



# Distributed control and constraint preconditioners

H S Thorne

April 29, 2010

RAL-TR-2010-016

© Science and Technology Facilities Council

Enquires about copyright, reproduction and requests for additional copies of this report should be addressed to:

Library and Information Services  
SFTC Rutherford Appleton Laboratory  
Harwell Science and Innovation Campus  
Didcot  
OX11 0QX  
UK  
Tel: +44 (0)1235 445384  
Fax: +44(0)1235 446403  
Email: library@rl.ac.uk

The STFC ePublication archive (epubs), recording the scientific output of the Chilbolton, Daresbury, and Rutherford Appleton Laboratories is available online at: <http://epubs.cclrc.ac.uk/>

**ISSN 1358-6254**

Neither the Council nor the Laboratory accept any responsibility for loss or damage arising from the use of information contained in any of their reports or in any communication about their tests or investigation

# Distributed control and constraint preconditioners<sup>1</sup>

H. Sue Thorne<sup>2</sup>

## ABSTRACT

Optimization problems with constraints that involve a partial differential equation arise widely in many areas of the sciences and engineering, in particular in problems of design. The solution of such PDE-constrained optimization problems is usually a major computational task. Here we consider simple problems of this type: distributed control problems in which the 2- and 3-dimensional Poisson problem is the PDE. Large dimensional linear systems result from the discretization and need to be solved: these systems are of saddle-point type. We introduce an optimal preconditioner for these systems that leads to convergence of symmetric Krylov subspace iterative methods in a number of iterations which does not increase with the dimension of the discrete problem. These preconditioners are block structured and involve standard multigrid cycles. The optimality of the preconditioned iterative solver is proved theoretically and verified computationally in several test cases. The theoretical proof indicates that these approaches may have much broader applicability for other partial differential equations.

---

<sup>1</sup> This work was supported by the EPSRC grant EP/E053351/1.

<sup>2</sup> Computational Science and Engineering Department, Rutherford Appleton Laboratory,  
Chilton, Oxfordshire, OX11 0QX, England, EU.  
Email: sue.thorne@stfc.ac.uk  
Current reports available from “<http://www.numerical.rl.ac.uk/reports/>”.

Computational Science and Engineering Department  
Atlas Centre  
Rutherford Appleton Laboratory  
Oxfordshire OX11 0QX  
April 29, 2010

## 1 Introduction

In this paper, we consider the iterative solution of distributed control problems. The problems considered consist of a cost functional to be minimized subject to a partial differential equation (PDE) posed on a domain in  $\Omega \subset \mathbb{R}^2$  or  $\mathbb{R}^3$  (in this case, the Poisson equation with Dirichlet boundary conditions):

$$\min_{u,f} \frac{1}{2} \|u - \hat{u}\|_{L^2(\hat{\Omega})}^2 + \beta \|f\|_{L^2(\Omega)}^2 \quad (1.1)$$

$$\text{subject to } -\nabla^2 u = f \text{ in } \Omega \quad (1.2)$$

$$\text{with } u = g \text{ on } \delta\Omega. \quad (1.3)$$

Here, the function  $\hat{u}$  (the ‘desired state’) is known over a domain  $\hat{\Omega} \subseteq \Omega$  and we want to find  $u$  that satisfies the PDE over  $\Omega$  and is as close to  $\hat{u}$  as possible in the  $L_2$ -norm sense over  $\hat{\Omega}$ . In order to do this, the right-hand side of the PDE,  $f$ , (also known as the ‘control’) can be varied. The second term in the cost functional (1.1), a Tikhonov regularization term, is added because the problem may be either ill-posed or the right-hand side of the PDE,  $f$ , rapidly varies across the domain  $\Omega$ . In general, the Tikhonov parameter  $\beta$  needs to be determined, although it is often selected a priori – a value around  $\beta = 10^{-2}$  is commonly used (see [3, 7, 8] ) but, as shown in [12], more realistic values for these problems are  $\beta = \mathcal{O}(10^{-5})$ .

In PDE-constrained optimization we can either discretize-then-optimize or optimize-then-discretize, and there are differing opinions regarding which route to take (see Collis and Heinkenschloss [3] for a discussion). We have chosen to discretize-then-optimize, as we are then guaranteed symmetry in the resulting linear system. The underlying optimization problems are naturally self-adjoint and by this choice we avoid non-symmetry due to discretization that can arise with the optimize-then-discretize approach (as shown in, for example, Collis and Heinkenschloss [3]). We discuss the formulation and general structure of our discretized problem in Section 2. We employ the Galerkin finite element method for discretization here, but see no reason why other approximation methods could not be used with our approach.

In this paper, we will consider the use of the projected preconditioned conjugate method (PPCG) to solve the resulting linear systems. The PPCG method requires the use of a very special form of preconditioner, the *constraint preconditioner*. We derive and analyse both theoretically and by computation a preconditioning approach that leads to the optimal solution of the PDE-constrained optimization problem. That is, a preconditioner which when employed with PPCG gives a solution algorithm which requires  $O(n)$  computational operations to solve a discrete problem with  $n$  degrees of freedom.

### 1.1 Notation

We will use the notation  $\lambda_{\min}(A)$ ,  $\lambda_{\min^+}(A)$  and  $\lambda_{\max}(A)$  to denote the minimum, minimum positive and maximum eigenvalues, respectively, of a matrix  $A$ . We define

$$\min_x^+ (f(x)) = \min \{f(x) : f(x) > 0\}.$$

## 2 Formulation and structure

We have chosen to discretize our problem with finite elements. In order to use these, we require weak formulations of (1.1)–(1.3). We wish to find  $u \in H_g^1 = \{u : u \in H^1(\Omega), u = g \text{ on } \delta\Omega\}$  such that

$$\int_{\Omega} \nabla u \cdot \nabla v = \int_{\Omega} v f \quad \forall v \in H_0^1(\Omega). \quad (2.1)$$

We assume that  $V_0^h \subset H_0^1$  is an  $n$ -dimensional vector space of test functions with basis  $\{\phi_1, \dots, \phi_n\}$ . For the boundary condition to be satisfied, we extend the basis by defining functions  $\phi_{n+1}, \dots, \phi_{n+\delta n}$  and coefficients  $U_j$  so that  $\sum_{j=n+1}^{n+\delta n} U_j \phi_j$  interpolates the boundary data. Then, if  $u_h \in V_g^h \subset H_g^1(\Omega)$ , it is uniquely determined by  $\mathbf{u} = (U_1 \dots U_{n+\delta n})^T$  in

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

Here the  $\phi_i$ ,  $i = 1, \dots, n$ , define a set of shape functions. We also assume that this approximation is conforming, i.e.  $V_g^h = \text{span}\{\phi_1, \dots, \phi_{n+\delta n}\} \subset H_g^1(\Omega)$ . Hence, we obtain the finite-dimensional analogue of (2.1): find  $u_h \in V_g^h$  such that

$$\int_{\Omega} \nabla u_h \cdot \nabla v_h = \int_{\Omega} v_h f \quad \forall v_h \in V_0^h.$$

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

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

since it is well known that  $f_h = 0$  on  $\partial\Omega$ . Thus we can write the discrete analogue of the minimization problem as

$$\min_{u_h, f_h} \frac{1}{2} \|u_h - \hat{u}\|_2^2 + \beta \|f_h\|_2^2 \quad (2.2)$$

$$\text{such that} \quad \int_{\Omega} \nabla u_h \cdot \nabla v_h = \int_{\Omega} v_h f \quad \forall v_h \in V_0^h. \quad (2.3)$$

If  $\hat{u}$  is defined over the whole of  $\Omega$ , we can write the discrete cost functional as

$$\min_{u_h, f_h} \frac{1}{2} \|u_h - \hat{u}\|_2^2 + \beta \|f_h\|_2^2 = \min_{\mathbf{u}, \mathbf{f}} \frac{1}{2} \mathbf{u}^T \bar{M} \mathbf{u} - \mathbf{u}^T \mathbf{b} + \alpha + \beta \mathbf{f}^T M \mathbf{f}, \quad (2.4)$$

where  $\mathbf{u} = (U_1, \dots, U_n)^T$ ,  $\mathbf{f} = (F_1, \dots, F_n)^T$ ,  $\mathbf{b} = \{\int \hat{u} \phi_i\}_{i=1 \dots n}$ ,  $\alpha = \|\hat{u}\|_2^2$ ,  $M = \{\int \phi_i \phi_j\}_{i,j=1 \dots n}$  is a mass matrix, and  $\bar{M} = M$ . If  $\hat{u}$  is only known on part of the domain, then defining

$$\tilde{u}_i = \begin{cases} \hat{u}_i & \text{if } \hat{u}_i \text{ defined} \\ 0 & \text{if } \hat{u}_i \text{ not defined,} \end{cases}$$

we obtain (2.4) where  $\alpha = \|\tilde{u}\|_2^2$ ,  $\mathbf{b}_i = \int \tilde{u} \phi_i$  and

$$\bar{M}_{ij} = \begin{cases} M_{i,j} & \text{if } \hat{u} \text{ is defined at nodes } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases}$$

In this case  $\bar{M}$  will be singular.

We now turn our attention to the constraint: (2.3) is equivalent to finding  $\mathbf{u}$  such that

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

Rearranging we obtain

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

or

$$K\mathbf{u} = M\mathbf{f} + \mathbf{d}, \quad (2.5)$$

where the matrix  $K = \{\int \nabla \phi_i \cdot \nabla \phi_j\}_{i,j=1..n}$  is the discrete Laplacian (the stiffness matrix) and  $\mathbf{d}$  contains the terms coming from the boundary values of  $u_h$ . Thus (2.4) and (2.5) together are equivalent to (2.2) and (2.3).

One way to solve this minimization problem is by considering the Lagrangian

$$\mathcal{L} := \frac{1}{2} \mathbf{u}^T \bar{M} \mathbf{u} - \mathbf{u}^T \mathbf{b} + \alpha + \beta \mathbf{f}^T M \mathbf{f} + \lambda^T (K\mathbf{u} - M\mathbf{f} - \mathbf{d}),$$

where  $\lambda$  is a vector of Lagrange multipliers. Using the stationarity conditions of  $\mathcal{L}$ , we find that  $\mathbf{f}$ ,  $\mathbf{u}$  and  $\lambda$  are defined by the linear system

$$\begin{bmatrix} 2\beta M & 0 & -M \\ 0 & \bar{M} & K^T \\ -M & K & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f} \\ \mathbf{u} \\ \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{b} \\ \mathbf{d} \end{bmatrix}. \quad (2.6)$$

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

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix}, \quad (2.7)$$

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

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 zero blocks,  $K$  and  $M$  are themselves sparse because of the finite element discretization. Thus matrix-vector multiplications can be easily achieved and the work in a symmetric Krylov subspace iteration method will be linear at each iteration (assuming that the work in applying the preconditioner is also linear with problem size).

## 2.1 Properties of $K$ , $M$ and $\bar{M}$

Throughout this paper, we will assume that a shape regular, quasi-uniform division of the domain is used [5] with  $\mathbf{P}_m$  or  $\mathbf{Q}_m$  ( $m \geq 1$ ) finite element approximations. Using these assumptions, we have the following theorem [5, 12]:

**Theorem 2.1** Consider the  $p$ -dimensional problem with  $p \in \{2, 3\}$ . Now

$$\begin{aligned}\lambda_{\min}(K) &= ch^p, & \lambda_{\max}(K) &= Ch^{p-2}, \\ \lambda_{\min}(M) &= dh^p, & \lambda_{\max}(M) &= Dh^p,\end{aligned}$$

where  $c, d, C$  and  $D$  are constants independent of the mesh size  $h$  but dependent on  $p$ . If the target  $\hat{u}$  in (1.1) is only defined on a sub-domain of  $\Omega$ . Then

$$\begin{aligned}\lambda_{\min}(\bar{M}) &= 0, & \lambda_{\max}(\bar{M}) &= \bar{D}h^p, \\ \lambda_{\min^+}(\bar{M}) &= \bar{d}h^p,\end{aligned}$$

where  $\bar{d} \geq d$  and  $\bar{D} \leq D$  are constants independent of the mesh size  $h$  but dependent on  $p$  and  $\hat{\Omega}$ . In particular, if  $\hat{\Omega}_1 \subset \hat{\Omega}_2$ , then  $\bar{d}(\hat{\Omega}_1) \geq \bar{d}(\hat{\Omega}_2)$  and  $\bar{D}(\hat{\Omega}_1) \leq \bar{D}(\hat{\Omega}_2)$ .

## 2.2 The role of $\beta$ in (1.1)

The second term in the cost functionals is added because, in general, the problem will be ill-posed or the control may rapidly vary from one extreme to another over the domain [10] and would often be difficult to impose in real life applications. By varying the value of the regularization parameter  $\beta$ , the balance between the two terms in the cost functionals will be altered. If it is extremely important for  $\|u - \hat{u}\|$  to be very small but we are less concerned by the size of  $\|f\|$ , then a small value of  $\beta$  should be chosen. Conversely, if  $u$  does not need to closely match  $\hat{u}$  but it is important that  $\|f\|$  remains small, then a larger value of  $\beta$  would be used. In practice, a tolerance is often given that determines how small  $\|u - \hat{u}\| / \|\hat{u}\|$  should be. A coarse grid is then used to *cheaply* determine the value of  $\beta$  that corresponds to this tolerance for this grid size: this value of  $\beta$  is then used to solve the problem on the refined mesh [1]. Of course, the coarse grid must be fine enough such that grid refinement is not expected to make a marked difference in terms of the regularization. As we will see in Section 4, there may be instances when the coarse grid has to have a very small mesh size for this to be the case.

## 3 Projected PCG method and constraint preconditioners

As noted in Section 2, the coefficient matrix in (2.6) is of a saddle-point form. In recent years, the projected conjugate gradient (PPCG) method [6] has become an increasingly popular method for solving saddle-point systems that occur within optimization problems. The method requires the use of a preconditioner that has a very specific structure. If, as in (2.7), we write the coefficient matrix  $\mathcal{A}$  of (2.6) as

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

where  $B \in \mathbb{R}^{k \times l}$ , then the preconditioner must take the form

$$\mathcal{P} = \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix},$$

where  $G \in \mathbb{R}^{l \times l}$  is a symmetric matrix. Let  $Z \in \mathbb{R}^{l \times (l-k)}$  be such that its columns span the nullspace of  $B$ . If both  $Z^T A Z$  and  $Z^T G Z$  are positive definite, then the PPCG method can be reliably used. The basic principles behind the PPCG method are as follows. Let  $W \in \mathbb{R}^{l \times k}$  be such that the columns of  $W$  together with the columns of  $Z$  span  $\mathbb{R}^l$ , then any solution  $x^*$  in (2.7) can be written as

$$x^* = Wx_w^* + Zx_z^*. \quad (3.8)$$

Substituting (3.8) into (2.7) and premultiplying the resulting system by  $\begin{bmatrix} W^T & 0 \\ Z^T & 0 \\ 0 & I \end{bmatrix}$ , we obtain

the linear system

$$\begin{bmatrix} W^T A W & W A Z & W B^T \\ Z^T A W & Z^T A Z & 0 \\ B W & 0 & 0 \end{bmatrix} \begin{bmatrix} x_w^* \\ x_z^* \\ y \end{bmatrix} = \begin{bmatrix} W^T \mathbf{c} \\ Z^T \mathbf{c} \\ \mathbf{d} \end{bmatrix}.$$

Therefore, we may compute  $x_w^*$  by solving

$$B W x_w^* = \mathbf{d},$$

and, having found  $x_w^*$ , we can compute  $x_z^*$  by applying the preconditioned conjugate gradient (PCG) method to the system

$$A_{zz} x_z^* = c_z,$$

where

$$\begin{aligned} A_{zz} &= Z^T A Z, \\ c_z &= Z^T (\mathbf{c} - A A X_w^*). \end{aligned}$$

If a preconditioner of the form  $Z^T A Z$  is used, then Gould *et al* [6] suggest terminating the iteration when the easily computable value  $\|x_k - x_z^*\|_{Z^T G Z}$  has sufficiently decreased. Additionally, they show that the PCG algorithm may be rewritten without the need for  $Z$  at all: this results in the PPCG algorithm, Algorithm 3.1.

**Algorithm 3.1** Choose an initial point  $x$  satisfying  $Bx = d$  and compute  $r = Ax - c$ . Solve  $\begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} g \\ v \end{bmatrix} = \begin{bmatrix} r \\ 0 \end{bmatrix}$  and set  $p = -g$ ,  $y = v$ ,  $r \leftarrow r - B^T y$ . Repeat the following steps until a convergence test is satisfied:

$$\begin{aligned} \alpha &= r^T g / p^T A p, \\ x &\leftarrow x + \alpha p, \\ r^+ &= r + \alpha A p, \\ \text{Solve} &\quad \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} g^+ \\ v^+ \end{bmatrix} = \begin{bmatrix} r^+ \\ 0 \end{bmatrix}, \\ \delta &= (r^+)^T g^+ / r^T g, \\ p &\leftarrow -g^+ + \delta p, \\ g &\leftarrow g^+, \\ r &\leftarrow r^+ - B^T v^+. \end{aligned}$$



If  $y^*$  is required, then one extra step must be carried out to compute it. However, in our case,  $y^*$  corresponds to the Lagrange multipliers and from (2.6) we see that  $\lambda = 2\beta\mathbf{f}$ . Note, in Algorithm 3.1,  $\|x_k - x_z^*\|_{Z^T G Z} = r^T g$  (see [6]) and, hence, we can still efficiently calculate Gould *et al*'s suggested measure for termination.

The following theorem gives the main properties of the preconditioned matrix  $\mathcal{P}^{-1}\mathcal{A}$  : the proof can be found in [9].

**Theorem 3.1** Let

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \quad \text{and} \quad \mathcal{P} = \begin{bmatrix} G & B^T \\ B & 0 \end{bmatrix},$$

where  $B \in \mathbb{R}^{k \times l}$  has full rank,  $G \in \mathbb{R}^{l \times l}$  is symmetric and  $\mathcal{P}$  is nonsingular. Let the columns of  $Z \in \mathbb{R}^{l \times (l-k)}$  span the nullspace of  $B$ , then  $\mathcal{P}^{-1}\mathcal{A}$  has

- $2k$  eigenvalues at 1; and
- the remaining eigenvalues satisfy the generalized eigenvalue problem

$$Z^T A Z x_z = \lambda Z^T G Z x_z. \quad (3.9)$$

Keller *et al.* also show that the Krylov subspace

$$\mathcal{K}(\mathcal{P}^{-1}\mathcal{A}; \mathbf{r}) = \text{span}(\mathbf{r}, \mathcal{P}^{-1}\mathcal{A}\mathbf{r}, (\mathcal{P}^{-1}\mathcal{A})^2\mathbf{r}, \dots)$$

will be of dimension at most  $l - k + 2$ , see [9].

Clearly, for our problem (2.7),  $A$  is positive semi-definite and, since  $\mathcal{A}$  is nonsingular,  $Z^T A Z$  is positive definite. It remains for us to show that we can choose a symmetric matrix  $G$  such that  $Z^T G Z$  is positive definite, the eigenvalues of  $\mathcal{P}^{-1}\mathcal{A}$  are clustered, and we can efficiently carry out solves with  $\mathcal{P}$ .

It is straightforward to show that the columns of  $Z = [KM^{-1}, I]^T$  span the nullspace of  $[-M, K]$  and, therefore,

$$Z^T A Z = \bar{M} + 2\beta K^T M^{-1} K.$$

Suppose that we set

$$\mathcal{P} = \left[ \begin{array}{cc|c} 0 & 0 & -M \\ 0 & 2\beta K^T M^{-1} K & K^T \\ \hline -M & K & 0 \end{array} \right],$$

then, if  $z = [z_1^T \ z_2^T \ z_3^T]^T$  and  $r = [r_1^T \ r_2^T \ r_3^T]^T$ , we may solve systems of the form  $\mathcal{P}z = r$  by carrying out the following steps:

$$\text{Solve} \quad M z_3 = -r_1, \quad (3.10)$$

$$\text{Solve} \quad 2\beta K^T M^{-1} K z_2 = r_2 - K^T z_3, \quad (3.11)$$

$$\text{Solve} \quad M z_1 = K z_2 - r_3. \quad (3.12)$$

Systems of the form (3.10) and (3.12) may be solved efficiently because  $M$  is a mass matrix and, hence, we can use a fixed number of iterations of the Chebyshev semi-iteration to solve these systems to a good accuracy [13]. We will discuss the efficient (approximate) solution of (3.11) at the end of this section. Clearly,  $Z^T G Z = K^T M^{-1} K$  is positive definite.

**Theorem 3.2** Let  $r = \text{rank}(\bar{M})$

$$\mathcal{A} = \left[ \begin{array}{cc|c} 2\beta M & 0 & -M \\ 0 & \bar{M} & K^T \\ \hline -M & K & 0 \end{array} \right] \quad \text{and} \quad \mathcal{P} = \left[ \begin{array}{cc|c} 0 & 0 & -M \\ 0 & 2\beta K^T M^{-1} K & K^T \\ \hline -M & K & 0 \end{array} \right],$$

where  $K, M \in \mathbb{R}^{n \times n}$ . The preconditioned matrix  $\mathcal{P}^{-1} \mathcal{A}$  has

- $3n - r$  eigenvalues at 1; and
- the remaining eigenvalues satisfy

$$1 + \frac{\alpha_1}{2\beta} h^4 \leq \lambda \leq 1 + \frac{\alpha_2}{2\beta}.$$

The constants  $\alpha_1$  and  $\alpha_2$  are independent of  $p$  and  $h$  but dependent on  $\hat{\Omega}$ . If  $\hat{\Omega}_1 \subset \hat{\Omega}_2$ , then  $\alpha_1(\hat{\Omega}_1) \geq \alpha_1(\hat{\Omega}_2)$  and  $\alpha_2(\hat{\Omega}_1) \leq \alpha_2(\hat{\Omega}_2)$ .

**Proof.** From Theorem 3.1,  $\mathcal{P}^{-1} \mathcal{A}$  has  $2n$  eigenvalues at 1 and the remaining eigenvalues satisfy

$$(\bar{M} + 2\beta K^T M^{-1} K) x = 2\lambda \beta K^T M^{-1} K x. \quad (3.13)$$

If  $\hat{\Omega} = \Omega$ , then the result follows from [11]. If  $\hat{\Omega} \neq \Omega$ , then  $\bar{M} \neq M$ . If  $\bar{M} x = 0$ , then (3.13) reduces to  $\lambda = 1$ . Therefore, there are a further  $n - r$  eigenvalues at 1. Suppose that  $\bar{M} x \neq 0$ . Rearranging (3.13) gives

$$K^{-1} M K^{-T} \bar{M} x = 2\beta(\lambda - 1)x.$$

Now

$$2\beta(\lambda - 1) \leq \lambda_{\max}(K^{-1} M K^{-T} \bar{M}) \leq \lambda_{\max}(K^{-1} M) \lambda_{\max}(K^{-T} \bar{M}),$$

$$\lambda_{\max}(K^{-1} M) = \max_x \frac{x^T M x}{x^T K x} \leq \frac{D}{c} \quad \text{and} \quad \lambda_{\max}(K^{-T} \bar{M}) = \max_x \frac{x^T \bar{M} x}{x^T K x} \leq \frac{\bar{D}}{c}.$$

Similarly,

$$2\beta(\lambda - 1) \geq \lambda_{\min+}(K^{-1} M K^{-T} \bar{M}) \geq \lambda_{\min}(K^{-1} M) \lambda_{\min+}(K^{-T} \bar{M}),$$

$$\lambda_{\min}(K^{-1} M) = \min_x \frac{x^T M x}{x^T K x} \leq \frac{d}{C} h^2 \quad \text{and} \quad \lambda_{\min+}(K^{-T} \bar{M}) = \min_x \frac{x^T \bar{M} x}{x^T K x} \leq \frac{\bar{d}}{C} h^2.$$

This completes the proof.  $\square$

Therefore, so long as  $\beta \gg \frac{\alpha_1}{2}h^4$ , as we refine the mesh, the bounds in Theorem 3.2 will not get worse. If  $\beta \ll \frac{\alpha_1}{2}h^4$ , then refining the mesh will result in a worsening of the bounds. Of course, at some stage you will reach a point where  $\beta$  will start to dominate the term  $2\beta + \alpha_1h^4$  and the bounds will cease to grow worse as  $h$  is refined. This suggests that this will be an optimal preconditioner for (2.6) if  $\beta$  dominates the term  $2\beta + \alpha_1h^4$ . Additionally, if  $\hat{\Omega}_1 \subset \hat{\Omega}_2$ , we will expect PPCG to converge in less iterations for the case  $\hat{\Omega}_1$  than for the case  $\hat{\Omega}_2$ . However, if  $\beta$  dominates the term  $2\beta + \alpha_1h^4$ , then as the regularization parameter  $\beta$  decreases, the bounds will worsen and we will expect the PPCG method to take more iterations to reach the same tolerance.

It remains for us to consider how we might solve (3.11). Instead of exactly carrying out solves with  $K$ , we may approximate  $K$  by a matrix  $\tilde{K}$ . If our approximation is good enough, then the spectral bounds will be close to those in Theorem 3.2. In the case of our PDE, Poisson's equation, we will employ use a fixed number of multigrid V-cycles. This gives us the preconditioner

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

Here  $\tilde{K}$  denotes the approximation of the solves with  $K$  by two (three for 3D problems) AMG V-cycles applied by using the HSL package HSL\_MI20 [2], and  $\tilde{M}$  denotes 20 iterations of the Chebyshev semi-iterative method.  $\mathcal{P}_{C2}$  is not exactly of the form of a constraint preconditioner since  $\tilde{M}$  is not exactly  $M$ . However,  $\tilde{M}$  is close to  $M$  and we see no deterioration in using PPCG in any of our numerical results. Note the exact use of  $K$  and  $K^T$  in the constraint blocks: this is possible because we only require matrix-vector multiplications with these matrices.

## 4 Numerical Examples

For all of our tests, we discretize the problem with bilinear quadrilateral  $\mathbf{Q}_1$  finite elements. First, we will describe the target functions that we use. We will consider a continuous target that is described over the whole of  $\Omega$ , and a target that is only defined on a sub-domain of  $\Omega$ .

For the 2D and 3D problems, we define  $\Omega = [0, 1]^2$  and  $\Omega = [0, 1]^3$ , respectively. Additionally, let

$$\begin{aligned} \tilde{\Omega}_2 &= \left\{ (x, y) : (x - \frac{5}{8})^2 + (y - \frac{3}{4})^2 \leq \frac{1}{25} \right\}, \\ \tilde{\Omega}_3 &= \left\{ (x, y, z) : (x - \frac{5}{8})^2 + (y - \frac{3}{4})^2 + (z - \frac{7}{10})^2 \leq \frac{1}{16} \right\}. \end{aligned}$$

In Table 4.1, we define the domain  $\hat{\Omega}$  over which the target  $\hat{u}$  is defined.

	$\hat{\Omega}$	$\hat{\Omega}_1$	$\hat{\Omega}_2$	$\hat{u}(x, y) _{\hat{\Omega}_1}$	$\hat{u}(x, y) _{\hat{\Omega}_2}$
Target 1 (2D)	$\hat{\Omega}_1 \cup \hat{\Omega}_2$	$[0, \frac{1}{2}]^2$	$\Omega/\hat{\Omega}_1$	$(2x - 1)^2 (2y - 1)^2$	0
Target 1 (3D)	$\hat{\Omega}_1 \cup \hat{\Omega}_2$	$[0, \frac{1}{2}]^3$	$\Omega/\hat{\Omega}_1$	$(2x - 1)^2 (2y - 1)^2$	0
Target 2 (2D)	$\tilde{\Omega}_1 \cup \tilde{\Omega}_2$	$\tilde{\Omega}_2$	$\partial\Omega$	2	0
Target 2 (3D)	$\hat{\Omega}_1 \cup \hat{\Omega}_2$	$\tilde{\Omega}_3$	$\partial\Omega$	2	0

Table 4.1: Target functions and domains for 2D and 3D distributed control problems

Our test example will take the form:

**Example 4.1** Let  $\Omega = [0, 1]^2$  or  $\Omega = [0, 1]^3$ , and  $\hat{\Omega} \subseteq \Omega$  be the domain over which  $\hat{u}$  is defined. Consider the problem

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

$$\text{s.t.} \quad -\nabla^2 u = f \text{ in } \Omega, \quad (4.1)$$

$$u = \hat{u} \text{ on } \partial\Omega. \quad (4.2)$$

The PPCG method has been implemented in **Fortran 95**: we terminate the method when  $r^T g$  (as defined in Algorithm 3.1) has been reduced by a factor of  $10^{-9}$ . All tests were carried out on a Dell Precision T340 with a single Core2 Quad Q9550 processor (2.83GHz, 1333MHz FSB, 12MB L2 Cache) and 4GB RAM. Codes were compiled using the **NAG f95** compiler with the `-O4` flag and **Goto BLAS**. Our implementation of solves with  $\mathcal{P}$  uses **HSLMA57** [4] to factorise  $M$  and  $K$  and these factorisations are then used to solve systems involving  $M$  and  $K$ ; the implementation of solves with  $\tilde{\mathcal{P}}$  is as described in Section 3. The set-up time for the preconditioners is included in the times below. For large problems and preconditioner  $\mathcal{P}$ , the set-up time completely dominates the overall time to solve the problem; for the preconditioner  $\tilde{\mathcal{P}}$ , this set-up phase takes up 30% of the overall time. We also compare the PPCG method with applying a direct method to the whole of the saddle-point problem: we use **HSLMA57**.

In Table 4.2 we compare the CPU time (in seconds) and the number of iterations to reach our desired level of convergence for different methods when solving 2D and 3D versions of Example 4.1 with Target 1 and  $\beta = 5 \times 10^{-5}$ . As expected, the direct method does not solve the problems in  $\mathcal{O}(n)$  time. Using the PPCG method with preconditioner  $\mathcal{P}$ , we find that the rate of convergence is independent of the mesh size  $h$  but, because we are using a direct method to perform the solves with  $M$  and  $K$ , the amount of time to solve the problem is also not increasing linearly with the problem size. If the PPCG method is used with preconditioner  $\tilde{\mathcal{P}}$ , then there is only a minor increase in the number of iterations over using  $\mathcal{P}$  but the rate of convergence remains independent of  $h$ . Through the use of the Chebyshev semi-iteration to (approximately) solve the systems involving  $M$  and AMG to approximate solves with  $K$ , we find that the time to solve the system grows (almost) linearly with the problem size: this is an optimal preconditioner.

If  $\hat{u}$  is only defined on a subdomain of  $\Omega$ , then our earlier analysis would lead us to expect that the PPCG method with preconditioners  $\mathcal{P}$  and  $\tilde{\mathcal{P}}$  will require fewer iterations to reach the same level of convergence than if  $\hat{u}$  was defined over the whole domain. In Table 4.3 we compare the different methods for solving 2D and 3D versions of Example 4.1 with  $\beta = 5 \times 10^{-5}$  and, in this case, Target 2. Hence,  $\hat{u}$  is only defined over part of the domain. Comparing Tables 4.2 and 4.3, we observe that, as expected, less iterations are required but the two preconditioners are still producing mesh independent convergence results. If the preconditioner  $\tilde{\mathcal{P}}$  is used, then we obtain an optimal preconditioner.

In Figure 4.1, we plot the number of iterations for PPCG with preconditioner  $\tilde{\mathcal{P}}$  to reach our desired level of convergence for different values of the regularisation parameter  $\beta$  and different mesh sizes  $h$ . We consider the 2D version of Example 4.1 with  $\beta = 5 \times 10^{-5}$  and both Targets 1 and 2. As predicted, if  $\beta$  and  $h$  are the same, then the preconditioner  $\tilde{\mathcal{P}}$  performs better on the problem with Target 2. Decreasing  $\beta$  leads to an increase in the number of iterations. If  $\alpha_1 h^4$  dominates the term  $2\beta + \alpha_1 h^4$ , then refining the mesh size results in an increase in the number of iterations. However, at some stage,  $\alpha_1 h^4$  no longer dominates the term  $2\beta + \alpha_1 h^4$ , and the refinement of  $h$  does not effect the convergence of the PPCG method.

$N$	$n$	Direct	PPCG( $\mathcal{P}$ )	PPCG( $\tilde{\mathcal{P}}$ )
8	147	0.002	0.001 (8)	0.003 (9)
16	675	0.01	0.01 (8)	0.011 (9)
32	2883	0.04	0.03 (8)	0.044 (9)
64	11907	0.19	0.12 (8)	0.17 (8)
128	48487	1.59	0.55 (7)	0.72 (8)
256	195075	8.82	3.27 (6)	3.18 (8)
512	783363	53.5	21.5 (6)	14.2 (8)

$N$	$n$	Direct	PPCG( $\mathcal{P}$ )	PPCG( $\tilde{\mathcal{P}}$ )
4	81	0.001	0.002 (7)	0.002 (7)
8	1029	0.04	0.02 (8)	0.05 (8)
16	10125	1.25	0.33 (8)	0.64 (8)
32	89373	38.0	6.61 (7)	7.32 (7)
64	750141	1000+	217 (5)	59.0 (6)

Table 4.2: Comparison of CPU times (in seconds) and iterations to solve Example 4.1 with Target 1 in 2D (left) and 3D right) and  $\beta = 5 \times 10^{-5}$ . We compare a direct method (`hs1_ma57`), PPCG with preconditioner  $\mathcal{P}$ , and PPCG with preconditioner  $\tilde{\mathcal{P}}$ .

$N$	$n$	Direct	PPCG( $\mathcal{P}$ )	PPCG( $\tilde{\mathcal{P}}$ )
8	147	0.002	0.001 (4)	0.002 (4)
16	675	0.01	0.01 (4)	0.007 (4)
32	2883	0.10	0.02 (4)	0.03 (4)
64	11907	0.35	0.10 (4)	0.13 (5)
128	48487	2.78	0.50 (5)	0.53 (5)
256	195075	16.8	3.11 (5)	2.36 (5)
512	783363	147	20.5 (5)	10.3 (5)

$N$	$n$	Direct	PPCG( $\mathcal{P}$ )	PPCG( $\tilde{\mathcal{P}}$ )
4	81	0.001	0.001 (3)	0.001 (3)
8	1029	0.05	0.02 (4)	0.03 (4)
16	10125	1.19	0.31 (5)	0.49 (5)
32	89373	59.2	6.32 (5)	6.00 (5)
64	750141	1000+	219 (5)	58.9 (5)

Table 4.3: Comparison of CPU times (in seconds) and iterations to solve Example 4.1 with Target 2 in 2D (left) and 3D right) and  $\beta = 5 \times 10^{-5}$ . We compare a direct method (`hs1_ma57`), PPCG with preconditioner  $\mathcal{P}$ , and PPCG with preconditioner  $\tilde{\mathcal{P}}$ .

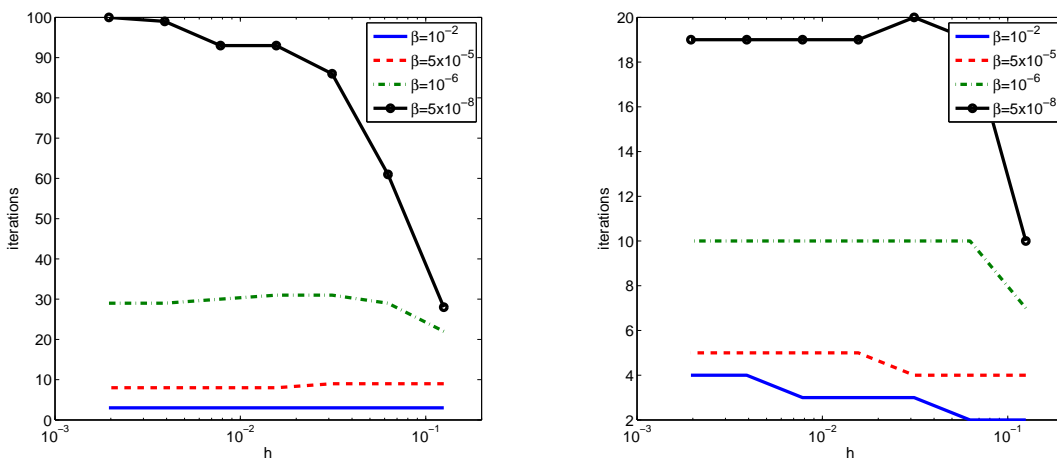


Figure 4.1: Comparison of the number iterations to solve the 2D version of Example 4.1 with  $\beta = 5 \times 10^{-5}$  and either Target 1 (left) or Target 2 (right) with the PPCG method and preconditioner  $\tilde{\mathcal{P}}$ .

## 5 Conclusions

We have presented an optimal preconditioner for distributed control problems where the PDE considered is the Poisson equation. We have considered the cases where the target  $\hat{u}$  is defined over the whole of the domain or just part of the domain. This preconditioner has a block form and uses a fixed number of Chebyshev semi-iterations for some blocks and a fixed number of multigrid cycles for over blocks. The preconditioner is used in conjunction with the projected preconditioned conjugate gradient method for which we have reliable convergence bounds. We have demonstrated its effectiveness for a range of regularization parameters and mesh sizes.

The simplest case of PDE has been considered, namely the Poisson equation, but the inherent block structure of saddle-point problem for other PDEs allows us to use similar preconditioning methods and analysis. We also foresee that these methods will be applicable for problems that include bound constraints.

## Acknowledgment

I would like to thank Tyrone Rees for providing me with numerical test examples that I was able to adapt for use within this paper. I would also like to thank Tyrone, Nick Gould and Andy Wathen for their helpful discussions and valuable suggestions during the process of this work.

## References

- [1] U. M. ASCHER AND E. HABER, *Grid refinement and scaling for distributed parameter estimation problems*, Inverse Problems, 17 (2001), pp. 571–590.
- [2] J. BOYLE, M. D. MIHAJLOVIC, AND J. A. SCOTT, *HSL\_MI20: an efficient AMG preconditioner*, International Journal on Numerical Methods in Engineering, 82 (2009), pp. 64–98.
- [3] S. S. COLLIS AND M. HEINKENSCHLOSS, *Analysis of the streamline upwind/Petrov Galerkin method applied to the solution of optimal control problems*, Tech. Rep. TR02–01, Department of Computational and Applied Mathematics, Rice University, 2002.
- [4] I. S. DUFF, *MA57– a new code for the solution of sparse symmetric definite and indefinite systems*, ACM Transactions on Mathematical Software, 30 (2004), pp. 118–154.
- [5] H. C. ELMAN, D. J. SILVESTER, AND A. J. WATHEN, *Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2005.
- [6] N. I. M. GOULD, M. E. HRIBAR, AND J. NOCEDAL, *On the solution of equality constrained quadratic programming problems arising in optimization*, SIAM J. Sci. Comput., 23 (2001), pp. 1376–1395.
- [7] E. HABER AND U. M. ASCHER, *Preconditioned all-at-once methods for large, sparse parameter estimation problems*, Inverse Problems, 17 (2001), pp. 1847–1864.
- [8] K. ITO AND K. KUNISCH, *Augmented Lagrangian-SQP methods for nonlinear optimal control problems of tracking type*, SIAM J. Control Optim., 34 (1996), pp. 874–891.

- [9] C. KELLER, N. I. M. GOULD, AND A. J. WATHEN, *Constraint preconditioning for indefinite linear systems*, SIAM J. Matrix Anal. Appl., 21 (2000), pp. 1300–1317.
- [10] H. MAURER AND H. D. MITTELMANN, *Optimization techniques for solving elliptic control problems with control and state constraints. I. Boundary control*, Comput. Optim. Appl., 16 (2000), pp. 29–55.
- [11] T. REES, H. S. DOLLAR, AND A. J. WATHEN, *Optimal solvers for PDE-constrained optimization*, SIAM J. Sci. Comput., 32 (2010), pp. 271–298.
- [12] H. S. THORNE, *Properties of linear systems in PDE-constrained optimization. Part I: Distributed control*, Tech. Rep. RAL-TR-2009-017, Rutherford Appleton Laboratory, 2009.
- [13] A. J. WATHEN AND T. REES, *Chebyshev semi-iteration in preconditioning for problems including the mass matrix*, Elect. Trans. Numer. Anal., 34 (2009), pp. 125–135.