A BLOCK DIAGONAL PRECONDITIONER FOR PDE CONSTRAINED OPTIMIZATION

TYRONE REES*

Abstract. Optimization problems where the constraint is a partial differential equation are important in many areas of the sciences and engineering. One such formulation is the Dirichlet distributed control problem, which can be written as a saddle point system. Here we present an 'all-at-once' method based on preconditioned MINRES, together with an efficient, optimal preconditioner which uses multigrid methods, to solve such problems.

AMS subject classifications. 49M25, 49K20, 65F10, 65N22, 65F50, 65N55.

Key words. Saddle point problems, Linear systems, Optimal Control, Preconditioning, PDE Constrained Optimization, All-at-once

1. Introduction. In this paper we consider the Dirichlet distributed control problem, which consists of a cost functional (1.1) to be minimized subject to a differential equation.

$$\min_{u,f} \frac{1}{2}||u - \hat{u}||_2^2 + \beta||f||_2^2 \tag{1.1}$$

$$\text{s.t.} \qquad -\nabla^2 u = f \text{ in } \Omega \tag{1.2}$$

$$u = g \text{ on } \partial\Omega. \tag{1.3}$$

Such problems were introduced by J.L. Lions in [9]. Here, the function $\hat{u}$, or the 'desired state' is known, and we want to get a solution u as close to it (in the sense of the L_2 norm) as possible, while still satisfying the PDE. In order to achieve this, the right hand side of the PDE, f, can be varied. The second term in the cost functional is added because, in general, the problem would be ill-posed, and so needs this Tikhonov regularization term. β is a parameter that needs to be determined, although it is usually selected a priori.

For simplicity we are considering the case where the boundary condition is Dirichlet, although Neumann and mixed boundary conditions are not a problem and can be treated in a similar way. We are looking at the simplest of PDEs, Poisson's equation, but again our method is not specific to this type of equation: all that is required is a preconditioner for the PDE.

There is a choice now as to whether to discretize-then-optimize or optimize-then-discretize, and there are differing opinions regarding which route to take. We have chosen to discretize-then-optimize, as then we guarantee symmetry in the resulting linear system, which is not the case for all PDEs if we optimize-then-discretize (see, e.g., [3]).

We discretize our problem using finite elements. The weak formulation of (1.2) and

*Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford, OX1 3QD, United Kingdom, (tyrone.rees@comlab.ox.ac.uk)

(1.3) is to find $u \in H^1_g(\Omega) = \{u \ : \ u \in H^1(\Omega), \ u = g \text{ on } \partial\Omega\}$ such that

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega vf \qquad \forall v \in H^1_0(\Omega) \tag{1.4}$$

We assume that $V^h_0 \subset H^1_0$ is an n-dimensional vector space of test functions with $\{\phi_1, ..., \phi_n\}$ as a basis. Then, for the boundary condition to be satisfied, we extend the basis by defining functions $\phi_{n+1}, ..., \phi_{n+\partial n}$ and coefficients U_j so that $\sum_{j=n+1}^{n+\partial n} U_j\phi_j$ interpolates the boundary data. Then, if $u_h \in V^h_g \subset H^1_g(\Omega)$, it is uniquely determined by $\mathbf{u} = (U_1 \dots U_n)^T$ in

$$u_h = \sum_{j=1}^{n} U_j\phi_j + \sum_{j=n+1}^{n+\partial n} U_j\phi_j.$$

Here the ϕ_i, $i = 1, \dots, n$, define a set of shape functions. We also assume that this approximation is conforming, i.e. $V^h_g = \text{span}\{\phi_1, \dots, \phi_{n+\partial n}\} \subset H^1_g(\Omega)$. Then we get the finite-dimensional analogue of (1.4): find $u_h \in V^h_g$ such that

$$\int_\Omega \nabla u_h \cdot \nabla v_h = \int_\Omega v_h f \qquad \forall v_h \in V^h_0.$$

We also need a discretization of f, as this appears in (1.1). We discretize this using the same basis used for u, so

$$f_h = \sum_{j=1}^{n+\partial n} F_j\phi_j.$$

Thus we can write the discrete analogue of the minimization problem as

$$\min_{u_h, f_h} \frac{1}{2}||u_h - \hat{u}||^2_2 + \beta||f_h||^2_2 \tag{1.5}$$

$$\text{such that} \qquad \int_\Omega \nabla u_h \cdot \nabla v_h = \int_\Omega v_h f \qquad \forall v_h \in V^h_0. \tag{1.6}$$

We can write the discrete cost functional as

$$\min_{u_h, f_h} \frac{1}{2}||u_h - \hat{u}||^2_2 + \beta||f_h||^2_2 = \min_{\mathbf{u},\mathbf{f}} \frac{1}{2}\mathbf{u}^T M\mathbf{u} - \mathbf{u}^T\mathbf{b} + \beta\mathbf{f}^T M_f\mathbf{f} \tag{1.7}$$

where $\mathbf{u} = (U_1, \dots, U_n)^T$, $\mathbf{f} = (F_1, \dots, F_{n+\partial n})^T$, $\mathbf{b} = \{\int \hat{u}\phi_i\}_{i=1\dots n}$ and $M = \{\int \phi_i\phi_j\}_{i,j=1\dots n}$ and $M_f = \{\int \phi_i\phi_j\}_{i,j=1\dots n+\partial n}$ are mass matrices.

We now turn our attention to the constraint. (1.6) is equivalent to finding $\mathbf{u}$ such that

$$\int_\Omega \nabla\left(\sum_{i=1}^{n} U_i\phi_i\right)\cdot\nabla\phi_j + \int_\Omega \nabla\left(\sum_{i=n+1}^{n+\partial n} U_i\phi_i\right)\cdot\nabla\phi_j = \int_\Omega \left(\sum_{i=1}^{n+\partial n} F_i\phi_i\right)\phi_j \quad \forall j = 1, \dots, n$$

$$\sum_{i=1}^{n} U_i \int_\Omega \nabla\phi_i \cdot \nabla\phi_j = \sum_{i=1}^{n+\partial n} F_i \int_\Omega \phi_i\phi_j - \sum_{i=n+1}^{n+\partial n} U_i \int_\Omega \nabla\phi_i \cdot \nabla\phi_j \quad \forall j = 1, \dots, n$$

$$K\mathbf{u} = N\mathbf{f} + \mathbf{d}. \tag{1.8}$$

So (1.7) and (1.8) together are equivalent to (1.5) and (1.6). One way to solve this minimization problem is by considering the Lagrangian

$$\mathscr{L} := \frac{1}{2}\mathbf{u}^T M\mathbf{u} - \mathbf{u}^T\mathbf{b} + \beta\mathbf{f}^T M_f\mathbf{f} + \lambda^T(K\mathbf{u} - N\mathbf{f} - \mathbf{d})$$

where λ is a vector of Lagrange multipliers. Then as we know that $\mathbf{f}$, $\mathbf{u}$ and λ make $\mathscr{L}$ stationary we get the following linear system

$$\begin{bmatrix} 2\beta M_f & 0 & -N \\ 0 & M & K^T \\ -N & K & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f} \\ \mathbf{u} \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \\ \mathbf{d} \end{bmatrix}. \tag{1.9}$$

Note that system of equations has the saddle point system structure, i.e. it is of the form

$$\begin{bmatrix} A & B^T \\ B & C \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}$$

where $A = \begin{bmatrix} 2\beta M_f & 0 \\ 0 & M \end{bmatrix}$, $B = [-N \quad K]$, $C = 0$, etc.

This system is usually very large – each of the blocks K is itself a discretization of the PDE – and sparse, since as well as the 0 blocks, K, M, M_f and N are themselves sparse. Therefore we are going to solve the system iteratively. In general the system will be symmetric but indefinite, so we will use the Minimal Residual (MINRES) method of Paige and Saunders [14], which is a Krylov subspace method for symmetric linear systems.

2. Preconditioning. Suppose we want to solve the linear system

$$\mathcal{A}\mathbf{x} = \mathbf{b}$$

where $\mathcal{A}$ is an $n \times n$ matrix. MINRES works by, at each step, selecting the next iterate by choosing $\mathbf{x}_k \in \mathcal{K}_k(\mathcal{A}; \mathbf{r}_0) := \text{span}(\mathbf{r}_0, \mathcal{A}\mathbf{r}_0, \ldots, \mathcal{A}^{k-1}\mathbf{r}_0)$ such that

$$||\mathbf{r}_k|| \leq ||\mathbf{s}|| \ \forall \ \mathbf{s} \in \mathbf{r}_0 + \text{span}\{\mathcal{A}\mathbf{r}_0, \ldots, \mathcal{A}^k\mathbf{r}_0\}$$

where $\mathbf{r}_m := \mathbf{b} - \mathcal{A}\mathbf{x}_m$. The following convergence result is well known - see [4], for example. If the eigenvalues of $\mathcal{A}$ are denoted by μ_j, then

$$||\mathbf{r}_k|| \leq \min_{p_k \in \Pi_k, p_k(0)=1} \max_j |p_k(\mu_j)| \ ||\mathbf{r}_0||.$$

Thus if the eigenvalues of $\mathcal{A}$ lie in some inclusion interval $[-\alpha, -\beta] \cup [\gamma, \delta]$ for some $\alpha, \beta, \gamma, \delta > 0$ where the intervals are small and well seperated from the origin, then we will get convergence of MINRES in a small number of iterations.

A given matrix will not, in general, have such nice properties, so we use preconditioning. The idea is to find a matrix (or a linear process) $\mathcal{P}$ for which $\mathcal{P}^{-1}\mathcal{A}$ has

better spectral properties (and such that $\mathcal{P}^{-1}\mathbf{v}$ is cheap to evaluate for any given vector $\mathbf{v}$). We then solve a symmetric preconditioned system equivalent to

$$\mathcal{P}^{-1}\mathcal{A}\mathbf{x} = \mathcal{P}^{-1}\mathbf{b}.$$

With preconditioning it can be shown that it is $||\mathbf{r}_k||_{\mathcal{P}^{-1}}$ that is minimized, and the corresponding preconditioned MINRES convergence estimate becomes

$$||\mathbf{r}_k||_{\mathcal{P}^{-1}} \leq \min_{p_k \in \Pi_k, p_k(0)=1} \max_j |p_k(\mu_j)| \; ||\mathbf{r}_0||_{\mathcal{P}^{-1}}.$$

Note that the preconditioner $\mathcal{P}$ must be positive definite in order for this to define a norm. The aim of preconditioning is to choose a matrix $\mathcal{P}$ such that the eigenvalues of $\mathcal{P}^{-1}\mathcal{A}$ are clustered. The following result by Murphy, Golub and Wathen [12] illustrates this idea:

THEOREM 2.1. *If*

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

is preconditioned by

$$\mathcal{P} = \begin{bmatrix} A & 0 \\ 0 & BA^{-1}B^T \end{bmatrix}$$

Then the preconditioned matrix $\mathcal{T} = \mathcal{P}^{-1}\mathcal{A}$ *satisfies*

$$\mathcal{T}(\mathcal{T} - I)(\mathcal{T}^2 - \mathcal{T} - I) = 0$$

This shows us that $\mathcal{T}$ is diagonalizable and has at most four distinct eigenvalues $(0, 1, \frac{1\pm\sqrt{5}}{2})$, or only the three non-zero eigenvalues if $\mathcal{T}$ is nonsingular. This means that the Krylov subspace $\mathcal{K}(\mathcal{T}; \mathbf{r}) = \text{span}(\mathbf{r}, \mathcal{T}\mathbf{r}, \mathcal{T}^2\mathbf{r}, \dots)$ will be of dimension at most three if $\mathcal{T}$ is nonsingular or four if $\mathcal{T}$ is singular. Therefore, any Krylov subspace method with an optimality property (such as MINRES) will terminate in at most three iterations.

If we apply this approach to the matrix in our saddle point system (1.9) then we get the preconditioner

$$\mathcal{P}_{\text{MGW}} = \begin{bmatrix} 2\beta M_f & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & \frac{1}{2\beta}NM_f^{-1}N + KM^{-1}K^T \end{bmatrix}.$$

While MINRES with this preconditioner will always terminate in at most three steps, and so satisfies one requirement of a preconditioner, it fails on another count as it is, in general, not cheap to solve the corresponding system with $\mathcal{P}_{\text{MGW}}$. However, we could still make use of the properties of this preconditioner by approximating it in such a way that the eigenvalues remain clustered. Looking at the structure of $\mathcal{P}_{\text{MGW}}$, the mass matrices in the (1,1) and the (2,2) blocks do not pose too much of a problem: they can be cheaply solved by, for example, using PCG with the diagonal as the preconditioner, as shown in [18]. Thus the difficulty comes from the (3,3) block,

which is the only part that contains the PDE.

One way to approximate this is to consider only the dominant term in the (3,3) block which is, for all but the very smallest values of β, the $KM^{-1}K^T$ term, thus making the preconditioner

$$\mathcal{P}_{\mathrm{D1}} = \begin{bmatrix} 2\beta M_f & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & KM^{-1}K^T \end{bmatrix}.$$

One can show the following result that tells us about the clustering of the eigenvalues using this preconditioner, which is an application and extension of a result in [1]:

PROPOSITION 2.2. *Let λ be an eigenvalue of $\mathcal{P}_{\mathrm{D1}}^{-1}\mathcal{A}$. Then either $\lambda = 1$, $\frac{1}{2}(1+\sqrt{1+4\sigma_1}) \leq \lambda \leq \frac{1}{2}(1+\sqrt{1+4\sigma_m})$ or $\frac{1}{2}(1-\sqrt{1+4\sigma_m}) \leq \lambda \leq \frac{1}{2}(1-\sqrt{1+4\sigma_1})$, where $0 \leq \sigma_1 \leq \cdots \leq \sigma_m$ are the eigenvalues of $\frac{1}{2\beta}(KM^{-1}K^T)^{-1}M + I$.*

We can use this general result to get more concrete bounds that are dependent both on the PDE in the problem being considered and on what finite element discretization is used. In our tests we have discretized the problem (1.1) using bilinear quadrilateral $\mathbf{Q_1}$ finite elements, and for this choice one can prove the following.

COROLLARY 2.3. *Let λ be an eigenvalue of $\mathcal{P}_{\mathrm{D1}}^{-1}\mathcal{A}$. Then λ satisfies one of*

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

where α_1, α_2 are positive constants.

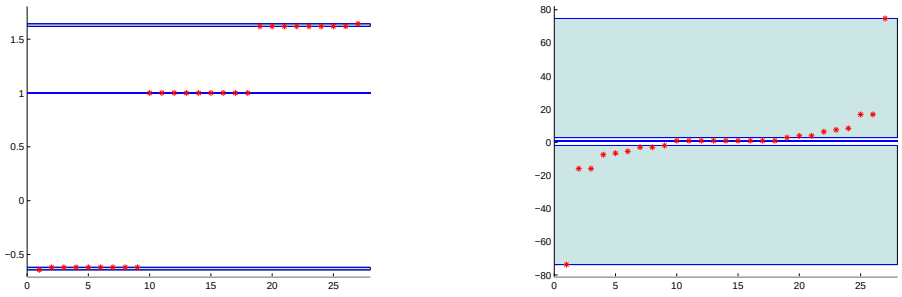
Note that the bounds in Corollary 2.3 do not get worse as we refine the mesh – they depend on h in a multiplicative sense only – which suggests that this is an optimal preconditioner in the sense that it is independent of the mesh size. Figure 2.1 shows a plot of the actual eigenvalues and the bounds of Corollary 2.3 for two choices of β. As seen in the figure, the eigenvalues are much more clustered for the larger value of β, and we see that for values around this our method is most successful. Taking β around this value is quite common in the literature – see Collis and Heinkenschloss [3], Haber and Ascher [7], Maurer and Mittelmann [10],[11] or Ito and Kunisch [8], for example.

We illustrate our method with the following example:

EXAMPLE 2.4. *Let $\Omega = [0,1]^m$, where $m = 2, 3$, and consider the problem*

$$\min_{u,f} \frac{1}{2}||u-\hat{u}||_2^2 + \beta||f||_2^2 \tag{2.1}$$

$$\text{s.t.} \qquad -\nabla^2 u = f \text{ in } \Omega \tag{2.2}$$
$$u = \hat{u}|_{\partial\Omega} \text{ on } \partial\Omega \tag{2.3}$$



(a) $\beta = 10^{-2}$ (b) $\beta = 10^{-7}$

FIG. 2.1. *Eigenvalues of $\mathcal{P}_{\mathrm{D1}}^{-1}\mathcal{A}$ and eigenvalue bounds predicted by Proposition 2.2 (dark lines are the extrema, light shading is where eigenvalues lie)*

where

$$\hat{u} = \begin{cases} \prod_{i=1}^{m}(2x_i - 1)^2 & \text{if } (x,y) \in [0,\frac{1}{2}]^m \\ 0 & \text{otherwise} \end{cases} .$$

i.e. $\hat{u}$ 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 $[0,\frac{1}{2}]^m$. Consider the value of $\beta = 10^{-2}$.

Applying preconditioned MINRES with preconditioner $\mathcal{P}_{\mathrm{D1}}$ to Example 2.4 we get the following results. Note that, to solve a system with $\mathcal{P}_{\mathrm{D1}}$, we have to solve a system with $KM^{-1}K^T$, i.e. we have to solve for K and K^T once each. Here we do these solves using a sparse direct method. All tests were done using MATLAB version 7.0.4 on a machine with a dual processor AMD Opteron 244 (1.8GHz).

TABLE 2.1
2D with $\mathcal{P}_{\mathrm{D1}}$

		Tolerance 1e-12		Tolerance 1e-6	
N	**# of unknowns**	**# of iterations**	**time (s)**	**# of iterations**	**time (s)**
4	43	11	0.002	7	0.001
8	179	13	0.018	9	0.018
16	739	13	0.106	9	0.076
32	3011	13	0.707	9	0.516
64	12163	13	4.932	9	3.594
128	48899	13	39.219	9	28.168
256	196099	13	418.478	9	298.052
512	785411	-	-	-	-

As we see from the data in Tables 2.1 and 2.2, for both tolerances we get mesh size independent convergence and, just as the theory predicts, we get convergence in a relatively small number of iterations. The preconditioner works just as effectively in 3D as it does in 2D. The only problem is, since $\mathcal{P}_{\mathrm{D1}}$ depends on a direct solver to

TABLE 2.2
3D with $\mathcal{P}_{\mathrm{D1}}$

		Tolerance 1e-12		Tolerance 1e-6	
N	**# of unknowns**	**# of iterations**	**time (s)**	**# of iterations**	**time (s)**
4	179	9	0.094	7	0.005
8	1415	11	0.567	7	0.386
16	11663	13	102.845	7	58.844
32	95519	-	-	-	-
64	774719	-	-	-	-

solve for K, as h decreases the computer will run out of memory and be unable to perform the calculation (as is the case for $N = 512$ in Table 2.1 and $N = 32$ in Table 2.2). Thus this preconditioner fails to be easily invertible for fine discretizations.

3. Approximating the solves. While using $\mathcal{P}_{\mathrm{D1}}$ as a preconditioner, the main drawback is that at each iteration we have to solve for K and K^T once each, which is equivalent to solving the forward problem twice per iteration, and is costly, especially for more complicated PDEs. Because these solves are only needed for the preconditioner, which is itself just an approximation, then all we need is to solve these approximately. Thus we want to consider

$$\mathcal{P}_{\mathrm{D2}} = \begin{bmatrix} 2\beta\tilde{M}_f & 0 & 0 \\ 0 & \tilde{M} & 0 \\ 0 & 0 & \tilde{K}M^{-1}\tilde{K}^T \end{bmatrix}$$

where $\tilde{K}$ is some approximation to K and $\tilde{M}$ is an approximation to M. If these are good enough approximations then the spectral bounds should be close to those shown above, and so only taking slightly more Krylov subspace iterations, but with a solve for $\tilde{K}$ being much faster than the original sparse direct solve for K, hence giving us a much more effective preconditioner.

For any choice of the PDE in a problem of the type in (1.1) it is likely that there has been work done on solving the forward problem (i.e. solving just for a solution to the PDE), and we propose to draw from ideas here to help us develop effective preconditioners. If we have an effective preconditioner for the forward problem then we can incorporate it into our methods to give us an effective preconditioner for the PDE constrained optimization problem.

In the case of our PDE, Poisson's equation, a fixed number of multigrid iterations is a good preconditioner [4]. As a smoother we use relaxed Jacobi, i.e. if we have to solve $B\mathbf{u} = \mathbf{f}$, take $D = \mathrm{diag}(B)$ and iterate

$$\mathbf{u}^{(m+1)} = (I - \omega D^{-1}B)\mathbf{u}^{(m)} + \omega D^{-1}\mathbf{f} \tag{3.1}$$

where $\mathbf{u}^{(0)} = \mathbf{0}$ and for some relaxation parameter ω. Thus, here we have let $\tilde{K}$ denote two multigrid V-cycles with 2 pre- and 2 post-smoothing steps of relaxed Jacobi with the optimal (see [4]) relaxation parameter of $\omega = \frac{8}{9}$ in the 2D case. In 3D we use 3 pre- and 3 post-smoothing steps of unrelaxed Jacobi, i.e. take $\omega = 1$ in the above, which is optimal.

For the mass matrices, M and M_f, what we would like to use is a few steps for the preconditioned conjugate gradient method with, say, the diagonal as a preconditioner applied to the matrix, as this will give us a good approximation. However, PCG is not linear in the right hand side, so we cannot use it straight forwardly as a preconditioner. The Chebyshev semi-iteration [6] is a method of accelerating convergence of a simple iterative method which is linear, so we can employ it here. In 2D we use relaxed Jacobi with a relaxation parameter of $\frac{4}{5}$, which, when applied to a $\mathbf{Q_1}$ mass matrix, gives an iteration matrix with eigenvalues satisfying $|\lambda| \leq \frac{4}{5} =: \rho$. In 3D the optimal relaxation parameter is $\frac{4}{7}$, which gives eigenvalues such that $|\lambda| \leq \frac{13}{14} =: \rho$. In both cases, if we want to solve $M\mathbf{u} = \mathbf{f}$, say, then the k^{th} iterate of the Chebyshev semi-iteration is given by

$$\mathbf{y}^{(k)} = \sum_{i=0}^{k} \nu_i \mathbf{u}^{(i)}$$

where $\mathbf{u}^{(i)}$ are the iterates of the underlying iterative method (so $\mathbf{u}^{(i)} = S\mathbf{u}^{(i-1)} + g$ where S is some iteration matrix, defined here by relaxed Jacobi) and ν_i are the coefficients of the scaled Chebyshev polynomial $\hat{T}_k(z) = \frac{T_k(z/\rho)}{T_k(1/\rho)}$. This can be implemented more efficiently by performing the iteration

$$\mathbf{y}^{(k+1)} = w_{k+1}(S\mathbf{y}^{(k)} + g - \mathbf{y}^{(k-1)}) + \mathbf{y}^{(k-1)} \tag{3.2}$$

where $w_{k+1} = \frac{T_k(1/\rho)}{\rho T_{k+1}(1/\rho)}$ (see Varga [17]). It is very cheap to carry out an iteration using this scheme. Moreover, we get the following convergence result, which shows this method has the same convergence as classical conjugate gradients,

$$||\mathbf{u} - \mathbf{y}^{(k)}||_2 \leq \max_{r \in [-\rho, \rho]} |\hat{T}_k(r)| \; ||\mathbf{u} - \mathbf{u}^{(0)}||_2. \tag{3.3}$$

This suggests that a fixed number of these iterations will give us a good approximation to M. This is a linear operation which is cheap to implement, so is valid to use as a preconditioner. We therefore let $\tilde{M}$ and $\tilde{M}_f$ in $\mathcal{P}_{\text{D2}}$ denote 20 iterations of the Chebyshev semi-iteration, as defined above.

Applying preconditioned MINRES with our approximated preconditioner $\mathcal{P}_{\text{D2}}$ to Example 2.4 we get the following results.

Here we see that, in both 2D and 3D, the number of iterations taken to converge to the required tolerance has grown by just few when doing the multigrid approximation and the Chebyshev semi-iteration as opposed to solving for K, M and M_f exactly, suggesting that the approximations we make here are a reasonable approximation to the original preconditioner. Note that P_{D2} is still optimal here, in the sense that the number of iterations is independent of the mesh size.

The main thing that we see from the data is that, although preconditioning with $\mathcal{P}_{\text{D2}}$ takes a slightly larger number of iterations to converge, as the mesh size h decreases each iteration is cheaper than an iteration with $\mathcal{P}_{\text{D1}}$. As the method with $\mathcal{P}_{\text{D2}}$ is entirely factorization free we have no memory problems such as we had with $\mathcal{P}_{\text{D1}}$,

TABLE 3.1
2D with $\mathcal{P}_{D2}$

		Tolerance 1e-12		Tolerance 1e-6	
N	**# of unknowns**	**# of iterations**	**time (s)**	**# of iterations**	**time (s)**
4	43	12	0.015	7	0.010
8	179	14	0.055	7	0.026
16	739	16	0.149	9	0.087
32	3011	16	0.455	9	0.265
64	12163	16	1.879	9	1.105
128	48899	16	8.195	9	4.691
256	196099	16	37.961	9	22.268
512	785411	16	161.670	9	97.152

TABLE 3.2
3D with $\mathcal{P}_{D2}$

		Tolerance 1e-12		Tolerance 1e-6	
N	**# of unknowns**	**# of iterations**	**time (s)**	**# of iterations**	**time (s)**
4	179	9	0.023	7	0.021
8	1415	13	0.204	7	0.115
16	11663	14	2.702	7	1.476
32	95519	15	35.063	7	17.882
64	774719	15	397.167	9	298.521

and so this method is suitable for large scale problems. Indeed, looking at the results in Table 3.1 we see the timings are very close to scaling linearly with the problem size, so we have nearly optimal complexity.

4. Conclusion. We have presented an optimal preconditioner for a Dirichlet distributed control problem. We have demonstrated that our preconditioner works very effectively with regularization parameter $\beta = 10^{-2}$, but the approximations are less valid as β approaches zero. Only a handful of papers in the literature consider the saddle point structure of the matrices when solving problems of this type, and we believe using this to help us is a vital part of creating an efficient algorithm, given the large amount of work that has been done on solving these systems (e.g. see the survey paper by Benzi, Golub and Liesen [2]).

There are some β-independent methods in the literature, in particular there is a recent paper by Schöberl and Zulehner [15], which also uses block preconditioners, that gives a preconditioner that is both h and β inedependent. However, their method is specific to the problem they consider, and so is not as general as the method we present here. Also, for the values of β where our method is effective it outperforms the Schöberl and Zulehner method.

We have presented a preconditioner for the simplest case of PDE, Poisson's equation, but our method is more general than that. For any PDE, if there is a preconditioner available for the forward problem one could use that as $\tilde{K}$ and should still get good convergence.

Also, although we have presented the Dirichlet distributed control problem, changing the problem to, for example, a Neumann distributed control problem or a boundary control problem will give a matrix with the same block structure as (1.9). We can therefore use the same ideas to develop optimal preconditioners for these classes of problems too.

Acknowledgments. Thank you to Andy Wathen of the University of Oxford and Sue Dollar of the Rutherford Appleton Laboratory for their help and advice with this work.

REFERENCES

[1] Owe Axelsson and Maya Neytcheva. Eigenvalue estimates for preconditioned saddle point matrices. *Numer. Linear Algebra Appl.*, 13:339–360, 2006.

[2] Michele Benzi, Gene H. Golub, and Jörg Liesen. Numerical solution of saddle point problems. *Acta Numer.*, 14:1–137, 2005.

[3] S. S. Collis and M. Heinkenschloss. Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems. Technical Report TR02–01, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77005–1892, 2002.

[4] Howard Elman, David Silvester, and Andy Wathen. *Finite Elements and Fast Iterative Solvers with Applications in incompressible fluid dynamics*. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2005.

[5] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[6] G.H. Golub and R.S. Varga. Chebyshev semi iterative methods, successive overrelaxation iterative methods and second order Richardson iterative methods. *Numer. Math.*, 3(1):147–156, 1961.

[7] E. Haber and U. Ascher. Preconditioned all–at–once methods for large sparse parameter estimation problems, 2000.

[8] Kazufumi Ito and Karl Kunisch. Augmented Lagrangian–SQP methods for nonlinear optimal control problems of tracking type. *SIAM J. Control Optim.*, 34(3):874–891, 1996.

[9] J. L. Lions. *Optimal Control of Systems*. Springer, 1968.

[10] Helmut Maurer and Hans D. Mittelmann. Optimization techniques for solving elliptic control problems with control and state constraints. Part 1: Boundary control. *Comput. Optim. Appl.*, 16(2):29–55, 2000.

[11] Helmut Maurer and Hans D. Mittelmann. Optimization techniques for solving elliptic control problems with control and state constraints. Part 2: Distributed control. *Comput. Optim. Appl.*, 18(2):141–160, 2001.

[12] Malcolm F. Murphy, Gene H. Golub, and Andrew J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM Journal on Scientific Computing*, 21(6):1969–1972, 2000.

[13] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization*. Springer, 1999.

[14] C. C. Paige and M. A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, 12(4):617–629, 1975.

[15] Joachim Schöberl and Walter Zulehner. Symmetric indefinite preconditioners for saddle point problems with applications to PDE-constrained optimization problems. *SIAM Journal on Matrix Analysis and Applications*, 29(3):752–773, 2007.

[16] Lloyd N. Trefethen and David Bau. *Numerical Linear Algebra*. SIAM, 1997.

[17] R.S. Varga. *Matrix Iterative Analysis*. Prentice Hall, 1962.

[18] A. J. Wathen. Realistic eigenvalue bounds for the Galerkin mass matrix. *IMA J Numer Anal*, 7(4):449–457, 1987.