P}^{-1}(b - \mathcal{A}x_0)$ and $p_0 = r_0$
for $k = 0, 1, \ldots$ **do**
  $\alpha = \frac{\langle r_k, r_k \rangle_{\mathcal{H}}}{\langle \mathcal{P}^{-1}\mathcal{A}p_k, p_k \rangle_{\mathcal{H}}}$
  $x_{k+1} = x_k + \alpha p_k$
  $r_{k+1} = r_k - \alpha \mathcal{P}^{-1}\mathcal{A}p_k$
  $\beta = \frac{\langle r_{k+1}, r_{k+1} \rangle_{\mathcal{H}}}{\langle r_k, r_k \rangle_{\mathcal{H}}}$
  $p_{k+1} = r_{k+1} + \beta p_k$
end for

Algorithm 1: Bramble and Pasciak CG

A_0 is needed when evaluating the inner products with $\mathcal{H}$—i.e. the application of a fixed number of multigrid cycles is sufficient. Elman showed in [13] that the same is true for the Schur-complement preconditioner S_0. The implementation of the Bramble–Pasciak is subtle and details for general problems are given in [13] and in the case of PDE-constrained optimization problems we refer to [29]. We will demonstrate this only on one of the quantities involved and refer the interested reader to [11, 13, 23] for details of implementation. The quantity to compute is given by $\langle \mathcal{P}^{-1}\mathcal{A}r^{(k+1)}, p^{(k)} \rangle_{\mathcal{H}}$ which reduces to evaluating $\mathcal{H}\mathcal{P}^{-1}\mathcal{A}r^{(k+1)}$ without using A_0 and S_0. In more detail, using $Ar^{(k+1)} = [(\hat{r}_1^{(k+1)})^{\mathrm{T}}(\hat{r}_2^{(k+1)})^{\mathrm{T}}]^{\mathrm{T}}$ we get

$$\mathcal{H}\mathcal{P}^{-1}\mathcal{A}r^{(k+1)} = \begin{bmatrix} A - A_0 & 0 \\ 0 & S_0 \end{bmatrix} \begin{bmatrix} A_0^{-1}\hat{r}_1^{(k+1)} \\ S_0^{-1}(BA_0^{-1}\hat{r}_1^{(k+1)} - \hat{r}_2^{(k+1)}) \end{bmatrix}$$

$$= \begin{bmatrix} AA_0^{-1}\hat{r}_1^{(k+1)} - \hat{r}_1^{(k+1)} \\ BA_0^{-1}\hat{r}_1^{(k+1)} - \hat{r}_2^{(k+1)} \end{bmatrix}$$

which is seen to involve only A_0^{-1}, not A_0. It can also be shown that for the evaluation of the inner product with $\mathcal{H}$ the Schur-complement preconditioner S_0 does not need to be evaluated [23]. In fact the preconditioner $\mathcal{P}$ has to be applied only once per iteration and in order to evaluate the inner product with $\mathcal{H}$ no extra multiplications with blocks of the saddle point matrix are required (see [13] for details). Hence, the extra cost of the Bramble–Pasciak method compared with MINRES with block-diagonal preconditioning is given by having to perform one additional multiplication with the B block of the matrix [13].

So far we have not talked about the drawbacks of the Bramble–Pasciak CG. For the algorithm to work we require $A - A_0$ to be positive definite—recall that $\mathcal{H}$ needs to define an inner product—which can only be achieved when A_0 is scaled appropriately. The scaling typically involves solving an eigenvalue estimation problem for the matrix $A_0^{-1}A$ and can only be avoided when a good knowledge of the eigenvalues of $A_0^{-1}A$ is at hand.

We now return to the 3×3 system derived in Section 2, i.e.

$$\underbrace{\begin{bmatrix} M & 0 & -K^{\mathrm{T}} \\ 0 & \beta M & M \\ -K & M & 0 \end{bmatrix}}_{\mathcal{A}} \begin{bmatrix} y_h{}^* \\ u_h{}^* \\ \lambda_h{}^* \end{bmatrix} = \begin{bmatrix} M\bar{y}_h \\ 0 \\ -Mf_h \end{bmatrix}.$$

DOI: 10.1002/nla

The preconditioner in this case is given by

$$\mathcal{P}=\begin{bmatrix} A_0 & 0 & 0 \\ 0 & A_1 & 0 \\ -K & M & -S_0 \end{bmatrix} \tag{13}$$

and the inner product is defined by

$$\mathcal{H}=\begin{bmatrix} M-A_0 & 0 & 0 \\ 0 & \beta M-A_1 & 0 \\ 0 & 0 & S_0 \end{bmatrix}. \tag{14}$$

In this case, the condition for the applicability of the Bramble–Pasciak CG is the positivity of $M-A_0$ and $\beta M-A_1$, along with the positive definiteness of S_0. If we can find scalings for A_0 and A_1 without having to approximate an eigenvalue of $A_0^{-1}M$ or $\beta A_1^{-1}M$ the Bramble–Pasciak CG represents a well-suited method for PDE-constrained optimization problems: we address this question in Section 3.1.

The Schur complement of the saddle point problem (9) is given by

$$\frac{1}{\beta}M+KM^{-1}K^{\mathrm{T}}. \tag{15}$$

Here, we want S_0 to represent a good approximation to the Schur complement. A strategy used by Rees *et al.* [10] is for S_0 to be an approximation to $KM^{-1}K^{\mathrm{T}}$ which means that the term $\frac{1}{\beta}M$ in (15) is neglected. The Schur-complement preconditioner is now given by $S_0=\tilde{K}M^{-1}\tilde{K}^{\mathrm{T}}$, where $\tilde{K}$ represents a fixed number of algebraic multigrid V-cycles using the `HSL` package `HSL_MI20` [30].

It is well known that the classical CG method [3] minimizes the $\mathcal{A}$-norm of the error, i.e. $\|e_k\|_{\mathcal{A}}$ where $e_k=x-x_k$, even in its preconditioned version, PCG; the method of Bramble and Pasciak with the non-standard inner product defined by $\mathcal{H}$ thus minimizes the error in the norm defined by $\mathcal{H}\widehat{\mathcal{A}}$, i.e. $\|e_k\|_{\mathcal{H}\widehat{\mathcal{A}}}$.

3.1. Chebyshev semi-iteration

As we mentioned earlier the downside of the Bramble–Pasciak CG is that the preconditioners A_0 and A_1 need to be scaled to guarantee that $\mathcal{H}$ defines an inner product. In the case of the optimal control problem (7) we have the blocks M and βM, a mass matrix and a scaled mass matrix. An efficient preconditioner for these systems is given when a fixed number of steps of the Chebyshev-semi-iteration (described below) can be applied, as shown in [31].

In [32, 33] Wathen provided eigenvalue bounds for mass matrices coming from different two- or three-dimensional (3D) finite elements. We are going to use these bounds to derive, first, bounds for the convergence of relaxed Jacobi (and the optimal relaxation parameter) and, second, bounds for the eigenvalues of $A_0^{-1}M$, where A_0 now represents the matrix that is used to get the nth Chebyshev semi-iterate approximation (Table I). First, suppose we want to solve the linear system

$$Mx=\hat{b} \tag{16}$$

DOI: 10.1002/nla

Table I. Upper and lower bounds for the eigenvalues of the iteration matrix coming from relaxed Jacobi accelerated by the Chebyshev semi-iteration.

k	Lower bound ($\lambda_{\min}$)	Upper bound ($\lambda_{\max}$)	Lower bound	Upper bound
	(a)		(b)	
1	0.200000000000000	1.800000000000000	0.071428571428571	1.928571428571429
2	0.529411764705882	1.470588235294118	0.242152466367712	1.757847533632288
3	0.753846153846154	1.246153846153846	0.433470861268694	1.566529138731306
4	0.875486381322957	1.124513618677043	0.597147975231673	1.402852024768327
5	0.937560975609756	1.062439024390244	0.720776486124292	1.279223513875708
6	0.968757627532341	1.031242372467659	0.808846507572246	1.191153492427754
7	0.984375953616112	1.015624046383888	0.869897818711463	1.130102181288537
8	0.992187619207471	1.007812380792529	0.911689150016541	1.088310849983459
9	0.996093764901104	1.003906235098896	0.940130893927575	1.059869106072427
10	0.998046876862643	1.001953123137357	0.959435805298037	1.040564194701956
11	0.999023437732830	1.000976562267170	0.972523033141943	1.027476966858060
12	0.999511718779104	1.000488281220896	0.981390172808929	1.018609827191073
13	0.999755859378638	1.000244140621363	0.987396479797611	1.012603520202398
14	0.999877929687954	1.000122070312045	0.991464472578415	1.008535527421550
15	0.999938964843807	1.000061035156194	0.994219521276648	1.005780478723391
16	0.999969482421881	1.000030517578119	0.996085332018970	1.003914667981061
17	0.999984741210937	1.000015258789063	0.997348906791884	1.002651093208068
18	0.999992370605466	1.000007629394531	0.998204627482791	1.001795372516598
19	0.999996185302733	1.000003814697268	0.998784139008908	1.001215860991333
20	0.999998092651366	1.000001907348635	0.999176595616630	1.000823404383702

Upper and lower bounds for the eigenvalues of the iteration matrix coming from relaxed Jacobi accelerated by the Chebyshev semi-iteration. (a) Upper and lower bounds for $\lambda(A_0^{-1}M)$, where A_0 is given by k steps of the Chebyshev semi-iteration (without scaling) for square Q1 elements (in 2D) and (b) Upper and lower bounds for $\lambda(A_0^{-1}M)$, where A_0 is given by k steps of the Chebyshev semi-iteration (without scaling) for square Q1 elements (in 3D).

where M is the mass matrix introduced in Section 2 and $\hat{b}$ is an arbitrary right-hand side. The iterates of the relaxed Jacobi method [28] are given by

$$x^{(m+1)} = (I - \omega D^{-1}M)x^{(m)} + \omega D^{-1}\hat{b}$$

here $D = \mathrm{diag}(M)$. In [32] Wathen gives bounds for the matrix $D^{-1}M$ for different finite element discretizations. These bounds are easily obtained, even for irregular meshes—if D_i is the diagonal of the ith element matrix E_i, then the smallest and largest eigenvalues are given by the smallest and largest eigenvalues of $D_i^{-1}E_i$ over all i. This is just an $\mathcal{O}(n)$ calculation, and only needs to be done once for each mesh. We consider square Q1 elements here, and in this case

$$\lambda(D^{-1}M) \in [\tfrac{1}{4}, \tfrac{9}{4}].$$

By using this it is easy to see that the eigenvalues of the iteration matrix, $I - \omega D^{-1}M$ satisfy

$$\lambda(I - \omega D^{-1}M) \in \left[1 - \frac{9\omega}{4}, 1 - \frac{\omega}{4}\right].$$

The optimal relaxation parameter ω is computed such that the interval is symmetric, i.e. for $\omega = \frac{4}{5}$ we get $\lambda(I - \omega D^{-1}M) \in [-\frac{4}{5}, \frac{4}{5}]$ (see [33]). In summary, $\omega = \frac{4}{5}$ is the optimal relaxation parameter

DOI: 10.1002/nla

for square Q1 elements in 2D. In 3D for the equivalent elements we have $\lambda(D^{-1}M) \in [\frac{1}{8}, \frac{27}{8}]$ and hence the optimal relaxation parameter is $\omega = \frac{4}{7}$ with $\lambda(I - \omega D^{-1}M) \in [-\frac{13}{14}, \frac{13}{14}]$.

If we now label the iteration matrix as

$$S = I - \omega D^{-1} M,$$

then in a polynomial iterative method (see [34]) we take a linear combination of iterates $x^{(k)}$ to produce $z^{(k)} = \sum_{i=0}^{k} \alpha_i x^{(i)}$ which is, ideally, a better approximation to the solution x. We can think of the terms α_i as being coefficients of a polynomial of degree k, $p_k(z)$. Note that we require $p_k(1) = 1$ for the iterates to satisfy $z^{(k)} = x$ for all k if $x^{(0)} = x$. We can write the error as

$$x - z^{(k)} = p_k(S)(x - x^{(0)}).$$

If we start with an initial guess of zero, then this reduces to

$$x - z^{(k)} = p_k(S)x,$$

and, rearranging, we can write the kth iterate as

$$z^{(k)} = (I - p_k(S))x.$$

Now, from (16), we can write

$$z^{(k)} = (I - p_k(S))M^{-1}\hat{b}.$$

If we precondition M with the kth polynomial iterate, then we are actually preconditioning with the matrix A_0 which is the matrix with inverse

$$A_0^{-1} = (I - p_k(S))M^{-1}.$$

For the application of the Bramble–Pasciak CG we need good eigenvalue bounds for the matrix $A_0^{-1}M$. We now have $A_0^{-1}M$ given by

$$A_0^{-1}M = I - p_k(S).$$

We choose the underlying polynomial $p_k(z)$ so that it is small on the eigenvalues and $p_k(1)=1$. Note that this technique is also known as the polynomial acceleration technique (see [35]). If we only know the extremal eigenvalues but nothing about the distribution of the remaining eigenvalues, then the best choice of polynomial is the scaled and shifted Chebyshev polynomials. If $\lambda(S) \in [-\rho, \rho]$,

$$p_k(S) = \widehat{T}_k(S) = \frac{T_k(S/\rho)}{T_k(1/\rho)},$$

where T_k are the Chebyshev polynomials, i.e. $T_0(t) = 1, T_1(t) = t, T_2(t) = t^2 - 1/2, T_3(t) = t^3 - 3/4t$, etc. This then gives the Chebyshev semi-iterative method [36, 37]. An efficient way to compute the iterates $z^{(k+1)}$ of the Chebyshev semi-iterative method is given by the three-term recurrence relation

$$z^{(k+1)} = \vartheta_{k+1}(Sz^{(k)} + g - z^{(k-1)}) + z^{(k-1)}, \tag{17}$$

where $\vartheta_{k+1} = \{T_k(1/\rho)\}/\rho T_{k+1}(1/\rho)$ and $g = \omega D^{-1}\hat{b}$. (see Varga [34, Chapter 5]). The values ϑ_{k+1} can be computed using the bounds for eigenvalues of the iteration matrix and the recursive

DOI: 10.1002/nla

Table II. 2D results for $\beta = 1e-2$.

	Bramble Pasciak		MINRES			Bramble Pasciak		MINRES	
N	its	Time	its	Time	N	its	Time	its	Time
	(a)					(b)			
2	8	0.0110680	10	0.0153780	2	8	0.0117560	10	0.0141400
3	8	0.0196050	11	0.0371250	3	8	0.0137360	10	0.0180900
4	9	0.0395030	12	0.0399930	4	8	0.0252070	10	0.0324230
5	9	0.0920520	14	0.1387090	5	8	0.0773730	10	0.0989570
6	10	0.4437380	16	0.5800720	6	8	0.3600740	10	0.4652580
7	11	1.8411550	18	2.7647310	7	8	1.5217570	10	1.8988780
8	13	10.4573860	20	13.4715910	8	8	6.9457260	12	9.3605340
9	14	42.7101150	23	62.4852580	9	9	31.2425560	12	39.2574310

2D results for $\beta = 1e-2$. (a) tol $= 1e-6$, $\beta = 1e-2$ and one V-cycle and (b) tol $= 1e-6$, $\beta = 1e-2$ and two V-cycles.

definition of the Chebyshev polynomials. We know that $\lambda(S) \in [-\rho, \rho]$ and so the range of $\lambda(\widehat{T}_k(S))$ can be computed as well as $\lambda(A_0^{-1}M)$. These values are tabulated for square Q1 elements in 2D (Table II(a)) and cubic Q1 elements in 3D (Table II(b)) below.

We now go back to the construction of preconditioners A_0 and A_1 that we want to use in the Bramble–Pasciak method based on the presented analysis of the Chebyshev semi-iteration. Recall that in order for $\mathcal{H}$ to define an inner product we need the blocks $M - A_0$ and $\beta M - A_1$ to be positive definite. We only discuss the case $M - A_0 > 0$ as $\beta M - A_1 > 0$ can be seen as a scaled version of this condition. The condition $M - A_0 > 0$ is equivalent to $A_0^{-1}M - I$ being positive definite. In fact, if all the eigenvalues of $A_0^{-1}A$ are larger than one we have positive definiteness of $M - A_0$. Using the bounds presented in Table II(a, b) we have a good knowledge of the minimal eigenvalue $\lambda_{\min}$ of $A_0^{-1}M$ and choosing any parameter γ_0 with $0 < \gamma_0 < \lambda_{\min}$ guarantees the positivity of $M - \gamma_0 A_0$. Analogously, A_1 can be scaled to give the positive definiteness of $\beta M - \gamma_1 A_1 > 0$. This is a key advantage of our approach: the required eigenvalue bounds to ensure positive definiteness of $A_0^{-1}M - I$ are not commonly available when the Bramble–Pasciak method is used, whereas here we have simple *a priori* calculations that give precisely the tight bounds required. For notational convenience, we will only use A_0 and A_1 for the remainder of the paper assuming that they are scaled appropriately.

3.2. *Eigenvalue analysis*

Consider a general saddle point matrix (1),

$$\mathcal{A} = \begin{bmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{bmatrix}.$$

Suppose we precondition $\mathcal{A}$ based on the results of [9, 14] with a block-triangular preconditioner of the form

$$\mathcal{P}_{BT} = \begin{bmatrix} A_0 & 0 \\ B & -S_0 \end{bmatrix},$$

DOI: 10.1002/nla

where A_0 and S_0 are positive-definite approximations of A and $BA^{-1}B^{\mathrm{T}}$ that satisfy

$$1<\delta \leqslant \frac{xAx^{\mathrm{T}}}{xA_0x^{\mathrm{T}}}\leqslant\Delta \tag{18}$$

$$\phi \leqslant \frac{xBA^{-1}B^{\mathrm{T}}x^{\mathrm{T}}}{xS_0x^{\mathrm{T}}}\leqslant\Phi. \tag{19}$$

Note that the condition that $1<\delta$ above is equivalent to the condition that $A-A_0>0$, which we required for $\mathcal{H}$ in (11) to define an inner product. Equation (19) is equivalent to

$$\phi\leqslant\frac{x^{\mathrm{T}}B^{\mathrm{T}}S_0Bx}{xAx^{\mathrm{T}}}\leqslant\Phi. \tag{20}$$

It is well known that at least in the symmetric—or in general the self-adjoint—case, the effectiveness of a preconditioner is dependent on the eigenvalue distribution of the preconditioned system, so we want to find the eigenvalues of the generalized eigenvalue problem $\mathcal{A}u=\lambda\mathcal{P}_{BT}u$. This can be written as

$$Ax+B^{\mathrm{T}}y=\lambda A_0x \tag{21}$$

$$(1-\lambda)Bx=-\lambda S_0y. \tag{22}$$

First, note that all eigenvalues must be real, as $\mathcal{P}_{BT}^{-1}\mathcal{A}$ is self-adjoint in the inner product defined by $\mathcal{H}$. Multiplying (21) on the left by $(1-\lambda)x^{\mathrm{T}}$ and subtracting (22) multiplied on the right by y^{T}, we get

$$(1-\lambda)x^{\mathrm{T}}Ax=(1-\lambda)\lambda x^{\mathrm{T}}A_0x+\lambda y^{\mathrm{T}}S_0y<(1-\lambda)\lambda x^{\mathrm{T}}Ax+\lambda y^{\mathrm{T}}S_0y,$$

the inequality being a consequence of (18). Rearranging, we get that

$$(1-\lambda)^2x^{\mathrm{T}}Ax<\lambda y^{\mathrm{T}}S_0y.$$

Clearly $(1-\lambda)^2\geqslant0$, and as A and S_0 are positive-definite matrices we must have that $\lambda\geqslant0$.

Now, suppose $Bx=0$. Then, from (22), $-\lambda S_0y=0\Rightarrow y=0$, as S_0 is positive definite. Therefore, substituting this into (21) we get $Ax=\lambda A_0x$, which tells us that

$$\delta\leqslant\lambda=\frac{x^{\mathrm{T}}Ax}{x^{\mathrm{T}}A_0x}\leqslant\Delta. \tag{23}$$

Now suppose that $Bx\neq0$. Using (22) to eliminate y from (21) gives

$$\lambda x^{\mathrm{T}}Ax+(\lambda-1)x^{\mathrm{T}}B^{\mathrm{T}}S_0^{-1}Bx=\lambda^2x^{\mathrm{T}}A_0x.$$

We know that A is positive definite, so $x^{\mathrm{T}}Ax>0$, and hence we can write

$$\lambda=\lambda^2\frac{x^{\mathrm{T}}A_0x}{x^{\mathrm{T}}Ax}+(1-\lambda)\frac{x^{\mathrm{T}}B^{\mathrm{T}}S_0^{-1}Bx}{x^{\mathrm{T}}Ax}. \tag{24}$$

DOI: 10.1002/nla

Suppose that $\lambda<1$, so $1-\lambda>0$. Then, using (18) and (20) we get that

$$\frac{\lambda^2}{\Delta}+(1-\lambda)\phi\leqslant\lambda\leqslant\frac{\lambda^2}{\delta}+(1-\lambda)\Phi,$$

and hence

$$\lambda^2-(1+\phi)\Delta\lambda+\phi\Delta\leqslant 0,\quad \lambda^2-(1+\Phi)\delta\lambda+\Phi\delta\geqslant 0.$$

Analysis of the roots of these quadratic equations tells us that we must have

$$\frac{(1+\phi)\Delta-\sqrt{(1+\phi)^2\Delta^2-4\phi\Delta}}{2}\leqslant\lambda\leqslant\frac{(1+\phi)\Delta+\sqrt{(1+\phi)^2\Delta^2-4\phi\Delta}}{2},$$

and

$$\lambda\leqslant\frac{(1+\Phi)\delta-\sqrt{(1+\Phi)^2\delta^2-4\Phi\delta}}{2},\quad \lambda\geqslant\frac{(1+\Phi)\delta+\sqrt{(1+\Phi)^2\delta^2-4\Phi\delta}}{2}.$$

Note, however, that we assumed $\lambda<1$ to get these bounds, and it is easy to see by completing the square under the square root that the two upper bounds are greater than one, so are of no use. Therefore we have shown that if $\lambda<1$, then

$$\frac{(1+\phi)\Delta-\sqrt{(1+\phi)^2\Delta^2-4\phi\Delta}}{2}\leqslant\lambda\leqslant\frac{(1+\Phi)\delta-\sqrt{(1+\Phi)^2\delta^2-4\Phi\delta}}{2}. \tag{25}$$

Now let us return again to (24). If we take the case $\lambda>1$ we can do the same analysis as above. In this case we get the inequality

$$\frac{\lambda^2}{\Delta}+(1-\lambda)\Phi\leqslant\lambda\leqslant\frac{\lambda^2}{\delta}+(1-\lambda)\phi,$$

and proceeding as above we get that if $\lambda>1$, then

$$\frac{(1+\phi)\delta+\sqrt{(1+\phi)^2\delta^2-4\phi\delta}}{2}\leqslant\lambda\leqslant\frac{(1+\Phi)\Delta+\sqrt{(1+\Phi)^2\Delta^2-4\Phi\Delta}}{2}. \tag{26}$$

We have just proved the following Theorem.

Theorem 3.1
Let λ be an eigenvalue of the generalized eigenvalue problem

$$\begin{bmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}=\lambda\begin{bmatrix} A_0 & 0 \\ B & -S_0 \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix},$$

DOI: 10.1002/nla

where $A-A_0$ is positive definite. Then λ is real and positive, and moreover satisfies

$$\frac{(1+\phi)\Delta-\sqrt{(1+\phi)^2\Delta^2-4\phi\Delta}}{2}\leqslant\lambda\leqslant\frac{(1+\Phi)\delta-\sqrt{(1+\Phi)^2\delta^2-4\Phi\delta}}{2}$$

$$\delta\leqslant\lambda\leqslant\Delta \quad \text{or}$$

$$\frac{(1+\phi)\delta+\sqrt{(1+\phi)^2\delta^2-4\phi\delta}}{2}\leqslant\lambda\leqslant\frac{(1+\Phi)\Delta+\sqrt{(1+\Phi)^2\Delta^2-4\Phi\Delta}}{2},$$

where ϕ, Φ, δ and Δ are measures of the effectiveness of the approximation of A_0 to A and S_0 to $BA^{-1}B^{\mathrm{T}}$, as defined by (18) and (19).

Note that Theorem 3.1 can also be obtained as a direct consequence of Theorem 4.1 in [17]. The proof presented above is a more direct way to obtain the same bounds.

Now consider our application, where we have the system (9). Consider the preconditioner

$$\mathcal{P}_{BT}=\begin{bmatrix}\tilde{M} & 0 & 0\\ 0 & \beta\tilde{M} & 0\\ -K & M & -\tilde{K}M^{-1}\tilde{K}\end{bmatrix},$$

where we have taken $S_0=\tilde{K}M^{-1}\tilde{K}$ and $A_0=\mathtt{blkdiag}(\tilde{M},\beta\tilde{M})$, where $\tilde{\cdot}$ denotes an approximation to be defined below. We would like to find the bounds (18) and (19) for this particular case.

First, let us consider the idealized case where $\tilde{K}=K$. Then we need to know the eigenvalues of

$$(KM^{-1}K)^{-1}\left(KM^{-1}K+\frac{1}{\beta}M\right)v=\lambda v.$$

Corollary 3.3 in [10] gives us that the eigenvalue λ satisfies

$$\frac{1}{\beta}\alpha_1h^4+1\leqslant\lambda\leqslant\frac{1}{\beta}\alpha_2+1,$$

where α_1 and α_2 are constants independent of h and independent of β. These constants depend only on the discretization used, and are not large in general. For example, in 2D for **Q1** square elements the constants are $\alpha_1=1/1296$ and $\alpha_2=1/4\pi^2$. The lower bound can be made independent of h by simply noting that $(1/\beta)\alpha_1h^4>0$, hence we obtain the bound

$$1\leqslant\lambda\leqslant\frac{1}{\beta}\alpha_2+1. \tag{27}$$

In practice we will approximate the Schur complement as $\widetilde{K}M^{-1}\widetilde{K}$, where $\widetilde{K}$ is an approximation to the stiffness matrix. To see the effect of this, suppose that there exists a constant η such that $\|\widetilde{K}^{-1}K-I\|\leqslant\eta$. Then, using a duality argument as in Braess and Peisker [38, Section 4], where

DOI: 10.1002/nla

it was applied in the case of the biharmonic equation, we obtain that the generalized Rayleigh quotients satisfy

$$(1-\eta)^2 \leqslant \frac{w^{\mathrm{T}} K^{\mathrm{T}} M^{-1} K w}{w^{\mathrm{T}} \widetilde{K}^{\mathrm{T}} M^{-1} \widetilde{K} w} \leqslant (1+\eta)^2 \quad \forall w. \tag{28}$$

Thus, we can write

$$\frac{w^{\mathrm{T}} K M^{-1} K + \frac{1}{\beta} M w}{w^{\mathrm{T}} \widetilde{K} M^{-1} \widetilde{K} w} = \frac{w^{\mathrm{T}} K M^{-1} K + \frac{1}{\beta} M w}{w^{\mathrm{T}} K M^{-1} K w} \cdot \frac{w^{\mathrm{T}} K M^{-1} K w}{w^{\mathrm{T}} \widetilde{K} M^{-1} \widetilde{K} w}, \tag{29}$$

and so eigenvalues of

$$(\widetilde{K} M^{-1} \widetilde{K})^{-1} \left(K M^{-1} K + \frac{1}{\beta} M \right) \tag{30}$$

are bounded above and below by the product of the maximal and minimal values of the terms on the right-hand side of (29). Therefore, using (27) and (28), we have that these eigenvalues lie in the interval $[(1-\eta)^2, (1+\eta)^2((1/\beta\alpha_2+1)]$.

Therefore, if we take $\tilde{K}^{-1}$ to be a fixed number of steps of a multigrid routine, say, then this is a simple iteration, so certainly satisfies the required property $\|\widetilde{K}^{-1} K - I\| \leqslant \eta$ for some η. Therefore, we have a practical approximation of the Schur complement.

If we let $\tilde{M}$ denote a fixed number of steps of the Chebyshev semi-iteration (suitably scaled) then, as described in Section 3.1, we can ensure that we pick an approximation such that $A - A_0 > 0$ and can simply calculate the bounds δ and Δ from Table I.

Using Theorem 3.1 we can choose parameters in our Chebyshev semi-iteration and our multigrid to give us bounds in (18) and (19) that give us good clustering of the eigenvalues, and therefore good convergence of the Bramble–Pasciak CG. We discuss our choice of these parameters in Section 5.

4. BLOCK-DIAGONAL PRECONDITIONERS

The saddle point matrix (1) is typically symmetric and indefinite and one candidate to solve such problems is MINRES introduced by Paige and Saunders in [2]. In order to be able to apply MINRES to $\mathcal{A}$ we need a symmetric and positive-definite preconditioner such as the block-diagonal preconditioner

$$\mathcal{P}_{BD} = \begin{bmatrix} A_0 & 0 \\ 0 & S_0 \end{bmatrix} \tag{31}$$

where A_0 is a symmetric positive-definite preconditioner for the left upper block of $\mathcal{A}$ and S_0 is a preconditioner based on the Schur complement $BA^{-1}B^{\mathrm{T}}$ of $\mathcal{A}$. Preconditioners of such a form are well studied when the underlying matrix comes from the finite element treatment of a PDE (see [9, 12, 39] for further details). These preconditioners were also recently used by Rees, Dollar and Wathen to solve systems arising in the context of PDE-constrained optimization (see [10]).

DOI: 10.1002/nla

If we have a system of the form (9), then such a block-diagonal preconditioner is the one considered in [10]:

$$\mathcal{P}_{BD} = \begin{bmatrix} M & 0 & 0 \\ 0 & \beta M & 0 \\ 0 & 0 & KM^{-1}K^{\mathrm{T}} \end{bmatrix}. \tag{32}$$

Here $KM^{-1}K^{\mathrm{T}}$ is taken to be the approximation to the Schur complement $(1/\beta)M + KM^{-1}K^{\mathrm{T}}$. If the system is discretized using Q1 finite elements, as is the case here, then Rees, Dollar and Wathen show that if λ is an eigenvalue of the preconditioned system $\mathcal{P}_{BD}^{-1}\mathcal{A}$ then λ satisfies one of

$$\lambda = 1, \tag{33}$$

$$\frac{1}{2}\left(1+\sqrt{5+\frac{2\alpha_1 h^4}{\beta}}\right) \leqslant \lambda \leqslant \frac{1}{2}\left(1+\sqrt{5+\frac{2\alpha_2}{\beta}}\right) \tag{34}$$

$$\text{or} \quad \frac{1}{2}\left(1-\sqrt{5+\frac{2\alpha_2}{\beta}}\right) \leqslant \lambda \leqslant \frac{1}{2}\left(1-\sqrt{5+\frac{2\alpha_1 h^4}{\beta}}\right), \tag{35}$$

where α_1, α_2 are positive constants independent of the mesh size h and β, but which will change depending on the discretization used—as in Section 3, $\alpha_1 = \frac{1}{1296}$ and $\alpha_2 = 1/4\pi^2$ for the discretization considered here. More details can be found in [10].

To make this an effective preconditioner, further approximations to the blocks are made so that a solve with $\mathcal{P}_{BD}$ is cheaper. Therefore, the mass matrix is approximated by a certain number of iterations of the Chebyshev semi-iterative method (see Section 3.1), and the Schur complement by $\tilde{K}M^{-1}\tilde{K}^{\mathrm{T}}$, where $\tilde{K}$ represents a fixed number of algebraic multigrid V-cycles from the `HSL` package `HSL_MI20` [30]. This approximation will have a similar effect on the bounds (33)–(35) as the corresponding approximation had in Section 3.

5. NUMERICAL RESULTS

5.1. The problem

We illustrate our method using the following example, which is Example 5.1 in [10]. Let $\Omega = [0,1]^m$, where $m = 2, 3$, and consider the problem

$$\min_{y,u} \frac{1}{2}\|y - \bar{y}\|^2_{L_2(\Omega)} + \frac{\beta}{2}\|u\|^2_{L_2(\Omega)}$$

$$\text{s.t.} \quad -\nabla^2 y = u \quad \text{in } \Omega \tag{36}$$

$$y = \bar{y}|_{\partial\Omega} \quad \text{on } \partial\Omega \tag{37}$$

DOI: 10.1002/nla

where, in 2D,

$$\bar{y} = \begin{cases} (2x_1-1)^2(2x_2-1)^2 & \text{if } (x_1,x_2)\in[0,\tfrac{1}{2}]^2 \\ 0 & \text{otherwise} \end{cases}$$

and, in 3D,

$$\bar{y} = \begin{cases} (2x_1-1)^2(2x_2-1)^2(2x_3-1)^2 & \text{if } (x_1,x_2,x_3)\in[0,\tfrac{1}{2}]^3 \\ 0 & \text{otherwise} \end{cases}$$

i.e. $\bar{y}$ is bi- or tri-quadratic (depending on whether $m=2$ or 3) with a peak of unit height at the origin and is zero outside $[0,\frac{1}{2}]^m$. Note that we set $f\equiv 0$ here.

We discretize the problem using square Q1 finite elements, and in our preconditioner we approximate M by 10 steps of the Chebyshev semi-iteration and K by one or two V-cycles of the HSL package HSL_MI20 [30] (via a MATLAB interface). We choose $\gamma=0.9$ as the scaling parameter–this guarantees positive definiteness of $\mathcal{H}$ in (11).

One of the issues when comparing MINRES with block-diagonal preconditioning and the Bramble–Pasciak CG method is the right stopping criterion for both methods. It is well known that preconditioned MINRES minimizes the $\mathcal{P}^{-1}$-norm of the residual r_k and the Bramble–Pasciak CG minimizes the error in the $\mathcal{H}\widehat{\mathcal{A}}$-norm. In more detail, the Bramble–Pasciak CG minimizes

$$\|e_k\|^2_{\mathcal{H}\widehat{\mathcal{A}}} = (x-x_k)^{\mathrm{T}}\mathcal{H}\widehat{\mathcal{A}}(x-x_k)$$

which is equivalent to minimizing

$$(x-x_k)^{\mathrm{T}}\mathcal{A}\mathcal{A}^{-1}\mathcal{H}\widehat{\mathcal{A}}\mathcal{A}^{-1}\mathcal{A}(x-x_k) = r_k^{\mathrm{T}}\mathcal{A}^{-1}\mathcal{H}\widehat{\mathcal{A}}\mathcal{A}^{-1}r_k.$$

In the last equation the matrix $\mathcal{A}^{-1}\mathcal{H}\widehat{\mathcal{A}}\mathcal{A}^{-1}$ is symmetric and positive definite and hence its Cholesky factorization $\mathcal{A}^{-1}\mathcal{H}\widehat{\mathcal{A}}\mathcal{A}^{-1} = RR^{\mathrm{T}}$ exists. As a result, the following holds:

$$\|e_k\|^2_{\mathcal{H}\widehat{\mathcal{A}}} = r_k^{\mathrm{T}}\mathcal{A}^{-1}\mathcal{H}\widehat{\mathcal{A}}\mathcal{A}^{-1}r_k = \|Rr_k\|_2^2 \leqslant \|R\|_2^2\,\|r_k\|_2^2.$$

As the $\mathcal{H}\widehat{\mathcal{A}}$ is not a practical norm to work with we will use the 2-norm of the residual instead. We feel that this will make for a fair comparison of the two methods. In addition, we will show the comparison of both MINRES and Bramble–Pasciak CG on one example when both methods are terminated based on the 2-norm residual. Note that the meshsize h will be given by $\frac{1}{2}^N$.

5.2. Numerical experiments

The numerical results shown in Tables III(a) and IV(a) indicate that the Bramble–Pasciak CG shows mesh-independent convergence when only one V-cycle of the AMG preconditioner is applied. In these cases MINRES seems to be less independent of the number of V-cycles even when using the correct norm as a convergence test (see [10]). Both methods show mesh-independent convergence behaviour when two V-cycles are applied. We note that in every case shown the Bramble–Pasciak CG had fewer iterations than MINRES applied to the same problem. Recall that the cost of one step of the Bramble–Pasciak CG and one step of the block-diagonally preconditioned MINRES are essentially the same concerning the number of applications of the preconditioners A_0, A_1 and S_0. Additionally, the Bramble–Pasciak CG requires one more multiplication with the matrix B but no more additional matrix–vector products for the evaluation of the inner products. This

DOI: 10.1002/nla

Table III. 2D results for $\beta = 1e-4$.

	Bramble Pasciak		MINRES			Bramble Pasciak		MINRES	
N	its	Time	its	Time	N	its	Time	its	Time
	(a)					(b)			
2	14	0.0325600	16	0.0216490	2	14	0.0329460	16	0.0234080
3	15	0.0251500	24	0.0393010	3	15	0.0324720	24	0.0452570
4	15	0.0454840	26	0.0732190	4	15	0.0504730	26	0.0890670
5	15	0.1294110	26	0.2339460	5	15	0.1565560	26	0.2747820
6	15	0.6018610	26	0.9278930	6	14	0.6535900	26	1.0387500
7	15	2.4962350	26	3.9984930	7	14	3.4524540	26	6.9646050
8	16	12.2712530	28	18.8485520	8	14	13.3780360	26	27.9837960
9	18	56.6300730	30	88.5645490	9	15	52.8862420	24	78.4650840

2D results for $\beta = 1e-4$. (a) tol $= 1e-6$, $\beta = 1e-4$ and one V-cycle and (b) tol $= 1e-6$, $\beta = 1e-4$ and two V-cycles.

Table IV. 3D results for $\beta = 1e-2$.

	Bramble Pasciak		MINRES			Bramble Pasciak		MINRES	
N	its	Time	its	Time	N	its	Time	its	Time
	(a)					(b)			
2	8	0.0182060	11	0.0307280	2	8	0.0127420	11	0.0178690
3	9	0.0852650	12	0.1104170	3	8	0.0732610	12	0.1149870
4	9	1.1322740	13	1.5692170	4	8	1.2558520	12	1.7656210
5	8	10.4078570	13	16.9380940	5	8	14.7651730	12	22.6690480

3D results for $\beta = 1e-2$. (a) tol $= 1e-6$, $\beta = 1e-2$ and one V-cycle and (b) tol $= 1e-6$, $\beta = 1e-2$ and two V-cycles.

Table V. 3D results for $\beta = 1e-4$.

	Bramble Pasciak		MINRES			Bramble Pasciak		MINRES	
N	its	Time	its	Time	N	its	Time	its	Time
	(a)					(b)			
2	12	0.0172430	17	0.0242450	2	10	0.0158200	13	0.0262500
3	13	0.0920930	24	0.1653970	3	12	0.1300120	22	0.2266910
4	13	1.4422150	24	3.0399280	4	11	1.6798730	22	3.4427270
5	13	17.3131750	26	32.6883410	5	10	18.4158590	24	39.2577900

3D results for $\beta = 1e-4$. (a) tol $= 1e-6$, $\beta = 1e-4$ and one V-cycle and (b) tol $= 1e-6$, $\beta = 1e-4$ and two V-cycles.

makes the Bramble–Pasciak CG step only slightly more expensive than one step of MINRES. This means that for small systems the timings for MINRES and the Bramble–Pasciak CG do not differ as much as the iteration numbers indicate. On the other hand, we expect the timings to better reflect the difference between MINRES and the Bramble–Pasciak CG when both methods are implemented in a low-level language and are applied to much bigger systems. At the moment the AMG preconditioner relies on a fast FORTRAN implementation with the whole algorithm

DOI: 10.1002/nla

Table VI. 3D results for $\beta = 1e-5$.

	Bramble Pasciak		MINRES			Bramble Pasciak		MINRES	
N	its	Time	its	Time	N	its	Time	its	Time
	(a)					(b)			
2	17	0.0306910	27	0.0377740	2	14	0.0243660	16	0.0233870
3	26	0.2289160	50	0.3781130	3	21	0.1742860	43	0.3304610
4	25	2.8693070	53	5.8431290	4	23	2.8987280	49	5.7582700
5	24	33.4019030	54	61.3260090	5	21	30.7192610	48	66.9184560

3D results for $\beta = 1e-5$. (a) tol $= 1e-6$, $\beta = 1e-5$ and one V-cycle and (b) tol $= 1e-6$, $\beta = 1e-5$ and two V-cycles.

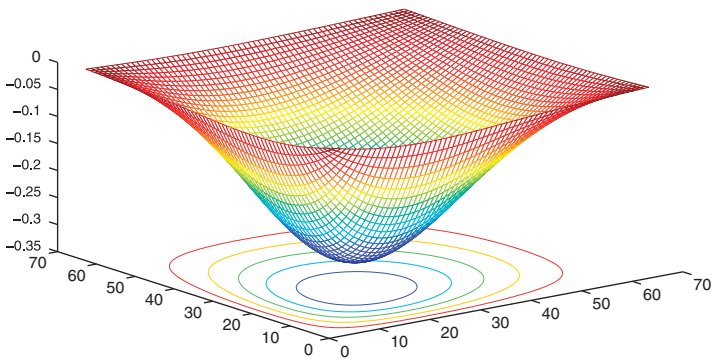

Figure 1. Control for $\beta = 1e-2$.

including the matrix multiplication being implemented in MATLAB. A low-level implementation of both methods allows for problems of much larger dimensions to be solved in which case the multiplication with the system matrix becomes increasingly expensive. This means that the extra iterations MINRES needs for convergence will effect the timings more significantly.

Tables II(a, b), IV(a, b) show the results for both methods when $\beta = 10^{-2}$ and Tables III(a, b), V(a, b) the results for both methods with $\beta = 10^{-4}$. In Tables VI(a, b) we show the results for $\beta = 10^{-5}$ for the 3D case. All results clearly indicate that the Bramble–Pasciak method outperforms MINRES with respect to the number of iterations and timings. For larger β, the difference between both methods is smaller compared with the difference for smaller β as shown in Tables IV(b), V(b) and VI(b). In addition, we show in (Figure 1 and 2) Figure 3 that the convergence of the Bramble–Pasciak method seems mesh-independent for a regularization parameter β down to 10^{-6}.

6. CONCLUSIONS

In this paper we have shown that the Bramble–Pasciak CG is a method well suited for problems from PDE-constrained optimization. The drawback of parameter estimation to ensure a positive-definite non-standard inner product, typically identified with this method, can be easily overcome when the Chebyshev semi-iteration is used as the preconditioners can be scaled very cheaply based

DOI: 10.1002/nla

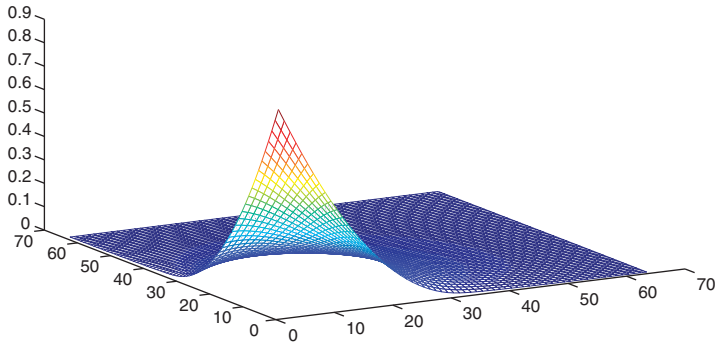

Figure 2. State for $\beta = 1e-2$.

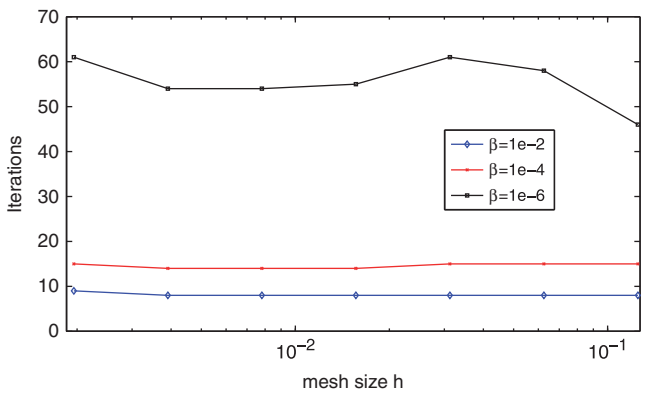

Figure 3. Iterations for convergence for different β.

on simple *a priori* calculations. We have shown an eigenvalue analysis for an idealized case as well as for a general setup. The competitiveness of the Bramble–Pasciak CG is illustrated by the numerical results where we show that this method also gives mesh independence for only one V-cycle of the AMG Schur-complement preconditioner.

ACKNOWLEDGEMENTS

The authors thank Andy Wathen for his support. They also thank Miro Rozložnik and an anonymous referee for their helpful comments and careful reading of the paper. This publication is partially based on work supported by Award No. KUK-C1-013-04, made by King Abdullah University of Science and Technology (KAUST); contract/grant number: KUK-C1-013-04.

REFERENCES

1. Benzi M, Golub GH, Liesen J. Numerical solution of saddle point problems. *Acta Numerica* 2005; **14**:1–137.
2. Paige CC, Saunders MA. Solutions of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis* 1975; **12**(4):617–629.

DOI: 10.1002/nla

3. Hestenes MR, Stiefel E. Methods of conjugate gradients for solving linear systems. *Journal of Research of the National Bureau of Standards* 1952; **49**:409–436 (1953).
4. Ashby S, Manteuffel T, Saylor P. A taxonomy for conjugate gradient methods. *SIAM Journal on Numerical Analysis* 1990; **27**(6):1542–1568.
5. Dollar S. Iterative linear algebra for constrained optimization. *Ph.D. Thesis*, University of Oxford, 2005. Available from: http://web.comlab.ox.ac.uk/oucl/research/na/thesis/thesisdollar.pdf.
6. Tröltzsch F. *Optimale Steuerung partieller Differentialgleichungen*: *Theorie*, *Verfahren und Anwendungen*. Vieweg Verlag: Wiesbaden, 2005.
7. Hinze M, Pinnau R, Ulbrich M, Ulbrich S. Mathematical modelling: theory and applications. *Optimization with PDE Constraints*. Springer: New York, 2009.
8. Stoll M, Wathen A. Preconditioning for active set and projected gradient methods as semi-smooth Newton methods for PDE-constrained optimization with control constraints. *SIAM Journal on Optimization*, 2009; submitted.
9. Elman HC, Silvester DJ, Wathen AJ. Numerical mathematics and scientific computation. *Finite Elements and Fast Iterative Solvers*: *with Applications in Incompressible Fluid Dynamics*. Oxford University Press: New York, 2005.
10. Rees T, Dollar HS, Wathen A. Optimal solvers for PDE-constrained optimization. *SIAM Journal on Scientific Computing* 2008; accepted for publication.
11. Bramble JH, Pasciak JE. A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems. *Mathematics of Computation* 1988; **50**(181):1–17.
12. Wathen A, Silvester D. Fast iterative solution of stabilised Stokes systems. I. Using simple diagonal preconditioners. *SIAM Journal on Numerical Analysis* 1993; **30**(3):630–649.
13. Elman HC. Multigrid and Krylov subspace methods for the discrete Stokes equations. In *Seventh Copper Mountain Conference on Multigrid Methods*, Melson ND, Manteuffel TA, McCormick SF, Douglas CC (eds), vol. CP 3339. NASA: Hampton, VA, 1996; 283–299. Available from: citeseer.ist.psu.edu/106673.html.
14. Klawonn A. Block-triangular preconditioners for saddle point problems with a penalty term. *SIAM Journal on Scientific Computing* 1998; **19**(1):172–184. Special issue on iterative methods (Copper Mountain, CO, 1996).
15. Carstensen C, Kuhn M, Langer U. Fast parallel solvers for symmetric boundary element domain decomposition equations. *Numerische Mathematik* 1998; **79**(3):321–347.
16. Ainsworth M, Sherwin S. Domain decomposition preconditioners for p and hp finite element approximation of Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 1999; **175**(3–4):243–266.
17. Zulehner W. Analysis of iterative methods for saddle point problems: a unified approach. *Mathematics of Computation* 2002; **71**(238):479–505.
18. Gatica GN, Heuer N. Conjugate gradient method for dual-dual mixed formulations. *Mathematics of Computation* 2002; **71**(240):1455–1472.
19. Axelsson O, Neytcheva M. Preconditioning methods for linear systems arising in constrained optimization problems. *Numerical Linear Algebra with Applications* 2003; **10**(1–2):3–31. Dedicated to the 60th birthday of Raytcho Lazarov.
20. Axelsson O, Neytcheva M. Robust preconditioners for saddle point problems. *Numerical Methods and Applications*. Lecture Notes in Computer Science, vol. 2542. Springer: Berlin, 2003, 158–166.
21. Peters J, Reichelt V, Reusken A. Fast iterative solvers for discrete Stokes equations. *SIAM Journal on Scientific Computing* 2005; **27**(2):646–666.
22. Murphy MF, Golub GH, Wathen AJ. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing* 2000; **21**(6):1969–1972.
23. Stoll M. Solving linear systems using the adjoint. *Ph.D. Thesis*, University of Oxford, 2009.
24. Elman HC. Preconditioners for saddle point problems arising in computational fluid dynamics. *Applied Numerical Mathematics* 2002; **43**(1–2):75–89. *19th Dundee Biennial Conference on Numerical Analysis*, 2001.
25. Stoll M, Wathen A. Combination preconditioning and the Bramble–Pasciak$^+$ preconditioner. *SIAM Journal on Matrix Analysis and Applications* 2008; **30**(2):582–608. DOI: 10.1137/070688961. Available from: http://link.aip.org/link/?SML/30/582/1.
26. Simoncini V. Block triangular preconditioners for symmetric saddle-point problems. *Applied Numerical Mathematics* 2004; **49**(1):63–80.
27. Rozložník M, Simoncini V. Krylov subspace methods for saddle point problems with indefinite preconditioning. *SIAM Journal on Matrix Analysis and Applications* 2002; **24**(2):368–391. DOI: 10.1137/S0895479800375540. URL: http://link.aip.org/link/?SML/24/368/1.
28. Saad Y. *Iterative Methods for Sparse Linear Systems*. SIAM: Philadelphia, PA, 2003.

DOI: 10.1002/nla

29. Rees T, Stoll M, Wathen A. All-at-once preconditioners for PDE-constrained optimization. *Kybernetica* 2009; submitted.
30. Boyle J, Mihajlovic MD, Scott JA. HSL_MI20: an efficient AMG preconditioner. *Technical Report RAL-TR-2007-021*, Rutherford Appleton Laboratory (CCLRC), 2007.
31. Wathen A, Rees T. Chebyshev semi-iteration in preconditioning. *Technical Report NA-08/14*, Oxford University Computing Laboratory, 2008.
32. Wathen AJ. Realistic eigenvalue bounds for the Galerkin mass matrix. *IMA Journal of Numerical Analysis* 1987; **7**(4):449–457.
33. Wathen AJ. On relaxation of Jacobi iteration for consistent and generalized mass matrices. *Communications in Applied Numerical Methods* 1991; **7**(2):93–102.
34. Varga RS. *Matrix Iterative Analysis*. Prentice-Hall Inc.: Englewood Cliffs, NJ, 1962.
35. Hageman LA, Young DM. *Applied Iterative Methods*. Academic Press Inc. [Harcourt Brace Jovanovich Publishers]: New York, 1981. Computer Science and Applied Mathematics.
36. Golub GH, Varga RS. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. I. *Numerical Mathematics* 1961; **3**:147–156.
37. Golub GH, Varga RS. Chebyshev semi-iterative methods, successive over-relaxation iterative methods, and second order Richardson iterative methods. II. *Numerical Mathematics* 1961; **3**:157–168.
38. Braess D, Peisker P. On the numerical solution of the biharmonic equation and the role of squaring matrices for preconditioning. *IMA Journal of Numerical Analysis* 1986; **6**(4):393–404.
39. Silvester D, Wathen A. Fast iterative solution of stabilised Stokes systems. II. Using general block preconditioners. *SIAM Journal on Numerical Analysis* 1994; **31**(5):1352–1367.

DOI: 10.1002/nla