Technical Report

TR-2009-027

A Preconditioning Technique for a Class of PDE-Constrained Optimization Problems

by

Michele Benzi, Eldad Haber, Lauren Taralli

Mathematics and Computer Science

EMORY UNIVERSITY

A PRECONDITIONING TECHNIQUE FOR A CLASS OF PDE-CONSTRAINED OPTIMIZATION PROBLEMS

MICHELE BENZI*, ELDAD HABER†, AND LAUREN TARALLI‡

Abstract. We investigate the use of a preconditioning technique for solving linear systems of saddle point type arising from the application of an inexact Gauss–Newton scheme to PDE-constrained optimization problems with a hyperbolic constraint. The preconditioner is of block triangular form and involves diagonal perturbations of the (approximate) Hessian to insure nonsingularity and an approximate Schur complement. We establish some properties of the preconditioned saddle point systems and we present the results of numerical experiments illustrating the preconditioner performance on a model problem motivated by image registration.

Key words. constrained optimization, KKT conditions, saddle point problems, hyperbolic PDEs, Krylov subspace methods, preconditioning, Monge–Kantorovich problem, image registration

AMS subject classifications. Primary 65F08, 65F22, 49M05, 49M15. Secondary 90C30.

1. Introduction. In this paper we consider the solution of a certain class of PDE-constrained optimization problems, i.e., optimization problems with partial differential equations as constraints where the constraint is a hyperbolic PDE. Problems of this kind arise in the description of a multitude of scientific and engineering applications including optimal design, control, and parameter identification [8]. Examples of PDE-constrained optimization problems arise in aerodynamics [31, 36], mathematical finance [10, 15, 16], medicine [4, 27], and geophysics and environmental engineering [2, 1, 28]. PDE-constrained optimization problems are infinite-dimensional and often ill-posed in nature, and their discretization invariably leads to systems of equations that are typically very large and difficult to solve. Developing efficient solution methods for PDE-constrained optimization is an active field of research; see, for instance, [5, 9, 20, 37] and the references therein. We further point to the recent papers [14, 33] for related work with a strong numerical linear algebra emphasis.

The specific problem considered in this paper is a version of the optimal transport problem, where the constraint is a scalar PDE of hyperbolic type. This somewhat special case is actually a good model for a broad class of PDE-constrained optimization problems, including some parameter identification problems. These are inverse problems where the user seeks to recover one or more unknown coefficients in a partial differential equation using some *a priori* knowledge of the solution of that equation. Parameter identification is an important subset of PDE-constrained optimization, due to its widespread occurrence in applications. Here we focus on problems with a first-order hyperbolic equation as the constraint. We use optimal transport as a model problem, motivated by its intrinsic interest for applications and also because it clearly demonstrates some of the main challenges encountered in the numerical solution of

*Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia 30322, USA (benzi@mathcs.emory.edu). The work of this author was supported in part by NSF grant DMS-0511336.

†Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia 30322, USA and Department of Mathematics, University of British Columbia, Vancouver, B. C., Canada V6T 1Z2 (haber@math.ubc.ca). The work of this author was supported in part by DOE under grant DEFG02-05ER25696, and by NSF under grants CCF-0427094, CCF-0728877, DMS-0724759 and DMS-0915121.

‡Quantitative Analytics Research Group, Standard & Poor's, New York, New York 10041, USA (lauren.taralli@gmail.com).

PDE-constrained optimization problems. As we shall see, the fact that the constraint involves only first order derivatives influences our choice of the preconditioner.

The paper is organized as follows. In section 2 we describe the formulation of a broad class of parameter identification problems. In section 3 we discuss the application of an inexact (Gauss–)Newton method to parameter identification problems. Not surprisingly, the most computationally demanding step, and the primary focus of the paper, is the solution of the linear system arising at each Newton iteration. In section 4 we describe a model problem with a hyperbolic PDE as the constraint and its discretization and regularization. Section 5 is devoted to a detailed discussion of a block triangular preconditioner for the linear system solutions. Numerical results are given in section 6, and a few closing remarks in section 7.

2. Formulation of parameter identification problems. In parameter identification one considers the problem of recovering an approximation for a model, or *parameter function*, based on measurements of solutions of a system of partial differential equations. In other words, one is interested in the *inverse problem* of recovering an approximation for a model, $m(\boldsymbol{x})$, based on measurement data b on the solution $u(\boldsymbol{x})$ of the *forward problem*. In general, the forward problem can be linear or nonlinear with respect to u. In this formulation, we consider an important class of problems that share two common features:

1. We assume that the forward problem is linear with respect to u and the PDE can be written as
$$\mathcal{A}(m)u = q, \tag{2.1}$$
where $\mathcal{A}$ is a differential operator which may contain both time and space derivatives and depends on the model $m(\boldsymbol{x})$; the differential problem is defined on an appropriate domain $\Omega \times [0,T] \subset \mathbb{R}^{d+1}$, where $d = 2$ or $d = 3$, and is supplemented with suitable boundary and initial conditions. For simplicity, we assume that there is a unique solution u for any fixed choice of m and q.
2. As we wish to explore relatively simple problems from a PDE standpoint, we assume that the discretization of the problem is "straightforward" and that no "exotic" features are needed such as flux limiters. In this case, the discrete forward problem is continuously differentiable with respect to both u and m.
3. We assume that the constraint is a first-order hyperbolic PDE.

Although our assumptions may look highly restrictive, problems that satisfy the first two assumptions constitute a large variety of applications such as electromagnetic inversion (of high frequencies), hydrology and diffraction tomography; see [11, 12, 18, 32, 37] and references therein. The third assumption characterizes the class of problems we focus on in this paper.

Given the forward problem for u, we define an operator Q to be the projection of u onto the locations in Ω (or $\Omega \times [0,T]$) to which the data b are associated. Thus, we can interpret the data as a nonlinear function of the model m:
$$b = Q\mathcal{A}(m)^{-1}q + \varepsilon\ . \tag{2.2}$$

Here, ε is the measurement noise. Because the data are finite and noisy, the inverse problem of recovering m is ill-posed. For this reason, a process of regularization is required to recover a relatively smooth, locally unique solution to a nearby problem; for details, see [38].

In this paper we employ Tikhonov regularization. More precisely, the inverse problem to approximate m becomes a minimization problem of the form

$$\min_m \ \frac{1}{2}\|Q\mathcal{A}(m)^{-1}q - b\|^2 + \alpha R(m - m_r) \ , \tag{2.3}$$

where m_r is a reference model and $\alpha > 0$ is the regularization parameter. A commonly used form of the regularization functional R is

$$R(m) = \frac{1}{2}\int_\Omega (\beta\, m^2 + |\nabla m|^2)\, d\boldsymbol{x} \ , \tag{2.4}$$

where β is a constant and $|\cdot|$ denotes the Euclidean length of a vector in $\mathbb{R}^d$. Our approach can be extended to other regularizers, but for the sake of brevity we do not discuss this here.

The formulation (2.3) implies that the PDE is eliminated to obtain an unconstrained optimization problem. However, solving the PDE in practice can be challenging, and eliminating the PDE at an early stage may prove to be computationally inefficient. We therefore consider the equivalent constrained formulation:

$$\min_{u,m} \ \frac{1}{2}\|Qu - b\|^2 + \alpha\, R(m - m_r) \tag{2.5a}$$

$$\text{s.t. } \mathcal{A}(m)u - q = 0 \ . \tag{2.5b}$$

The optimization problem (2.5) is an equality-constrained optimization problem. In many applications, simple bound constraints on m are added. For the sake of simplicity, in this paper we will not include bound constraints on m.

The above constrained optimization problem is infinite-dimensional. In order to obtain a finite-dimensional optimization problem that can be solved on a computer, the problem is discretized using, for instance, finite differences or finite elements. The discrete equality-constrained optimization problem is written as

$$\min_u \quad J(u) \tag{2.6a}$$

$$\text{s.t. } C(u) = 0, \tag{2.6b}$$

where $u \in \mathbb{R}^n$, $J : \mathbb{R}^n \longrightarrow \mathbb{R}$, and $C : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ with $m \leq n$. We impose the following restrictions on the equality-constrained optimization problem (2.6) for simplicity of presentation. First, we require that there are no redundant constraints. Second, we assume the objective function J to be twice continuously differentiable. Finally, we require that the Hessian of J, denoted by J_{uu}, is symmetric positive semidefinite. Note that such restrictions are common to many algorithms for equality-constrained optimization, and are often satisfied in practice. As we will see, these restrictions allow us to make use of an inexact variant of Newton's method, described in the next section.

3. Inexact Newton method. Inexact Newton methods are widely used in the solution of constrained optimization problems; see, for example, [1, 9, 13, 20, 37, 38]. Here we give a brief description of this class of algorithms; see [30] for a thorough treatment.

To solve (2.6) via an (inexact) Newton method, we first introduce the Lagrangian function:

$$\mathcal{L}(u, p) = J(u) + p^\top C(u) \ . \tag{3.1}$$

Here, $p \in \mathbb{R}^m$ is a vector of Lagrange multipliers. Next, a necessary condition for an optimal solution of (2.6) is to satisfy the *Karush-Kuhn-Tucker (KKT) conditions*:

$$\mathcal{L}_u = J_u + C_u^\top p = 0 \,, \tag{3.2a}$$
$$\mathcal{L}_p = C(u) \qquad = 0 \,, \tag{3.2b}$$

where J_u and C_u denote the gradient of J and the Jacobian of u, respectively. To solve system (3.2), Newton's method can be used, leading to a sequence of symmetric indefinite linear systems of the form

$$\begin{pmatrix} J_{uu} & C_u^\top \\ C_u & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta p \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_u \\ \mathcal{L}_p \end{pmatrix} \,. \tag{3.3}$$

The coefficient matrix in (3.3) is known as a *saddle point* (or *KKT*) matrix. The term "saddle point" comes from the fact that a solution to (3.3), say $(\delta u_*, \delta p_*)$, is a saddle point for the Lagrangian. In other words,

$$\min_{\delta u} \max_{\delta p} \mathcal{L}(\delta u, \delta p) = \mathcal{L}(\delta u_*, \delta p_*) = \max_{\delta p} \min_{\delta u} \mathcal{L}(\delta u, \delta p) \,. \tag{3.4}$$

Therefore, the main computational step in the solution process is the repeated solution of large linear systems in saddle point form. For an extensive review of solution methods for saddle point problems, we refer to [7].

An important ingredient in any inexact Newton method is the line search. Suppose that, given an iterate $[u_k; p_k]$, we have solved (3.3) to determine a direction for the step $[\delta u; \delta p]$, then the step length should be chosen so that the next iterate $[u_{k+1}; p_{k+1}] = [u_k; p_k] + \alpha_k [\delta u; \delta p]$ leads to the largest possible decrease of $\mathcal{L}(u, p)$. In other words, the step length α_k is chosen to satisfy

$$\min_{\alpha_k} \mathcal{L}(u_k + \alpha_k \delta u_k,\ p_k + \alpha_k \delta p_k) \,. \tag{3.5}$$

An exact computation of α_k that minimizes the function is expensive and unnecessary; for this reason, most line search methods find a loose approximation of the actual value of α_k that minimizes $\mathcal{L}(u_k + \alpha_k \delta u_k,\ p_k + \alpha_k \delta p_k)$. For details on different line search methods, see [30]. The inexact Newton algorithm to solve (2.6) is given in Algorithm 1 below.

Algorithm 1 Inexact Newton method to solve (2.6)

– Initialize u_0 and p_0;
for $k = 1, 2, \ldots$ **do**
– Compute $\mathcal{L}_u$, $\mathcal{L}_p$, and J_{uu};
– Approximately solve (3.3) to a given tolerance;
– Use a line search to accept or reject step:

$$\begin{pmatrix} u_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} u_k \\ p_k \end{pmatrix} + \alpha_k \begin{pmatrix} \delta u \\ \delta p \end{pmatrix}$$

– Test for termination, set $k \leftarrow k + 1$;
end for

Note that the term "inexact" refers to the approximate solution to (3.3) at each outer (Newton) iteration. An approximate solution of the linear system will generally

suffice to obtain a satisfactory search direction at each step. While an inexact linear solution may increase the number of outer iterations required to reach a given level of accuracy, the amount of work per Newton iteration can be greatly reduced, leading to an overall faster solution process. Choosing an appropriate stopping tolerance for the linear system solver will hopefully minimize the work required to compute the solution to the optimization problem.

Before applying the inexact Newton method to parameter identification problems we must discretize (2.5), so that we are solving a discrete constrained optimization problem of the form (2.6). First, we discretize the PDE constraint (2.1) using, e.g., finite differences and/or finite elements to obtain

$$A(m)u = q\,, \tag{3.6}$$

where A is a nonsingular matrix, u is the grid function approximating $u(\boldsymbol{x})$ (or $u(t,\boldsymbol{x})$) and arranged as a vector, and m and q likewise relate to $m(\boldsymbol{x})$ and $q(\boldsymbol{x})$. We discretize the regularization functional (2.4) similarly, so that

$$R(m-m_r) \approx \frac{1}{2}\|L(m-m_r)\|^2,$$

where L is a matrix not dependent on m. The resulting optimization problem is written in constrained form as

$$\min_{u,m} \frac{1}{2}\|Qu-b\|^2 + \frac{1}{2}\alpha\|L(m-m_r)\|^2 \tag{3.7a}$$

$$\text{s.t. } A(m)u - q = 0\,. \tag{3.7b}$$

Clearly, the discrete constrained optimization problem (3.7) is of the form (2.6), and we can apply an inexact Newton method to compute its solution. We begin by forming the Lagrangian,

$$\mathcal{L}(u,m,p) = \frac{1}{2}\|Qu-b\|^2 + \frac{1}{2}\alpha\|L(m-m_r)\|^2 + p^\top V(A(m)u-q)\,, \tag{3.8}$$

where p is a vector of Lagrange multipliers and V is a (mass) matrix such that for any functions $w(\boldsymbol{x}), p(\boldsymbol{x})$ and their corresponding grid functions w and p,

$$\int_\Omega p(\boldsymbol{x})w(\boldsymbol{x})\,d\boldsymbol{x} \approx p^\top V w.$$

Insertion of the matrix V in (3.8) is necessary, as this allows for the vector of Lagrange multipliers p to be interpreted as a grid function. It is important to note that 'standard' optimization algorithms do not require the matrix V. However, if we intend to keep the meaning of the grid function p as a discretization of a continuous (i.e., infinite-dimensional) Lagrange multiplier $p(\boldsymbol{x})$, the matrix V is indispensable. Note that when the problem is discretized (as we do here) by a simple finite difference scheme on a uniform grid with equal grid spacing in all d space directions, V is simply a scaled identity matrix.

The Euler–Lagrange equations associated with the above Lagrangian (i.e., the necessary conditions for an optimal solution of (3.7)) are

$$\mathcal{L}_u = Q^\top(Qu-b) + A(m)^\top Vp = 0, \tag{3.9a}$$

$$\mathcal{L}_m = \alpha L^\top L(m-m_r) + G(u,m)^\top Vp = 0, \tag{3.9b}$$

$$\mathcal{L}_p = V(A(m)u-q) = 0, \tag{3.9c}$$

where $G(u,m) = \partial(A(m)u)/\partial m$. This is in general a nonlinear system of equations. A Newton linearization for solving the nonlinear equations (3.9) leads to a linear KKT system at each iteration, of the form

$$\begin{pmatrix} Q^\top Q & * & A^\top V \\ * & \alpha L^\top L + * & G^\top V \\ VA & VG & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta m \\ \delta p \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_u \\ \mathcal{L}_m \\ \mathcal{L}_p \end{pmatrix}, \tag{3.10}$$

where the blocks denoted by '$*$' correspond to mixed second-order derivative terms. A common strategy is to simply ignore these terms, leading to a Gauss–Newton scheme. With this approximation, the linear systems to be solved at each step take the simpler form

$$\begin{pmatrix} Q^\top Q & 0 & A^\top V \\ 0 & \alpha L^\top L & G^\top V \\ VA & VG & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta m \\ \delta p \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_u \\ \mathcal{L}_m \\ \mathcal{L}_p \end{pmatrix}. \tag{3.11}$$

Although the rate of convergence of the Gauss–Newton method is guaranteed to be only linear rather than quadratic (as for an 'exact' Newton method), this approach is advantageous both because of the reduced cost per iteration and because in practice the rate of convergence is often faster than linear. Hence, in the remainder of the paper we will consider only the Gauss–Newton approach.

4. A problem with hyperbolic constraint. We consider a problem of the form (2.6) where the constraint corresponds to a hyperbolic forward problem with smooth initial data. The problem we consider can be regarded as a simplified version of the *Monge–Kantorovich (MKP) mass transfer problem*. The MKP frequently arises in many diverse fields such as optics, fluid mechanics, and image processing; see [3, 6], and references therein. The original transport problem was proposed by Monge in 1781, and consisted of finding how best to move a pile of soil ("deblais") to an excavation ("remblais") with the least amount of work. A modern formulation and generalization was given by Kantorovich during the 1940s; see [24, 25]. A large body of work, both theoretical and computational, has occurred in optimal mass transport. One particularly important development in recent years occurred in 2000, when Benamou and Brenier reformulated the Monge–Kantorovich mass transfer problem as a computational fluid mechanics (CFD) problem [6]. This formulation can allow for an efficient and robust numerical solver to be applied to solve the MKP; in particular, the fluid mechanics formulation opens the door to the application of solution techniques from PDE-constrained optimization and CFD.

While a simplification of the actual CFD formulation of the MKP, our formulation is still closely related to the Monge–Kantorovich mass transfer problem. The model problem considered here captures most of the difficulties inherent in MKP. It is also a useful model for applications in image processing.

4.1. Problem formulation. Let Ω be a bounded domain in $\mathbb{R}^d$ with a sufficiently smooth boundary. Consider two given bounded density functions $u_0(\boldsymbol{x}) \geq 0$ and $u_T(\boldsymbol{x}) \geq 0$ for $\boldsymbol{x} \in \Omega$. In Monge's original transport problem, u_0 described the density of the pile of soil, and u_T described the density of the excavation or fill. In an image processing application, u_0 and u_T describe the pixel intensities of two images that we seek to register to one another, see [21]. Integration of a density function over Ω yields the mass. In the classical Monge–Kantorovich problem, the density functions are assumed to have equal masses; in our formulation, density functions are allowed

to yield masses that are not exactly equal. In other words, in our model we only assume that

$$\int_\Omega u_0(\boldsymbol{x})\, d\boldsymbol{x} \approx \int_\Omega u_T(\boldsymbol{x})\, d\boldsymbol{x}. \tag{4.1}$$

Given these two masses, we wish to find a mapping from one density to the other that is optimal (in some sense). We define this optimal mapping $\varphi : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ to be the minimizer of the L^2 Kantorovich distance between u_0 and u_T; that is, we wish to find

$$\min_{\varphi} \int_\Omega |\varphi(\boldsymbol{x}) - \boldsymbol{x}|^2 u_0(\boldsymbol{x})\, d\boldsymbol{x} \tag{4.2}$$

among all maps φ that transport u_0 to u_T. In the original Monge problem, this corresponds to minimizing the work (described by the map φ) required to move the pile of dirt into the excavation (of equal size to the pile). In the image registration problem the aim is to establish (as accurately as possible) a point-by-point correspondence via the map φ between two images of a scene.

To set the problem in a PDE-constrained optimization framework we follow the approach proposed in [6], where it is shown that finding the solution to (4.2) is equivalent to the following optimization problem. Introduce a time interval $[0, T]$. For simplicity, we assume the problem is two-dimensional ($d = 2$) and that $\Omega = [0, 1] \times [0, 1]$. We seek a smooth, time-dependent density field $u(t, \boldsymbol{x})$ and a smooth, time-dependent velocity field $m(t, \boldsymbol{x}) = (m_1(t, \boldsymbol{x}), m_2(t, \boldsymbol{x}))$ that satisfy

$$\min_{u,m} \frac{1}{2} \|u(T, \boldsymbol{x}) - u_T(\boldsymbol{x})\|^2 \; + \; \frac{1}{2}\, \alpha\, T \int_\Omega \int_0^T u\, \|m\|^2\, dt\, d\boldsymbol{x} \tag{4.3a}$$

$$\text{s.t. } u_t + \nabla \cdot (um) = 0\,, \tag{4.3b}$$

$$u(0, \boldsymbol{x}) = y_0\,. \tag{4.3c}$$

Equation (4.3) is an infinite-dimensional PDE-constrained optimization problem in which the PDE constraint is a hyperbolic transport equation. The next section describes the finite difference discretization of the components of (4.3) used to obtain a finite-dimensional constrained optimization problem.

4.2. Discretization. As we mentioned in section 2, we wish to restrict our problems to those in which the discrete forward problem is continuously differentiable with respect to both u and m. As a result, we restrict our attention to problems in which both the initial and final densities are smooth. In this case, standard discretization techniques can be used.

Since the velocity field m is not known *a priori*, it is difficult to choose appropriate time steps to ensure stability of the scheme for explicit discretization. There are several implicit discretization schemes available for the forward problem. We choose an implicit Lax–Friedrichs scheme to discretize (4.3b) in order to have a stable discretization [29].

We discretize the time interval $[0, T]$ using n_t equal time steps, each with width $h_t = T/n_t$. Next, we discretize the spatial domain $\Omega = [0, 1] \times [0, 1]$ using n_x grid points in each direction, so that the side of each cell has length $h_x = 1/(n_x - 1)$. Once the domain has been discretized, for each time step t_k we form the vectors u^k, m_1^k, and m_2^k corresponding to $u(t_k, \boldsymbol{x})$, $m_1(t_k, \boldsymbol{x})$, and $m_2(t_k, \boldsymbol{x})$, respectively, where the unknowns are cell-centered. Using $u_{i,j}^k$ to denote the unknown in the vector

corresponding to $u(t_k, \boldsymbol{x}_{i,j})$ (and a similar notation for m_1 and m_2), we can write the finite difference approximations for the implicit Lax–Friedrichs scheme as follows:

$$\left(\frac{\partial u}{\partial t}\right)^k_{i,j} \approx \frac{1}{h_t}\left[u^{k+1}_{i,j} - \frac{1}{4}(u^k_{i+1,j} + u^k_{i-1,j} + u^k_{i,j+1} + u^k_{i,j-1})\right] , \tag{4.4a}$$

$$(\nabla \cdot (um))^k_{i,j} \approx \tag{4.4b}$$

$$\frac{1}{2h_x}\left[(u \odot m_1)^{k+1}_{i+1,j} - (u \odot m_1)^{k+1}_{i-1,j} + (u \odot m_2)^{k+1}_{i,j+1} - (u \odot m_2)^{k+1}_{i,j-1}\right] ,$$

where the symbol $\odot$ denotes the (componentwise) Hadamard product. Assuming periodic boundary conditions, a common assumption for this type of problem, this scheme can be expressed in matrix form as follows:

$$\frac{1}{h_t}\left[u^{k+1} - Mu^k\right] + B(m^{k+1})u^{k+1} = 0, \tag{4.5}$$

where M corresponds to an averaging matrix and $B(m)$ is the matrix which contains difference matrices in each direction. After rearranging (4.5), the system to solve at each time step is:

$$C^{k+1}u^{k+1} = Mu^k \quad \text{where} \quad C^{k+1} := I + h_t B(m^{k+1}), \quad k = 0, 1, \ldots, n_t - 1. \tag{4.6}$$

The set of equations (4.6) can be rewritten in a more compact form as

$$A(m)u = \begin{pmatrix} C(m^1) & & & \\ -M & C(m^2) & & \\ & \ddots & \ddots & \\ & & -M & C(m^{n_t}) \end{pmatrix} \begin{pmatrix} u^1 \\ u^2 \\ \vdots \\ u^{n_t} \end{pmatrix} = \begin{pmatrix} Mu^0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = q , \tag{4.7}$$

where u^0 is the vector obtained after discretizing the given density function u_0 consistently. The discretization of the forward problem (4.3) is now complete.

4.3. Jacobians. To compute the Jacobian $G(u, m) = \partial(A(m)u)/\partial m$, we first examine the structure of the difference matrix $B(m^k)$:

$$B(m^k) = \begin{pmatrix} D_1 & D_2 \end{pmatrix} \begin{pmatrix} \operatorname{diag}(m_1^k) \\ \operatorname{diag}(m_2^k) \end{pmatrix}, \tag{4.8}$$

where for a given vector v the notation $\operatorname{diag}(v)$ is used to denote the diagonal matrix with the entries of v on the main diagonal, and D_1 and D_2 denote central difference matrices in each direction. As a result, we can compute the Jacobian of $A(m)u$ with respect to m:

$$G(u, m) = \frac{\partial(A(m)u)}{\partial m} = \begin{pmatrix} G^1 & & & \\ & G^2 & & \\ & & \ddots & \\ & & & G^{n_t} \end{pmatrix},$$

$$\text{where} \quad G^k = \frac{\partial(C(m^k)u^k)}{\partial m^k} = h_t \begin{pmatrix} D_1 & D_2 \end{pmatrix} \begin{pmatrix} \operatorname{diag}(u^k) & \\ & \operatorname{diag}(u^k) \end{pmatrix} .$$

The Jacobian with respect to u is trivial. Now that the components of the forward problem (4.3) and its derivatives have been defined, we can discretize the remaining components of the problem.

4.4. Data and regularization. To represent the objective function (4.3a) in discrete form, we first define some matrices and vectors. Let

$$m = \begin{pmatrix} m_1^1 \\ m_2^1 \\ m_1^2 \\ m_2^2 \\ \vdots \\ m_1^{n_t} \\ m_2^{n_t} \end{pmatrix}, \tag{4.10a}$$

$$L = \begin{pmatrix} I & I & & & \\ & & I & I & \\ & & & \ddots & \ddots \\ & & & & I \quad I \end{pmatrix}, \quad \text{and} \quad Q = h_x \begin{pmatrix} 0 & \dots & 0 & I \end{pmatrix}, \tag{4.10b}$$

where I is the $n_x^2 \times n_x^2$ identity matrix. Note that we include the grid spacing h_x into the matrix Q to ensure grid independence in the data fitting term. Also, let b be the vector obtained after discretizing the density function u_T consistently (taking scaling into account). Then it is easy to show that the discrete representation of (4.3a) is

$$\frac{1}{2}\|Qu - b\|^2 + \frac{1}{2}\,\alpha\, T\, h_t\, h_x^2\, u^\top L\, \mathrm{diag}(m)\, m\,. \tag{4.11}$$

Combining the expressions (4.3) and (4.11), the discrete optimization problem becomes

$$\min_{u,m} \ \frac{1}{2}\,\|Qu - b\|^2 + \frac{1}{2}\,\xi\, u^\top L\, \mathrm{diag}(m)\, m \tag{4.12a}$$

$$\text{s.t.} \qquad A(m)u - q = 0. \tag{4.12b}$$

Here, $\xi = \alpha\, T\, h_t\, h_x^2$. The Lagrangian associated with (4.12) is

$$\mathcal{L}(u, m, p) = \frac{1}{2}\|Qu - b\|^2 + \frac{1}{2}\,\xi\, u^\top L\, \mathrm{diag}(m)\, m + p^\top V(A(m)u - q)\,, \tag{4.13}$$

where p is a vector of Lagrange multipliers and V is the diagonal matrix that allows for p to be interpreted as a grid function that discretizes a continuous Lagrange multiplier $p(\boldsymbol{x})$. Although V is just the scaled identity matrix for the simple space discretization used here, we use V in the subsequent formulas for the sake of generality. A necessary condition for an optimal solution of our problem is expressed as

$$\mathcal{L}_u = Q^\top(Qu - b) + \frac{1}{2}\,\xi\, L\, \mathrm{diag}(m)\, m + A(m)^\top V p = 0, \tag{4.14a}$$

$$\mathcal{L}_m = \xi\, \mathrm{diag}(L^\top u)\, m + G(u, m)^\top V p = 0, \tag{4.14b}$$

$$\mathcal{L}_p = V(A(m)u - q) = 0, \tag{4.14c}$$

where $G(u, m)$ was defined in section 4.3. Next, using a Gauss–Newton approximation (as described in section 3), we obtain a sequence of KKT systems of the form

$$\mathcal{H}s = \begin{pmatrix} Q^\top Q & 0 & A^\top V \\ 0 & \xi\, \mathrm{diag}(L^\top u) & G^\top V \\ VA & VG & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta m \\ \delta p \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_u \\ \mathcal{L}_m \\ \mathcal{L}_p \end{pmatrix}. \tag{4.15}$$

Comparing (4.15) with (3.11), we can see that the main difference is in the $(2,2)$ block, which is due to the slightly different regularization term in (4.12). The next section will present a technique to solve systems of the form (4.15), taking into account the structure of each block in the coefficient matrix.

5. The preconditioner. The essential ingredient in the overall solution process is an efficient solver for large, sparse, symmetric indefinite linear systems of the form (4.15), where the components of the matrix are defined as in the previous section. For simplicity, we will assume a uniform grid is used, so that $V = h^2 h_t I$.

Most of the work on preconditioning for PDE-constrained optimization is based on the use of approximations to the *reduced Hessian*; see, e.g., [20, 26]. This is appropriate for problems where the constraint is a second-order PDE. Here, on the other hand, we have a first-order constraint, which suggests a different approach. Let us rewrite (4.15) so that we are solving the following saddle point system with components $\tilde{A} \in \mathbb{R}^{n\times n}$ and $\tilde{B} \in \mathbb{R}^{m\times m}$, where $m < n$:

$$\mathcal{H}\, s = \begin{pmatrix} \tilde{A} & \tilde{B}^\top \\ \tilde{B} & 0 \end{pmatrix} \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \end{pmatrix} \;=\; r\,, \tag{5.1}$$

where

$$\tilde{A} = \begin{pmatrix} Q^\top Q & 0 \\ 0 & \xi \mathrm{diag}(L^\top u) \end{pmatrix}, \quad \text{and} \quad \tilde{B} = \begin{pmatrix} VA & VG \end{pmatrix}.$$

As a result of the construction of the matrices Q and L in (4.10), we can see that $\xi \mathrm{diag}(L^\top u)$ is a diagonal matrix with positive diagonal entries, and $Q^\top Q$ is diagonal with zeros on the diagonal for the first $(n_t - 1)$ blocks (each with size $n_x^2 \times n_x^2$), and a positive multiple of the identity matrix in the last block of size $n_x^2 \times n_x^2$. Consequently, $\tilde{A}$ is a diagonal matrix such that the first $n_z = (n_t - 1)\, n_x^2$ diagonal entries are zero, and the remaining diagonal entries are nonzero and positive. Therefore we can rewrite $\mathcal{H}$ as follows:

$$\mathcal{H} = \begin{pmatrix} 0 & 0 & B_1^\top \\ 0 & A_{22} & B_2^\top \\ B_1 & B_2 & 0 \end{pmatrix}, \tag{5.2}$$

where B_1 is a matrix containing the first n_z columns of $\tilde{B}$, and B_2 contains the remaining $n - n_z$ columns of $\tilde{B}$. We observe that $\mathcal{H}$ is invertible since $\tilde{B}$ has full row rank and $\mathrm{Ker}(\tilde{A}) \cap \mathrm{Ker}(\tilde{B}) = \{0\}$; see, e.g., [7, Theorem 3.2].

An "ideal" preconditioner for solving (5.1) for A invertible is the block triangular matrix

$$\mathcal{P}_{id} = \begin{pmatrix} \tilde{A} & \tilde{B}^\top \\ 0 & -\tilde{S} \end{pmatrix}, \quad \text{where} \quad \tilde{S} = \tilde{B}\tilde{A}^{-1}\tilde{B}^\top. \tag{5.3}$$

The matrix $-\tilde{S}$ is known as the Schur complement. Note that if $\tilde{A}$ is symmetric positive definite and B is full rank, $\tilde{S}$ is positive definite and therefore invertible. With $\mathcal{P}_{id}$ as a preconditioner, an optimal Krylov subspace method like GMRES [35] is guaranteed to converge in two steps; see, e.g., [7, 17]. In our case, $\tilde{A}$ is symmetric positive semidefinite and singular; hence, we cannot apply this technique directly to the saddle point problem (5.1). With this in mind, we define a block triangular

preconditioner in which $\tilde{A}$ is replaced by the diagonally perturbed matrix A_p defined as

$$A_p = \begin{pmatrix} \gamma I & 0 \\ 0 & A_{22} \end{pmatrix}, \tag{5.4}$$

where $\gamma > 0$ and I is the $n_z \times n_z$ identity matrix. Observe that A_p is nonsingular and easily invertible; A_p is also positive definite by the formulation of the problem. Note that, if we were to replace $\tilde{A}$ with A_p in the coefficient matrix of (5.1), then we would have an invertible matrix to which we will refer as the *perturbed Hessian*. However, the actual Hessian is not perturbed; only the preconditioner is. In summary, we propose using the following block triangular preconditioner for solving (5.1):

$$\mathcal{P} = \begin{pmatrix} A_p & \tilde{B}^\top \\ 0 & -S \end{pmatrix} = \begin{pmatrix} \gamma I & 0 & B_1^\top \\ 0 & A_{22} & B_2^\top \\ 0 & 0 & -S \end{pmatrix} \tag{5.5}$$

where $S = \tilde{B} A_p^{-1} \tilde{B}^\top = \frac{1}{\gamma} B_1 B_1^\top + B_2 A_{22}^{-1} B_2^\top$ is the Schur complement of the perturbed Hessian. Note that S is symmetric positive definite for all $\gamma > 0$. In practice, solving linear systems involving S can be expensive; however, exact solves are not necessary. In an actual implementation, an application of $\mathcal{P}^{-1}$ to a vector will involve an approximate inversion of the Schur complement S, typically achieved by some iterative scheme. We note that since the constraint is a first-order PDE, the Schur complement resembles an elliptic second-order partial differential operator, for which many solvers and preconditioners exist. We will return to this point in section 6.

5.1. Spectral properties of the preconditioned matrix. In this subsection we establish some properties of the preconditioned matrix $\mathcal{H}\mathcal{P}^{-1}$, assuming that linear systems associated with S are solved exactly. Theorem 5.2 below will give us an indication of the spectral properties of the preconditioned matrix, which in turn is an indication for the quality of the preconditioner in practice, assuming linear systems involving S are solved sufficiently accurately. We begin with a useful Lemma.

LEMMA 5.1. *Let J and K be two symmetric positive semidefinite matrices such that $J + \gamma K$ is symmetric positive definite for all $\gamma > 0$. Then the matrix*

$$P = \lim_{\gamma \to 0^+} J(J + \gamma K)^{-1}$$

is a projector onto the range of J.

Proof. It suffices to show that $P = \lim_{\gamma \to 0^+} J(J + \gamma K)^{-1}$ has the following three properties:

1. P is diagonalizable.
2. The eigenvalues of P are 0 and 1.
3. $\operatorname{rank}(P) = \operatorname{rank}(J)$.

Then it will follow that P is a (generally oblique) projector onto the range of J. First, to show that P is diagonalizable, we use a special case of Theorem 8.7.1 in [19]. The statement of interest is as follows: if J and K are symmetric and positive semidefinite, then there exists a nonsingular matrix W such that both $D = W^\top J W$ and $E = W^\top K W$ are diagonal. Then observe that, for all $\gamma > 0$,

$$\begin{aligned} W J (J + \gamma K)^{-1} W^{-1} &= W J W^\top W^{-\top} (J + \gamma K)^{-1} W^{-1} \\ &= (W J W^\top)(W (J + \gamma K) W^\top)^{-1} \\ &= (W J W^\top)(W J W^\top + \gamma W K W^\top)^{-1} \\ &= D(D + \gamma E)^{-1} . \end{aligned}$$

Hence, we have shown that there exists a nonsingular matrix W, *not dependent upon* γ, that yields a similarity transformation of $J(J+\gamma K)^{-1}$ to a diagonal matrix. In particular,

$$\begin{aligned} WPW^{-1} &= W\left(\lim_{\gamma\to 0^+} J(J+\gamma K)^{-1}\right)W^{-1} \\ &= \lim_{\gamma\to 0^+} WJ(J+\gamma K)^{-1}W^{-1} \\ &= \lim_{\gamma\to 0^+} D(D+\gamma E)^{-1}\,, \end{aligned}$$

which is diagonal. Therefore P is diagonalizable.

Now, since P is similar to $\lim_{\gamma\to 0^+} D(D+\gamma E)^{-1}$, we can easily compute the eigenvalues of P by computing the eigenvalues of $\lim_{\gamma\to 0^+} D(D+\gamma E)^{-1}$. In particular, consider the i^{th} entry of the diagonal matrix:

$$\begin{aligned} \left[\lim_{\gamma\to 0^+} D(D+\gamma E)^{-1}\right]_i &= \lim_{\gamma\to 0^+} \frac{d_i}{d_i+\gamma e_i} \\ &= \begin{cases} 0 & \text{if } d_i = 0\,, \\ 1 & \text{otherwise.} \end{cases} \end{aligned}$$

It follows that the eigenvalues of the matrix $\lim_{\gamma\to 0^+} D(D+\gamma E)^{-1}$ are 0 and 1, and in turn, the eigenvalues of P are 0 and 1.

Finally, to show the third property, note that $\operatorname{rank}(P) = \operatorname{rank}(D)$, and since $D = W^\top JW$, $\operatorname{rank}(P) = \operatorname{rank}(J)$. Therefore, all three properties hold for P, and the proof of the lemma is complete. □

THEOREM 5.2. *Suppose $\mathcal{H}$ and $\mathcal{P}$ are defined as in* (5.2) *and* (5.5)*, respectively. Let $\mathcal{A} = \mathcal{A}(\gamma) = \mathcal{H}\mathcal{P}^{-1}$. Then* 1 *is an eigenvalue of $\mathcal{A}(\gamma)$ for all γ, with algebraic multiplicity at least $n - n_z$. Furthermore, as $\gamma \to 0^+$, the eigenvalues of $\mathcal{A}(\gamma)$ tend to three distinct values:*

$$\lim_{\gamma\to 0^+} \quad \lambda(\mathcal{A}(\gamma)) \;=\; \begin{cases} 1 \\ \frac{1}{2}(1+\sqrt{3}\,i) \\ \frac{1}{2}(1-\sqrt{3}\,i) \end{cases} . \tag{5.6}$$

Proof. First, in order to find $\mathcal{A}(\gamma)$, we must determine $\mathcal{P}^{-1}$. It can be verified that

$$\mathcal{P}^{-1} = \begin{pmatrix} A_p^{-1} & A_p^{-1}\tilde{B}^\top S^{-1} \\ 0 & -S^{-1} \end{pmatrix} = \begin{pmatrix} \frac{1}{\gamma}I & 0 & \frac{1}{\gamma}B_1^\top S^{-1} \\ 0 & A_{22}^{-1} & A_{22}^{-1}B_2^\top S^{-1} \\ 0 & 0 & -S^{-1} \end{pmatrix} . \tag{5.7}$$

It follows that

$$\mathcal{A}(\gamma) = \mathcal{H}\mathcal{P}^{-1} = \begin{pmatrix} \tilde{A}A_p^{-1} & (\tilde{A}A_p^{-1} - I)\tilde{B}^\top S^{-1} \\ \tilde{B}A_p^{-1} & I \end{pmatrix} = \begin{pmatrix} 0 & 0 & -B_1^\top S^{-1} \\ 0 & I & 0 \\ \frac{1}{\gamma}B_1 & B_2A_{22}^{-1} & I \end{pmatrix} . \tag{5.8}$$

Now, to determine the eigenvalues of the preconditioned system $\mathcal{A}(\gamma)$, we need to find solutions (λ, x) to the homogeneous system

$$(\mathcal{A}(\gamma) - \lambda I)x = \begin{pmatrix} -\lambda I & 0 & -B_1^\top S^{-1} \\ 0 & (1-\lambda)I & 0 \\ \frac{1}{\gamma}B_1 & B_2A_{22}^{-1} & (1-\lambda)I \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = 0\,, \tag{5.9}$$

with $x \neq 0$. It is easy to see that $\det(\mathcal{A}(\gamma) - \lambda I) = 0$ when $\lambda = 1$. Therefore, $\lambda = 1$ is an eigenvalue of the preconditioned system $\mathcal{A}(\gamma)$, independent of γ. It is clear that the algebraic multiplicity of $\lambda = 1$ as an eigenvalue of $\mathcal{A}(\gamma)$ is at least $n - n_z$ (the size of the $(2,2)$ block in $\mathcal{A}(\gamma) - \lambda I$), regardless of γ. Let us now seek the eigenvalues of $\mathcal{A}(\gamma)$ that are not equal to 1. Assuming $\lambda \neq 1$, the second equation of (5.9) implies that $x_2 = 0$. Therefore, we seek λ, x_1 and x_3 satisfying

$$-\lambda x_1 - B_1^\top S^{-1} x_3 = 0\,, \tag{5.10a}$$

$$\frac{1}{\gamma} B_1 x_1 + (1-\lambda) x_3 = 0\,. \tag{5.10b}$$

Solving for x_1 in (5.10a) and substituting the result into (5.10b), we obtain

$$(\lambda^2 I - \lambda I + \frac{1}{\gamma} B_1 B_1^\top S^{-1}) x_3 = 0\,. \tag{5.11}$$

Now, to ensure that the eigenvector x is nonzero, observe that we must have $x_3 \neq 0$. This is because $x_3 = 0$ implies that $x_1 = 0$ from equation (5.10a), and we saw that $x_2 = 0$ if $\lambda \neq 1$. Next, normalize x_3 so that $x_3^* x_3 = 1$, and multiply equation (5.11) by x_3^* on the left to obtain

$$\lambda^2 - \lambda + \frac{1}{\gamma} x_3^* B_1 B_1^\top S^{-1} x_3 = 0\,. \tag{5.12}$$

We now substitute $S = \tilde{B} A_p^{-1} \tilde{B}^\top = \frac{1}{\gamma} B_1 B_1^\top + B_2 A_{22}^{-1} B_2^\top$ into (5.12) and rearrange γ to obtain the equivalent formulation

$$\lambda^2 - \lambda + x_3^*\, B_1 B_1^\top (B_1 B_1^\top + \gamma B_2 A_{22}^{-1} B_2^\top)^{-1}\, x_3 = 0\,. \tag{5.13}$$

From equation (5.13), the eigenvalue λ can be expressed as

$$\lambda = \frac{1}{2}\left(1 \pm \sqrt{1 - 4(x_3^* B_1 B_1^\top (B_1 B_1^\top + \gamma B_2 A_{22}^{-1} B_2^\top)^{-1} x_3)}\right) \tag{5.14}$$

Now, in order to evaluate the expression under the square root as γ approaches 0, we use Lemma 5.1 with $J = B_1 B_1^\top$ and $K = B_2 A_{22}^{-1} B_2^\top$ (both are symmetric positive semidefinite) to conclude that

$$P = \lim_{\gamma \to 0^+} B_1 B_1^\top (B_1 B_1^\top + \gamma B_2 A_{22}^{-1} B_2^\top)^{-1}$$

is a projector onto $\mathcal{R}(B_1 B_1^\top) = \mathcal{R}(B_1)$, where $\mathcal{R}$ denotes the range. Next, use equation (5.10b), rewritten as

$$x_3 = \left(\frac{1}{\gamma(\lambda - 1)}\right) B_1 x_1\,,$$

to observe that $x_3 \in \mathcal{R}(B_1)$. As a result,

$$\begin{aligned} &\lim_{\gamma \to 0^+} x_3^* B_1 B_1^\top (B_1 B_1^\top + \gamma B_2 A_{22}^{-1} B_2^\top)^{-1} x_3 \\ &\qquad = x_3^* P_{\mathcal{R}(B_1)} x_3 \\ &\qquad = x_3^* x_3 \\ &\qquad = 1\,. \end{aligned}$$

Taking the limit as $\gamma \to 0^+$ of both sides of equation (5.14), we can see that the eigenvalues of $\mathcal{A}(\gamma)$ that are not equal to 1 can be expressed as:

$$\begin{aligned}\lim_{\gamma\to 0^+} \lambda &= \frac{1}{2}\left(1 \pm \sqrt{1 - \lim_{\gamma\to 0^+} 4(x_3^* B_1 B_1^\top (B_1 B_1^\top + \gamma B_2 A_{22}^{-1} B_2^\top)^{-1} x_3)}\right) \\ &= \frac{1}{2}\left(1 \pm \sqrt{1-4}\right) \\ &= \frac{1}{2}\left(1 \pm \sqrt{3}\, i\right) .\end{aligned}$$

This concludes the proof of the theorem. □□

The foregoing theorem indicates that chosing a small positive value of γ will result in a preconditioned matrix with eigenvalues clustered around the three values 1, $\frac{1}{2}\left(1+\sqrt{3}\, i\right)$, and $\frac{1}{2}\left(1-\sqrt{3}\, i\right)$, so that in practice one can expect a rapid convergence of preconditioned GMRES. The actual choice of γ will be discussed in section 6 below.

It is worth mentioning a variation in the above choice of preconditioner for the Hessian $\mathcal{H}$: consider the preconditioner $\mathcal{P}_+$, defined by

$$\mathcal{P}_+ = \begin{pmatrix} A_p & \tilde{B}^\top \\ 0 & S \end{pmatrix} , \tag{5.15}$$

where $S = \tilde{B} A_p^{-1} \tilde{B}^\top$ is the Schur complement of the perturbed Hessian. Observe that $\mathcal{P}_+$ only differs from $\mathcal{P}$ (defined in (5.5)) in the sign in front of S. Hence, we can use the same reasoning as that of the proof of the theorem to analyze the spectrum of $\mathcal{H}\mathcal{P}_+^{-1}$ for $\gamma \to 0^+$. In particular, we compute

$$\mathcal{A}_+(\gamma) = \mathcal{H}\mathcal{P}_+^{-1} = \begin{pmatrix} 0 & 0 & B_1^\top S^{-1} \\ 0 & I & 0 \\ \frac{1}{\gamma} B_1 & B_2 A_{22}^{-1} & I \end{pmatrix} \tag{5.16}$$

and we can easily see that 1 is an eigenvalue of $\mathcal{H}\mathcal{P}_+^{-1}$. Using the eigenvalue-eigenvector equation to find the eigenvalues not equal to 1, we obtain

$$\lambda = \frac{1}{2}\left(1 \pm \sqrt{1 + 4(x_3^* B_1 B_1^\top (B_1 B_1^\top + \gamma B_2 A_{22}^{-1} B_2^\top)^{-1} x_3)}\right) .$$

Applying Lemma 5.1 we find

$$\begin{aligned}\lim_{\gamma\to 0^+} \lambda &= \frac{1}{2}\left(1 \pm \sqrt{1 + \lim_{\gamma\to 0^+} 4(x_3^* B_1 B_1^\top (B_1 B_1^\top + \gamma B_2 A_{22}^{-1} B_2^\top)^{-1} x_3)}\right) \\ &= \frac{1}{2}\left(1 \pm \sqrt{1+4}\right) \\ &= \frac{1}{2}\left(1 \pm \sqrt{5}\right) .\end{aligned}$$

In particular, as $\gamma \to 0^+$, the eigenvalues of $\mathcal{H}\mathcal{P}_+^{-1}$ tend to three nonzero values, all of which are real. This choice of $\mathcal{P}_+$ may also be used in the solution of the saddle point problem.

5.2. Applying the preconditioner. The easily verified identity

$$\mathcal{P}^{-1} = \begin{pmatrix} A_p^{-1} & O \\ O & I_m \end{pmatrix} \begin{pmatrix} I_n & \hat{B}^\top \\ O & I_m \end{pmatrix} \begin{pmatrix} I_n & O \\ O & -S^{-1} \end{pmatrix} \tag{5.17}$$

shows that the action of the preconditioner on a given vector requires one application of A_p^{-1}, one of S^{-1}, and one sparse matrix-vector product with $\hat{B}^\top$. Since A_p is diagonal, the first task is trivial and the critical (and potentially very expensive) step is the application of S^{-1}. We propose to perform this step inexactly using some inner iterative scheme. As it is well known, if these linear systems are solved inexactly by, say, a preconditioned conjugate gradient method (PCG), then the corresponding inexact variant of the block triangular (right) preconditioner $\mathcal{P}$ must be used within a flexible variant of a Krylov method, such as FGMRES [34].

In our analysis of the spectrum of the preconditioned matrix in the previous section we have assumed that we were able to obtain an exact inverse of the Schur complement S of the perturbed Hessian. If the exact inverse of S is replaced by an approximate inverse, the eigenvalues of the preconditioned matrix will form clusters around the eigenvalues of the exactly preconditioned matrix, $\mathcal{H}\mathcal{P}^{-1}$. The more accurate the solution of linear systems involving S, the smaller the cluster diameter can be expected to be; in turn, the convergence of the preconditioned FGMRES iteration is expected to improve as the accuracy of the solution of linear systems involving S is increased. The numerical experiments presented in the next section will confirm this intuition.

Another practical issue not yet addressed is the choice of the perturbation constant γ. Theorem 5.2 suggests to choose γ as small as possible so as to obtain tighter clusters of eigenvalues and, one hopes, lower FGMRES iteration counts for convergence. However, as $\gamma \to 0^+$, the perturbed Hessian Schur complement S becomes increasingly ill-conditioned, and the application of the preconditioner $\mathcal{P}$ will require more computational work. Therefore we must strike a balance in the choice of the perturbation constant γ so that it is "small enough" to make FGMRES converge quickly and "large enough" to make the (approximate) Schur complement solve require minimal computational effort.

Recall that $S = \frac{1}{\gamma} B_1 B_1^\top + B_2 A_{22}^{-1} B_2^\top$. Each of the two components $\frac{1}{\gamma} B_1 B_1^\top$ and $B_2 A_{22}^{-1} B_2^\top$ of S is symmetric positive semidefinite (singular) while their sum is symmetric positive definite (nonsingular). Moving from the principle that the solver should treat each component of S "equally", we choose the perturbation constant

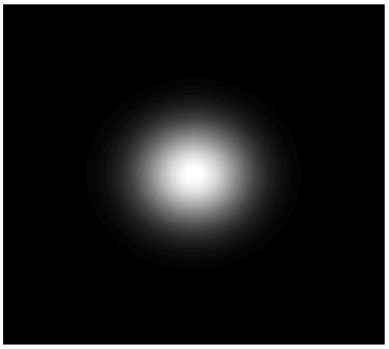
$$\gamma = \frac{1}{\text{mean}(\text{diag}(B_2 A_{22}^{-1} B_2^\top))} \,. \tag{5.18}$$

Here, "mean" is used to denote the arithmetic mean or average of the entries of a vector. This choice of γ attempts to effectively balance the tasks of clustering the eigenvalues of $\mathcal{H}\mathcal{P}^{-1}$ and keeping the Schur complement from becoming too ill-conditioned.

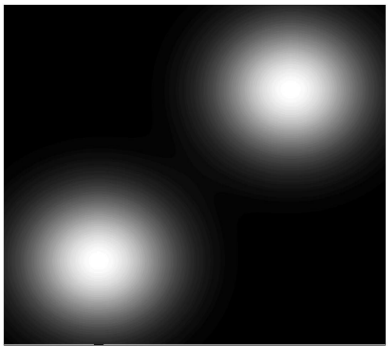
Once we have set γ, we can apply $\mathcal{P}^{-1}$ to a vector by (approximately) solving a system of linear equations involving the Schur complement, $Sx = b$. To achieve this, we use the conjugate gradient (CG) method preconditioned with an off-the-shelf algebraic multigrid solver. Specifically, we use the distributed memory algebraic preconditioning package ML developed by Hu et al. as part of the Trilinos Project at Sandia National Laboratories [23], with the default choice of parameters. Note that here S is a sparse matrix which can be formed explicitly. We choose the ML

preconditioner because it is a popular solver for unstructured sparse linear systems; it is of course possible that better results may be obtained using a customized solver that exploits *a priori* knowledge about the properties and origin of the Schur complement matrix S.

6. Numerical results. We begin by recalling that in our version of the Monge–Kantorovich mass transfer problem we can think of the initial and final densities, u_0 and u_T, as images that we wish to register to one another. Figure 6.1 displays the images that correspond to our initial and final densities, u_0 and u_T, respectively. Note that, in the discrete hyperbolic problem formulation (4.12), we obtain the vector b from u_T, and we obtain the vector q from u_0.



(a) u_0

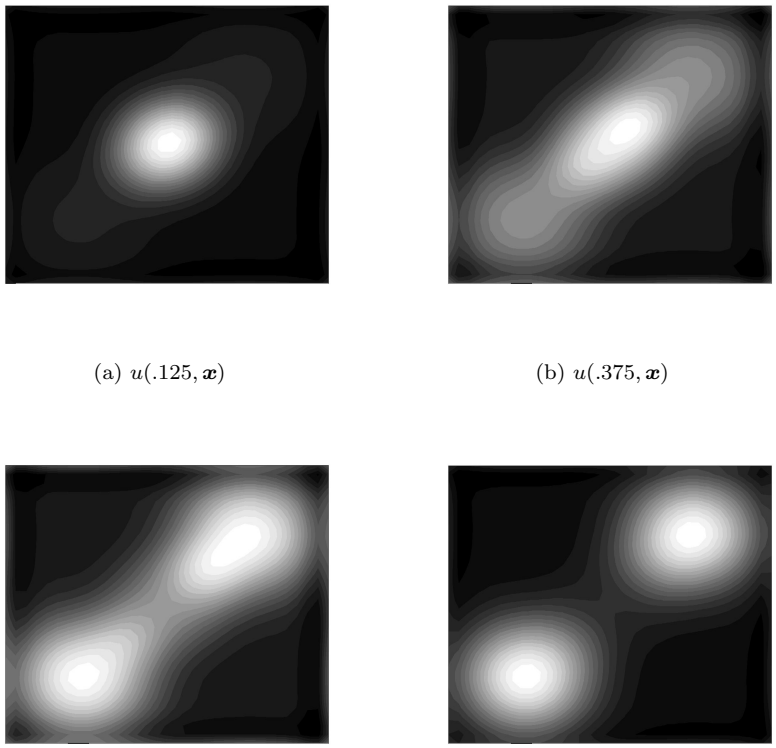


(b) u_T

Fig. 6.1. *Images corresponding to initial density, u_0, and final density, u_T.*

To visualize the solution of the hyperbolic model problem, Figure 6.2 displays the solution $u(t, \boldsymbol{x})$ for different values of t in the time interval $[0, 1]$. Recall that,

given the initial and final image of Figure 6.1, we are trying to determine an optimal mapping (or *morphing*) between the two images. The contour plots of the solution $u(t, \boldsymbol{x})$ at different times t help us visualize this optimal mapping.



(a) $u(.125, \boldsymbol{x})$ (b) $u(.375, \boldsymbol{x})$

(c) $u(.625, \boldsymbol{x})$ (d) $u(.875, \boldsymbol{x})$

FIG. 6.2. *Plots of the solution $u(t, \boldsymbol{x})$ for different values of t.*

In order to choose the regularization parameter α in (4.3), recall that α must be large enough to recover a smooth parameter function $m(\boldsymbol{x})$ and small enough to give significant weight to the data fitting term in (4.12). Following [5], we determine an optimal α on a coarse grid, and use this α for finer grids. For the hyperbolic model problem, we set $\alpha = 10$.

We apply the inexact Gauss–Newton method to solve the optimization problem, using FGMRES to solve the linear systems (4.15) arising at each Gauss–Newton step with inexact applications of the preconditioner $\mathcal{P}$ carried out in the manner explained in the previous section.

We stop the inexact Newton algorithm when the relative norm of the residuals in the Euler–Lagrange equations (3.9) falls below 10^{-4}. The tolerance for FGMRES is also set to 10^{-4}. Finally, the linear systems involving S (resulting from the application of the right preconditioner $\mathcal{P}^{-1}$) are solved with varying PCG convergence tolerance, in order to assess the effect of inexact solves and to find the "optimal" level of accuracy

Grid Size/ # Unknowns	PCG Tolerance	Newton Iterations	FGMRES Iterations	Ave. PCG Iterations	Total PCG Iterations
$8^2 \times 8$/ 12288	10^{-1}	7	22.7	1.1	180
	10^{-2}	7	22.4	2.1	328
	10^{-3}	7	21.1	3.3	492
	10^{-6}	7	16.4	9.8	1127
$16^2 \times 16$/ 98304	10^{-1}	6	30.2	2.0	371
	10^{-2}	6	28.3	3.2	544
	10^{-3}	6	28.2	6.0	993
	10^{-6}	6	21.5	18.2	2313
$32^2 \times 32$/ 786432	10^{-1}	5	37.0	3.3	604
	10^{-2}	5	36.4	4.6	842
	10^{-3}	5	34.2	11.2	1914
	10^{-6}	5	26.8	35.3	4585

Table 6.1

Results from solving the hyperbolic PDE-constrained optimization model problem with the right preconditioner $\mathcal{P}$ applied inexactly. Newton and FGMRES tolerance $= 10^{-4}$, $\alpha = 10$.

needed in terms of total work.

Table 6.1 displays, for three different grids on $\Omega \times [0,1] \equiv [0,1]^3$, the results obtained with the inexact Gauss–Newton method to solve the hyperbolic problem using the application of the preconditioner $\mathcal{P}$ in FGMRES. The column "FGMRES Iterations" displays the average number of FGMRES iterations per Newton (outer) iteration, the column "Ave. PCG Iterations" displays the average number of PCG iterations per FGMRES iteration, and the column "Total PCG Iterations" displays the total number of PCG iterations over all FGMRES and Newton iterations.

First of all we observe that the inexact Gauss–Newton iteration converges very rapidly to the solution of the discrete optimization problem, with rates independent of problem size. We further note that the PCG tolerance can be quite high (leading to a rather inexact Schur complement inverse, S^{-1}) without significantly affecting the convergence rate of FGMRES. Hence, choosing the PCG stopping tolerance of 10^{-1} results in relatively low FGMRES iteration counts (only mildly depending on the grid size) and therefore in the least amount of total work, as seen in the "Total PCG Iterations" column. We can safely conclude that it is preferrable to solve the Schur complement systems to rather low relative accuracy in the application of the preconditioner $\mathcal{P}$.

The results displayed in Table 6.1, while encouraging, are sub-optimal in one regard, namely, in terms of scaling of computational effort with respect to problem size. From the last column we can see that halving the space-time discretization parameter results in approximately twice as many PCG iterations. Clearly, the algebraic multigrid preconditioner ML is not scalable for this problem, which can be attributed to the fact that the Schur complement matrix S resemble a discretization of an elliptic PDE with strongly varying coefficients. The question of developing more efficient solvers for the approximate Schur complement problem is left for future work.

Additional tests were perfomed using the $\mathcal{P}_+$ variant of the block triangular preconditioner, with very similar results to those obtained with $\mathcal{P}$. We also experimented with a different solution scheme based on a reduced Hessian approach, which is widely used in optimization and particularly in PDE-constrained optimization; see,

e.g., [26, 20]. Unfortunately, this approach turned out to be much more expensive and even less scalable than the one based on block triangular preconditioning. We omit the details and refer interested readers to the third author's PhD thesis; see [22, Chapter 3.5].

7. Conclusions. We have considered the solution of a PDE-constrained optimization problem where the constraint is a hyperbolic PDE. This problem arises for instance in image registration and is closely related to the classical Monge–Kantorovich optimal transport problem. Formally, the problem fits within a parameter estimation framework for which extensive work on numerical solution algorithms has been performed in recent years. In this paper we have investigated the use of a block triangular preconditioner $\mathcal{P}$ for the saddle point system that arises in each inexact Gauss–Newton iteration applied to a discretization of the PDE-constrained optimization problem. Theoretical analysis of the preconditioned system indicates that the use of $\mathcal{P}$ can be expected to result in rapid convergence of a Krylov subspace iteration like GMRES, with convergence rates independent of discretization and other parameters. In practice, however, exact application of the preconditioner is too expensive due to the need to solve a linear system involving the Schur complement of the perturbed Hessian. Instead, we propose to solve this linear system inexactly using a PCG iteration. Numerical experiments indicate that solving these linear systems to a low relative accuracy is sufficient to maintain the rapid convergence of the preconditioned Krylov subspace iteration applied to the saddle point problem, with convergence rates only mildly dependent on problem parameters. Additional accuracy does not significantly improve the convergence rates and it increases significantly the overall costs. In our experiments, we used the ML smoothed aggregation-based AMG from Trilinos as the preconditioner for the inner CG iteration. While not optimal in terms of scalability, the resulting inexact block triangular preconditioner outperforms a reduced Hessian-based approach and appears to be promising. Future work should be aimed at improving the scalability of the inner PCG method used for the approximate Schur complement solves.

REFERENCES

[1] V. Akcelik, G. Biros, and O. Ghattas, *Parallel multiscale Gauss–Newton–Krylov methods for inverse wave propagation*, Proceedings of the IEEE/ACM Conference (2002), pp. 1–15.

[2] V. Akcelik, G. Biros, O. Ghattas, et. al., *High resolution forward and inverse earthquake modeling on terascale computers*, Proceedings of the IEEE/ACM Conference (2003), pp. 1–52.

[3] S. Angenent, S. Haker, and A. Tannenbaum, *Minimizing flows for the Monge–Kantorovich problem*, SIAM J. Math. Anal., 35 (2003), pp. 61–97.

[4] S. R. Arridge *Optical tomography in medical imaging*, Inverse Problems, 15 (1999), pp. R41–R93.

[5] U. M. Ascher and E. Haber, *Grid refinement and scaling for distributed parameter estimation problems*, Inverse Problems, 17 (2001), pp. 571–590.

[6] J. D. Benamou and Y. Brenier *A computational fluid mechanics solution to the Monge-Kantorovich mass transfer problem*, Numer. Math., 84 (2000), pp. 375–393.

[7] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica, 14 (2005), pp. 1–137.

[8] L. Biegler, O. Ghattas, M. Heinkenschloss, and B. Waanders, *Large-Scale PDE-Constrained Optimization*, Lecture Notes in Computational Science and Engineering, Vol. 30, Springer-Verlag, New York, 2003.

[9] G. Biros and O. Ghattas, *Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Parts I-II*, SIAM J. Sci. Comput., 27 (2005), pp. 687–738.

[10] I. BOUCHOUEV AND V. ISAKOV, *Uniqueness, stability and numerical methods for the inverse problem that arises in financial markets*, Inverse Problems, 15 (1999), pp. R95–R116.
[11] R. CASANOVA, A. SILVA, AND A. R. BORGES, *A quantitative algorithm for parameter estimation in magnetic induction tomography*, Meas. Sci. Technol., 15 (2004), pp. 1412–1419.
[12] M. CHENEY, D. ISAACSON, AND J. C. NEWELL, *Electrical impedance tomography*, SIAM Review, 41 (1999), pp. 85–101.
[13] P. DEUFLHARD AND F. POTRA, *Asymptotic mesh independence of Newton-Galerkin methods via a refined Mysovskii theorem*, SIAM J. Numer. Anal, 29 (1992), pp. 1395–1412.
[14] H. S. DOLLAR, *Properties of Linear Systems in PDE-Constrained Optimization. Part I: Distributed Control*, Tech. Rep. RAL-TR-2009-017, Rutherford Appleton Laboratory, 2009.
[15] B. DUPIRE, *Pricing with a smile*, Risk, 7 (1994), pp. 32–39.
[16] H. EGGER AND H. W. ENGL, *Tikhonov regularization applied to the inverse problem of option pricing: convergence analysis and rates*, Inverse Problems, 21 (2005), pp. 1027–1045.
[17] H. ELMAN, D. SILVESTER, AND A. WATHEN, *Finite Elements and Fast Iterative Solvers with Applications in Incompressible Fluid Dynamics*, Oxford University Press, Oxford, 2005.
[18] G. EL-QADY AND K. USHIJIMA, *Inversion of DC resistivity data using neural networks*, Geophys. Prosp., 49 (2001), pp. 417–430.
[19] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations, Third Edition*, John Hopkins University Press, 1996.
[20] E. HABER AND U. M. ASCHER, *Preconditioned all-at-once methods for large, sparse parameter estimation problems*, Inverse Problems, 17 (2001), pp. 1847–1864.
[21] E. HABER AND J. MODERSITZKI, *A multilevel method for image registration*, SIAM J. Sci. Comput., 27 (2006), pp. 1594–1607.
[22] L. R. HANSON, *Techniques in Constrained Optimization Involving Partial Differential Equations*, PhD thesis, Emory University, Atlanta, GA, 2007.
[23] J. HU, M. SALA, C. TONG, R. TUMINARO ET. AL., *ML: Multilevel Preconditioning Package*, The Trilinos Project, Sandia National Laboratories, 2006. <`http://trilinos.sandia.gov/packages/ml/`>.
[24] L. V. KANTOROVICH, *On the translocation of masses*, Dokl. Akad. Nauk SSSR, 37 (1942), pp. 227–229 (in Russian). English translation in J. Math. Sciences, 133 (2006), p. 1381–1382.
[25] L. V. KANTOROVICH, *On a problem of Monge*, Uspekhi Mat. Nauk, 3 (1948), pp. 225–226 (in Russian). English translation in J. Math. Sciences, 133 (2006), p. 1383.
[26] 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.
[27] M. V. KLIBANOV AND T. R. LUCAS, *Numerical solution of a parabolic inverse problem in optical tomography using experimental data*, SIAM J. Appl. Math., 59 (1999), pp. 1763–1789.
[28] C. D. LAIRD, L. T. BIEGLER, B. WAANDERS, AND R. A. BARTLETT, *Time-dependent contaminant source determination for municipal water networks using large scale optimization*, ASCE J. Water Res. Mgt. Plan., 131 (2005), pp. 125–134.
[29] R. J. LEVEQUE, *Numerical Methods for Conservation Laws*, Birkhauser, New York, 1990.
[30] J. NOCEDAL AND S. J. WRIGHT, *Numerical Optimization*, Springer, New York, 1999.
[31] C. OROZCO AND O. GHATTAS, *Massively parallel aerodynamic shape optimization*, Comp. Syst. Eng., 1 (1992), pp. 311–320.
[32] R. L. PARKER, *Geophysical Inverse Theory*, Princeton University Press, Princeton, NJ, 1994.
[33] T. REES, H. S. DOLLAR, AND A. J. WATHEN, *Optimal Solvers for PDE-Constrained Optimization*, Tech. Rep. RAL-TR-2008-018, Rutherford Appleton Laboratory, 2008.
[34] Y. SAAD, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Sci. Comput., 14 (1993), pp. 451–469.
[35] Y. SAAD AND M. H. SCHULTZ, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comput., 7 (1986), pp. 856–869.
[36] A. SHENOY, M. HEINKENSCHLOSS, AND E. M. CLIFF, *Airfoil design by an all-at-once method*, Int. J. Comput. Fluid Dyn., 11 (1998), pp. 3–25.
[37] C. R. VOGEL, *Sparse matrix computations arising in distributed parameter identification*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1027–1037.
[38] C. R. VOGEL, *Computational Methods for Inverse Problems*, Frontiers in Applied Mathematics Series, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2002.