Technical Report

TR-2007-009

Model problems in PDE-constrained optimization

by

E. Haber, L. Hanson

Mathematics and Computer Science

EMORY UNIVERSITY

Technical Report

TR-200X-00X-A

Model problems in PDE-constrained optimization

by

E. Haber and L. Hanson

Mathematics and Computer Science

EMORY UNIVERSITY

Model Problems in PDE-Constrained Optimization

Eldad Haber Lauren Hanson *

May 7, 2007

Abstract

This work aims to aid in introducing, experimenting and benchmarking algorithms for PDE-constrained optimization problems by presenting a set of such model problems. We specifically examine a type of PDE-constrained optimization problem, the parameter estimation problem. We present three model parameter estimation problems, each containing a different type of partial differential equation as the constraint. We also shortly describe discretization and solution techniques for each problem. We further supply a simple to modify matlab code with the article.

1 Introduction

PDE-constrained optimization problems are optimization problems with partial differential equations as constraints. Partial differential equations mathematically represent a multitude of natural phenomena, and in turn, applications in science and engineering ubiquitously give rise to problems formulated as PDE-constrained optimization problems. For instance, PDE-constrained optimization problems arise in such diverse areas as aerodynamics ([48, 50]), mathematical finance ([16, 21, 22]), medicine ([4, 40]), and environmental engineering ([1, 2, 41]). PDE-constrained optimization problems are generally infinite dimensional in nature, large and complex. As a result, this class of optimization problems present significant computational challenges, many of which have been studied in recent years in such works as [6, 13, 29, 53, 7, 11, 14, 19, 31, 32, 33, 35, 36, 37, 38, 39, 51, 56], just to name a few.

As computing power grows and optimization techniques become more advanced, one wonders whether there are enough commonalities among PDE-constrained optimization problems from different fields to develop algorithms for more than a single application. This question has been the topic of many papers, workshops and recent grants (see, for example, [12, 43, 44, 45]). It is, without a doubt, a difficult question to answer. At first, developing generic solution techniques for PDE-constrained optimization seems unrealistic; as John

*Department of Mathematics and Computer Science, Emory University, Atlanta, Georgia 30322, USA. haber, lrhanso@mathcs.emory.edu.

Dennis once said, *"if we could solve all PDE-constrained optimization problems with a single toolbox, then we could solve all PDEs by optimizing a constant subject to the PDE".*[1] While it may be the case that different PDEs give rise to different challenges, there are several classification approaches that might aid in broadening the scope of solution algorithms. For example, PDEs could be classified as hyperbolic, parabolic, or elliptic, or they could be classified as linear or nonlinear. If we find a useful classification from an optimization point of view, then we may be able to develop more generic algorithms and software. The goal of this paper is to assist in the development of such generic algorithms by classifying some PDE-constrained optimization problems. We propose a few relatively simple model problems (at least from a PDE standpoint) that are involved enough to explore some of the major difficulties in the field. To allow for other practitioners to test their ideas, we also supply Matlab code for the solution of each problem, as well as the Jacobians needed for optimization.

When attempting to solve a PDE-constrained optimization problem, one might first ask an important question: should I *discretize the optimization problem first and solve a discrete optimization problem*, or should I *optimize the continuous problem first and obtain a set of equations to discretize?* The first approach is often referred to as "Discretize-Optimize" (DO) and the second approach is known as "Optimize-Discretize" (OD). An important challenge in optimization is that, in general, these two steps does not commute. As a result, the two different approaches could lead to two different solutions. The advantages and disadvantages of both problems are nicely summarized by Gunzburger in [26]:

- Optimize-Discretize (OD): With this approach, one can obtain inconsistent gradients of the (discrete) objective functionals. In other words, unless the grid is fine enough, the approximate gradient obtained with OD is not a true gradient of anything: not of the continuous functional nor the discrete functional.

- Discretize-Optimize (DO): This approach requires the differentiation of computational facilitators such as turbulence models, shock capturing devices or outflow boundary treatment.

These two approaches to solve a PDE-constrained optimization problem lead to a simple example of a classification of such problems. The first group contains problems that can be trivially discretized first, implying a DO approach. In other words, the first group contains problems whose discretization does not lead to non-differentiable components (such as flux limiters). In contrast, while the second group of problems are continuously differentiable before discretizing, we obtain non-differentiable components after discretizing such problems. Our intention in this paper is to work with discrete optimization algorithms for smooth functions. Therefore, we only consider problems in which the DO approach can be taken.

The model problems chosen for this paper are parameter estimation problems, one of the most important and common types of PDE-constrained optimization problems. A large amount of interest on parameter estimation problems has been generated in recent years;

[1] personal communication

see, for example, [6, 29, 52, 53] and references therein. A *parameter estimation problem* is an inverse problem that seeks to recover one or more unknown coefficients in a partial differential equation using some *a priori* knowledge of the solution of that equation.

This paper is organized as follows: section 2 describes the general formulation of the parameter estimation problem. We discuss possible difficulties in the solution of the parameter estimation problem, while reviewing reduced Hessian-based techniques, in section 3. Section 4 presents the model problems. Finally, results of numerical experimentation with our Matlab code are presented in section 5.

2 General Formulation

We consider the problem of recovering an approximation for a model (parameter function) based on measurements of solutions of a system of partial differential equations (PDEs). In other words, we are interested in the *inverse problem* of recovering an approximation for a model, $u(\mathbf{x})$, based on measurement data $\mathbf{d}$ on the solution $y(\mathbf{x})$ of the *forward problem*. In general, the forward problem can be nonlinear with respect to y. In this paper, we will consider an important class of problems that share two common features:

- We assume that the forward problem is linear with respect to y and the PDE can be written as
$$\mathcal{A}(u)y = q, \tag{2.1}$$
where $\mathcal{A}$ is a differential operator depending on the model $u(\mathbf{x})$, the problem is defined on an appropriate domain Ω, and the problem is equipped with suitable boundary conditions. For simplicity, we will assume that there is a unique solution y for any fixed choice of u and q.

- As explained in the introduction, we assume that the discretization of the problem is "straightforward" and that no "exotic" features such as flux limiters are needed. In this case, the discrete forward problem is continuously differentiable with respect to both y and u.

Although our assumptions may look highly restrictive, problems that satisfy the above assumptions constitute a large variety of applications such as magnetotelluric inversion, DC resistivity, hydrology and diffraction tomography; see [18, 20, 23, 49, 53] and references therein.

Given the forward problem for y we define an operator Q to be the projection of y onto the locations in Ω to which the data $\mathbf{d}$ are associated. Thus, we can interpret the data as a nonlinear function of the model u:

$$\mathbf{d} = Q\mathcal{A}(u)^{-1}q + \epsilon \tag{2.2}$$

Here, ϵ is the measurement noise. Because the data are finite and noisy, and the operator $\mathcal{A}^{-1}$ is typically compact, the inverse problem of recovering u 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 for example [54]).

Suppose we employ the regularization method of minimizing the Tikhonov functional to find u. More precisely, the inverse problem to approximate u becomes a minimization problem of the form

$$\min_u \quad \frac{1}{2}\|Q\mathcal{A}(u)^{-1}q - \mathbf{d}\|^2 + \alpha R(u - u_r) \tag{2.3}$$

where u_r is a reference model and $\alpha \geq 0$ is the regularization parameter. For this paper we will employ a common choice for the regularization functional R:

$$R(u - u_r) = \frac{1}{2}\int_\Omega (\beta(u - u_r)^2 + |\nabla(u - u_r)|^2) \tag{2.4}$$

where β is a constant. Many other choices for R can be made; in fact, the choice of R is a research topic on its own, however we choose to concentrate on other optimization issues.

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

$$\begin{aligned} \min_{u,y} \quad & \frac{1}{2}\|Qy - \mathbf{d}\|^2 + \alpha R(u - u_r) \\ \text{s.t} \quad & \mathcal{A}(u)y - q = 0 \end{aligned} \tag{2.5}$$

The optimization problem (2.5) is an equality constrained optimization problem. In many applications simple bound constraints on u are added however, here, we ignore those constraints for the sake of simplicity.

3 Discretization and Optimization

Suppose we discretize the PDE in (2.1) using a technique such as finite differences or finite elements, and we obtain

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

where A is a nonsingular matrix, y is the grid function approximating $y(\mathbf{x})$ and arranged as a vector, and u and q likewise relate to $u(\mathbf{x})$ and $q(\mathbf{x})$. Discretize the regularization functional (2.4) similarly, so that

$$R(u - u_r) = \frac{1}{2}\|L(u - u_r)\|^2,$$

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

$$\min_{y,u} \quad \frac{1}{2}\|Qy - d\|^2 + \frac{1}{2}\alpha\|L(u - u_r)\|^2 \tag{3.7a}$$

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

There exist several approaches for the solution of the discrete optimization problem. Starting with the discrete problem (3.7), consider an inexact Newton-type method to find a solution. We can vary the solver for the KKT system arising from each Newton iteration, but we will apply a preconditioned Krylov method directly to the KKT system. Note that this method is also referred to as the 'all-at-once' approach ([29, 34, 50]), as the solutions of the forward problem and the inverse problem are computed simultaneously.

Let us briefly review the inexact Newton-type method to solve the constrained formulation (3.7). Introduce the Lagrangian

$$\mathcal{L}(y,u,p) = \frac{1}{2}\|Qy-d\|^2 + \frac{1}{2}\alpha\|L(u-u_r)\|^2 + p^\top V(A(u)y-q) \tag{3.8}$$

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

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

It is interesting to note that classical 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 Lagrange multiplier $p(\mathbf{x})$, the matrix V is necessary.

Now, a necessary condition for an optimal solution of our problem is

$$\begin{aligned}
\mathcal{L}_y &= Q^\top(Qy-d) + A(u)^\top V p = 0, && \text{(3.9a)}\\
\mathcal{L}_u &= \alpha L^\top L(u-u_r) + G(y,u)^\top V p = 0, && \text{(3.9b)}\\
\mathcal{L}_p &= V(A(u)y-q) = 0, && \text{(3.9c)}
\end{aligned}$$

where $G(y,u) = \partial(A(u)y)/\partial u$. To compute $G(y,u)$ it is possible to use automatic differentiation tools (such as [10]). However, we show that if the forward problem is discretized in a simple way, then it is usually easier to compute $G(y,u)$ directly.

Although it is possible to use the Full Approximation Scheme (FAS) to solve the nonlinear system in the context of multigrid methods (see [15]) we concentrate here on a Newton's type approach. In our opinion, the approach has a few advantages. First, it is more modular, i.e. it does not necessarily require a multilevel solver. Second, while there is no current theory for global convergence of FAS methods for optimization problems there is such theory for constrained optimization. Furthermore, recent advances in the field allow for inaccurate solutions of the KKT system [17].

A Newton linearization for solving the nonlinear equations (3.9) leads to a KKT system at each iteration, so that a Newton step requires the solution of the system

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

where $*$ represents second order derivatives. Although a Newton method can be used, many practical codes use the Gauss-Newton method instead. In the Gauss-Newton approximation, one disregards second order terms in the KKT system, so that the system to solve becomes

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

This approach usually results in convergence that is only linear, but it guarantees a symmetric positive definite *reduced Hessian*. Since it is often difficult to check the inertia of the KKT system, such approximations can be beneficial in practice.

The problem of solving the KKT system (3.11) is an active area of research, with a wide variety of solvers proposed (see [6, 5, 9, 13, 29], or [53], just to name a few). One approach to solve the KKT system, the *reduced Hessian method*. The approach can be interpreted as a block elimination process, eliminating δy first and then δp, one obtains an equation for δu. The approach requires solving a linear system

$$H_{\text{red}} \delta u = -g_{\text{red}} \tag{3.12}$$

where

$$H_{\text{red}} = J^\top J + \alpha L^\top L$$

is the reduced Hessian in which $J = -QA^{-1}G$ and g_{red} is the reduced gradient. Although the reduced Hessian system is smaller than the KKT system, it is still large in distributed parameter estimation applications. In addition, the reduced Hessian is dense in most applications, and should not be formed or stored. However, depending on the problem, employing the reduced Hessian method can be an efficient way to solve the PDE-constrained optimization problem; preconditioning the reduced Hessian system is an open area of research [54].

The reduced Hessian method, when using inexact solves in one or more of the steps, yields an *approximate inverse* to the KKT matrix. If it is possible to find an inexpensive approximate inverse to the KKT matrix using the reduced Hessian method, we could consider using this approximate inverse as a preconditioner for a solver of system (3.11). For instance, if we run a small number of iterations of a stationary method such as multigrid, Gauss-Seidel or Jacobi when solving the systems involving the forward problem, we could use the (approximate) reduced Hessian as a preconditioner for a Krylov subspace method to solve the KKT system.

One attractive approach to obtain an approximation to the inverse of the reduced Hessian H_{red} is to use some quasi-Newton method [27, 13]. In particular, one can use the limited memory Broyden-Fletcher-Goldberg-Shanno (LBFGS) update to approximate the inverse of the reduced Hessian. Such an approach uses a sequence of reduced gradients to build an approximation to the reduced Hessian inverse. It is also possible to use an non-stationary iterative solver for the forward problem in the course of preconditioning. Such solver has the advantage of using existing codes for the forward and adjoint problems for the preconditioning of the KKT system. However, if such code is to be used then one has to choose a flexible

method for the KKT solver. The results in our numerical experiments are obtained by using a flexible GMRES method. We believe that flexible methods has been under-utilized for such problems and that they can offer better robustness.

The above description for the solution of the discrete problem is not complete, as it gives rise to some important open questions. First, classical SQP algorithms require an accurate solution of the linear system (3.11), which is usually very expensive for large scale problems. It is natural to wonder what the lowest accuracy might be in the solution of the KKT system to guarantee *global* convergence. Recently, a framework for inexact SQP method has been developed [17]. Our preliminary experienece with their approach is highly encoureging. Second, developing optimal solvers for the KKT system is an open problem. For some particular problems, multigrid methods have been applied successfully ([15, 43, 5]), while for others, multigrid methods can be difficult to apply.

Now that we have described a general formulation and some approaches to solve a general parameter estimation problem, we are ready to present three parameter estimation test problems.

4 Model Problems

In this section we describe three model problems. We present a problem whose underlying PDE is elliptic, one whose underlying PDE is parabolic, and one whose underlying PDE is hyperbolic. For each of the problems, we describe the discretization, present the Jacobians, and identify other optimization-related issues. For ease of presentation and programming, we choose to concentrate on the finite difference discretization, but finite element or spectral methods can easily be applied as well.

4.1 An Elliptic Model Problem

4.1.1 Formulation

The following model problem arises in groundwater modeling and DC resistivity applications ([23, 53]). In this case, the forward problem (2.1) takes the form

$$-\nabla \cdot (e^u \nabla y_i) = q_i \ , \ \mathbf{x} \in \Omega \tag{4.13a}$$

$$\nabla y_i \cdot n = 0 \ , \ \mathbf{x} \in \partial\Omega \tag{4.13b}$$

$$\int_\Omega y_i \, d\,\Omega = 0 \quad i = 1, ..., n_s \tag{4.13c}$$

where y_i is some potential field, u is known as the log conductivity, and $\Omega \subset \mathbb{R}^3$. Note that the choice of the log conductivity is applied to handle the practical difficulty often arising when the conductivity varies widely over a few orders of magnitude. Condition (4.13b) implies that the flux vanishes in the normal direction to the boundary, n. The first two equations in (4.13) specify y_i up to a constant; we add equation (4.13c) to ensure uniqueness of the solution y_i. The number of PDE's in applications can be rather large, which makes the

problem more computationally complex but yields overall better recovery of u. For simplicity of presentation in this section we use $n_s = 1$. In our numerical experiments we use $n_s = 6$.

4.1.2 Discretization

We now briefly describe finite difference discretization of (4.13). To aid in the development of algorithms, we also point out a few of the important challenges from an optimization point of view.

Assume that the domain Ω is the unit cube and consider discretization on a uniform grid to form a union of N^3 cubic cells of side h each. We define the components of the vector y (corresponding to the continuous function $y(\mathbf{x})$) in the cell centers. Note that the vector y has N^3 elements. Next, let D be the short, central difference matrix (which incorporates Neumann boundary conditions) in one dimension for a grid function of N points. The gradient of y can be approximated by the following tensor products:

$$\nabla_h := \begin{pmatrix} I \otimes I \otimes D \\ I \otimes D \otimes I \\ D \otimes I \otimes I \end{pmatrix}$$

where I is the $N \times N$ identity matrix. In a similar way, we can approximate the divergence by

$$[\nabla \cdot]_h := -\nabla_h^\top.$$

Next, place the components of the vector u (corresponding to the parameter function $u(\mathbf{x})$) in cell centers, as in y, so that the vector u has N^3 elements. Since we need e^u at cell faces (each component of e^u should lie in the same spatial location as the corresponding component of $\nabla_h\, y$), averaging is needed. It is well known that harmonic averaging of e^u is required in order to obtain accurate simulations (see [6] and references therein). Let A_c^f be an averaging matrix from cell centers to faces. The quantity e^u on cell faces can be written as

$$(\mathrm{e}^u)_{\text{faces}} = (A_c^f \mathrm{e}^{-u})^{-1}$$

where $(\cdot)^{-1}$ denotes the inversion of each component of the vector.

It is interesting to pause here and to look at the difference between the discrete and the continuous models. In the continuous model, the problem is bilinear with respect to $\sigma = \exp(u)$ and y. However, in the discrete setting, the problem is nonlinear with respect to σ due to the harmonic average. This is a simple demonstration of the difference between the OD approach and the DO approach. it is interesting to observe that the Euler-Lagrange equations of the discrete optimization problem are quite different than the discretization of the Euler-Lagrange equations of the continuous problem.

Using the gradient, divergence, and averaging matrices, we obtain a discretization of the forward problem:

$$A(u)\, y = \nabla_h^\top \operatorname{diag}\left((A_c^f \mathrm{e}^{-u})^{-1}\right)\, \nabla_h\, y = q \tag{4.14}$$

Here, q is a vector that corresponds to the function $q(\mathbf{x})$, with components that lie in the same spatial locations as the components of the grid function y.

The matrix $A(u)$, as defined in (4.14), is singular, since y is only specified up to a constant in this representation. We therefore must incorporate condition (4.13c) before continuing. In discrete terms, we will add in the condition that $e^\top y = 0$, where e is a vector of ones, to form a discretization of (4.13), the nonsingular system

$$\tilde{A}(u)\, y = (A(u) + h^3 e e^\top) y = q\,. \tag{4.15}$$

Note that $\tilde{A}(u)$ is dense, however, we only use matrix-vector products in iterative methods. As a result, the matrix $\tilde{A}(u)$ need not be formed explicitly.

4.1.3 Jacobians

To carry out gradient-based optimization, we need to compute the Jacobians of the forward problem. The Jacobian with respect to y is trivial, but the Jacobian with respect to u is more complicated. One can verify that, given y,

$$\begin{aligned} G(y,u) &= \frac{\partial(\tilde{A}(u)y)}{\partial u} \\ &= \nabla_h^\top \operatorname{diag}(\nabla_h\, y) \operatorname{diag}\left((A_c^f \mathrm{e}^{-u})^{-2}\right) A_c^f \operatorname{diag}\left(\mathrm{e}^{-u}\right)\,. \end{aligned}$$

It is interesting to note that, in this case, computing the Jacobians directly is not only possible, but it is a much better alternative to using Automatic Differentiation (AD) tools. In particular, observe that we can take advantage of the particular structure of the matrix $\tilde{A}(u)$ (the fact that its density is due to a rank one perturbation) when we use direct differentiation. However, AD tools simply return a dense matrix when differentiating $\tilde{A}(u)y$ with respect to y.

4.1.4 Data and regularization

Assume that the data is measured in discrete points in the domain. As a result, define the linear interpolation matrix Q so that it interpolates the potential field y linearly onto the measurement points.

To discretize the regularization operator, we use the previous discretization of the gradient, ∇_h, to obtain the matrix L:

$$L = h^{\frac{3}{2}} \begin{pmatrix} \nabla_h \\ \beta^{\frac{1}{2}} I \end{pmatrix} \tag{4.16}$$

where I is the $N^3 \times N^3$ identity matrix. As a result, the discretized regularization term is

$$R_h(u) = \|Lu\|^2 = h^3 \left(\|\nabla_h u\|^2 + \beta \|u\|^2\right)\,. \tag{4.17}$$

4.2 Parabolic Model Problem

4.2.1 Formulation

The forward problem in our next distributed parameter estimation problem is parabolic and time-dependent. The following problem is motivated by similar parameter estimation

problems that arise in applications such as optical tomography ([4, 40]) and electromagnetic imaging ([18, 20, 28]). Here, we consider a very simplified model and state the forward problem (2.1) as:

$$y_t - \nabla \cdot (e^u \nabla y) = 0 \ , \ \mathbf{x} \in \Omega \ , \ t \in [0, T] \tag{4.18a}$$

$$y(\mathbf{x}, 0) = y_0(\mathbf{x}) \ , \ \mathbf{x} \in \Omega \tag{4.18b}$$

$$\nabla y \cdot n = 0 \ , \ \mathbf{x} \in \partial\Omega \ , \ t \in [0, T] \tag{4.18c}$$

where $\Omega \subset \mathbb{R}^3$.

In the context of optical tomography in medical imaging, $y = y(\mathbf{x}, t)$ is the photon density and $u = u(\mathbf{x})$ is the diffusion coefficient. Also, in optical tomography, there is an additional term in (4.18a) that contains information about the absorption cross section at position $\mathbf{x}$. Note that the complexity of this forward problem is higher than that of the groundwater flow distributed parameter estimation problem due to the the addition of time. We wish to recover the parameter function u based on data taken on the solution y of the forward problem (4.18).

4.2.2 Discretization

To discretize the time dependent problem (4.18), we first fix a time t_k and discretize the PDE in space using the same discretization discussed above for the elliptic problem. In particular, we obtain the following discretization:

$$\nabla \cdot (e^u \nabla y(\mathbf{x}, t_k)) \approx A(u) y^k \ , \text{ where} \tag{4.19a}$$

$$A(u) \ = -\nabla_h^\top \operatorname{diag}\left((A_c^f \mathrm{e}^{-u})^{-1}\right) \nabla_h \ . \tag{4.19b}$$

Here, y^k is the grid function corresponding to $y(\mathbf{x}, t_k)$, and ∇_h and A_c^f are the matrices described in the discretization of the elliptic model problem. This representation yields a large and stiff ordinary differential equation (ODE) of the form

$$y_t = A(u) y \ . \tag{4.20}$$

Since the problem is stiff, we use the Backward Euler method to discretize the problem in time. More accurate methods can be used; such methods lead to a slight increase of complexity of presentation.

Assume that time ranges from 0 to 1 and consider a uniform discretization of the time interval with $(N_t + 1)$ grid points, so that each time step has length $h_t = 1/N_t$. This yields the system

$$\frac{1}{h_t}(y^k - y^{k-1}) - A(u) y^k = 0 \quad \text{for } k = 1, 2, \ldots, N_t \ . \tag{4.21}$$

To put all model problems on equal footing, it is useful to write equation (4.21) as a

single linear system for the vector $y = [y_1, \dots y_{N_t}]$:

$$\tilde{A}(u)y = \begin{pmatrix} B(u) & & & & \\ -I & B(u) & & & \\ & \cdot & \cdot & & \\ & & \cdot & \cdot & \\ & & & -I & B(u) \end{pmatrix} \begin{pmatrix} y^1 \\ y^2 \\ \cdot \\ \cdot \\ y^{N_t} \end{pmatrix} = \begin{pmatrix} y^0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{pmatrix} = q \tag{4.22}$$

where $B = B(u) = I - h_t A(u)$. Clearly, the forward problem system can be very large, even with a coarse grid discretization. Nevertheless, the forward system need not be formed or stored, and only matrix-vector products are needed for the solution of the problem.

4.2.3 Jacobians, data, and regularization

As in the elliptic model problem, a trivial computation leads to the Jacobian with respect to y, but we must perform some additional calculations to find the Jacobian with respect to u. It can be verified that, given y,

$$\begin{aligned} G(y,u) &= \frac{\partial(\tilde{A}(u)y)}{\partial u} = -h_t \begin{pmatrix} G_1 \\ G_2 \\ \cdot \\ \cdot \\ G_{N_t} \end{pmatrix}, \text{ where} \\ G_k &= \frac{\partial(A(u)y^k)}{\partial u} \\ &= -\nabla_h^\top \operatorname{diag}\left(\nabla_h y^k\right) \operatorname{diag}\left((A_c^f e^{-u})^{-2}\right) A_c^f \operatorname{diag}\left(e^{-u}\right) . \end{aligned}$$

Since it is possible to compute the Jacobians with direct differentiation, and this technique exploits the structure of the matrices, we prefer direct differentiation over AD tools. Similarly to the elliptic model problem, assume that the data is measured in discrete points in the domain, and define the matrix Q so that it interpolates y linearly onto the measurement points.

It only remains to express the discretization of the regularization operator. As we use the regularization functional (2.4), we can use the discretization of the regularization operator given in (4.16).

4.3 Hyperbolic Model Problem

Our third model problem corresponds to a hyperbolic forward problem with smooth initial data. A similar problem arises in the Monge-Kantorovich (MKP) mass transfer problem with applications of mass transfer to statistics, fluid mechanics, and image processing; see [3, 8], and references therein.

4.3.1 Formulation

Let $\mathbf{x} \in \Omega \subset \mathbb{R}^2$, and consider two given bounded density functions $y_0(\mathbf{x}) \geq 0$ and $y_T(\mathbf{x}) \geq 0$. Assume that

$$\int_\Omega y_0(\mathbf{x})\, d\mathbf{x} \approx \int_\Omega y_T(\mathbf{x})\, d\mathbf{x}. \tag{4.23}$$

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 $\phi(\mathbf{x}) : \mathbb{R}^2 \to \mathbb{R}^2$ to be the minimizer of the $\mathcal{L}^2$ Kantorovich distance between y_0 and y_T. In particular, we wish to find

$$\min_\phi \quad \int_\Omega |\phi(\mathbf{x}) - \mathbf{x}|^2 y_0(\mathbf{x})\, d\mathbf{x} \tag{4.24}$$

among all maps ϕ that transport y_0 to y_T.

Several approaches have been proposed to solve the above optimization problem. To set the problem in a parameter estimation framework, consider reformulating the problem to obtain a fluid mechanics problem, as Benamou and Brenier proposed in [8]. They proved that finding the solution to (4.24) is equivalent to the following optimization problem. Introduce a time interval $[0, T]$, and consider seeking a smooth, time-dependent density field $y(t, \mathbf{x})$ and a smooth, time-dependent velocity field $u(t, \mathbf{x}) = (u_1(t, \mathbf{x}), u_2(t, \mathbf{x}))$ that satisfy

$$\min_{y,u} \quad \frac{1}{2} \|y(T, \mathbf{x}) - y_T(\mathbf{x})\|^2 \; + \; \frac{1}{2}\alpha T \int_\Omega \int_0^T \; y||u||^2 dt\, d\mathbf{x} \tag{4.25a}$$

$$\text{st.} \quad y_t + \nabla \cdot (yu) = 0\,, \tag{4.25b}$$

$$y(0, \mathbf{x}) = y_0\,. \tag{4.25c}$$

Equation (4.25) is a continuous PDE-constrained optimization problem, where the PDE is a hyperbolic equation. The next section will describe the finite difference discretization of the components of (4.25), highlighting some of the unique challenges that arise when solving this problem.

4.3.2 Discretization

For simplicity, let us limit the problem to one in which the initial and final densities are smooth. In this case, simpler standard discretization techniques can be used. Observe that this example once again demonstrates the difference between the DO and the OD approach. If the initial or final densities are non-smooth, one cannot use simple discretization techniques, and methods such as upwinding or flux limiters are required (see [42]). Note that even the lowest form of upwind involves terms such as $|y|$; as a result, although the PDE is linear with respect to y, its discrete analog is not. Furthermore, the discrete analog in this case is not differentiable.

Since the velocity field u is unknown *a priori*, it is difficult to choose appropriate time steps to ensure stability of the scheme for explicit discretization. We therefore choose an implicit Lax-Friedrichs scheme to discretize (4.25b) in order to avoid the limitations of CFL conditions.

We discretize the time interval $[0, T]$ using N_t time steps, each with width $h_t = T/N_t$. Next, assume that $\Omega = [0, 1] \times [0, 1]$, and discretize the spatial domain Ω with N_x grid points in each direction, so that the side of each cell has length $h_x = 1/N_x$. Next, for each time step t_k, form the vectors y^k, u_1^k, and u_2^k, corresponding to $y(t_k, \mathbf{x})$, $u_1(t_k, \mathbf{x})$, and $u_2(t_k, \mathbf{x})$, respectively, where the elements are cell-centered. Using $y_{i,j}^k$ to denote the element in the vector corresponding to $y(t_k, \mathbf{x}_{i,j})$ (and a similar notation for u_1 and u_2), we can write the finite difference approximations for the implicit Lax-Friedrichs scheme as follows:

$$\left(\frac{\partial y}{\partial t}\right)_{i,j}^k \approx \frac{1}{h_t}\left[y_{i,j}^{k+1} - \frac{1}{4}(y_{i+1,j}^k + y_{i-1,j}^k + y_{i,j+1}^k + y_{i,j-1}^k)\right] , \tag{4.26a}$$

$$(\nabla \cdot (yu))_{i,j}^k \approx \tag{4.26b}$$

$$\frac{1}{2h_x}\left[(y \odot u_1)_{i+1,j}^{k+1} - (y \odot u_1)_{i-1,j}^{k+1} + (y \odot u_2)_{i,j+1}^{k+1} - (y \odot u_2)_{i,j-1}^{k+1}\right] .$$

Here, we use the symbol $\odot$ to denote 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:

$$\frac{1}{h_t}[y^{k+1} - My^k] + B(u^{k+1})y^{k+1} = 0, \tag{4.27}$$

where M corresponds to an averaging matrix and $B(u)$ is the matrix which contains difference matrices in each direction. After reordering (4.27), the system to solve is:

$$C^{k+1}y^{k+1} = My^k \tag{4.28}$$

for $k = 0, 1, ..., N_t - 1$ where $C^{k+1} = C(u^{k+1}) = I + h_t B(u^{k+1})$. Therefore, the forward problem (4.25b) can be written in discrete form as

$$A(u)y = \begin{pmatrix} C(u^1) & & & & \\ -M & C(u^2) & & & \\ & \ddots & \ddots & & \\ & & \ddots & \ddots & \\ & & & -M & C(u^{N_t}) \end{pmatrix} \begin{pmatrix} y^1 \\ y^2 \\ \vdots \\ \vdots \\ y^{N_t} \end{pmatrix} = \begin{pmatrix} My^0 \\ 0 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} = q . \tag{4.29}$$

where y^0 is the vector obtained after discretizing the given density function y_0 consistently. The discretization of the forward problem (4.25b) is now complete.

4.3.3 Jacobians

To compute the Jacobian with respect to u, first examine the structure of the difference matrix $B(u^k)$:

$$B(u^k) = \begin{pmatrix} D_1 & D_2 \end{pmatrix} \begin{pmatrix} \mathrm{diag}(u_1^k) \\ \mathrm{diag}(u_2^k) \end{pmatrix} \tag{4.30}$$

where D_1 and D_2 are central difference matrices in each direction. As a result, we can compute the Jacobian of $A(u)y$ with respect to u:

$$G(y,u) \quad = \quad \tfrac{\partial (A(u)y)}{\partial u} \quad = \begin{pmatrix} G^1 & & & \\ & G^2 & & \\ & & . & \\ & & & G^{N_t} \end{pmatrix}, \text{ where}$$

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

The Jacobian with respect to y is trivial. Now that the components of the forward problem (4.25b) and its derivatives have been well-defined, we can discretize the remaining components of the problem.

4.3.4 Data and regularization

To represent the objective function (4.25a) in discrete form, first define some matrices and vectors. Let

$$u = \begin{pmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \\ . \\ . \\ u_1^{N_t} \\ u_2^{N_t} \end{pmatrix} , \tag{4.32a}$$

$$L = \begin{pmatrix} I & I & & & & & \\ & & I & I & & & \\ & & & & . & . & \\ & & & & & & I & I \end{pmatrix} , \quad \text{and} \quad Q = h_x \begin{pmatrix} 0 & . & . & 0 & I \end{pmatrix} , \tag{4.32b}$$

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 d be the vector obtained after discretizing the density function y_T consistently (taking scaling into account). Then it is easy to show that the discrete representation of (4.25a) is

$$\frac{1}{2}\|Qy - d\|^2 + \frac{1}{2}\alpha T h_t h_x^2 y^\top L \operatorname{diag}(u) u \, . \tag{4.33}$$

Combining the expressions (4.29) and (4.33), the discrete optimization problem becomes

$$\begin{aligned} \min \quad & \tfrac{1}{2}\|Qy - d\|^2 + \tfrac{1}{2}\gamma y^\top L \operatorname{diag}(u) u && \text{(4.34a)} \\ \text{st.} \quad & A(u)y - q = 0 && \text{(4.34b)} \end{aligned}$$

Here, $\gamma = \alpha T h_t h_x^2$. Observe the variation in the regularization term from (3.7) to (4.34); for this reason, let us present the Euler-Lagrange equations and the KKT system associated with (4.34).

First, a necessary condition for an optimal solution of our problem is expressed in the Euler-Lagrange equations,

$$\begin{aligned} \mathcal{L}_y &= Q^\top(Qy - d) + \frac{1}{2}\gamma L \mathrm{diag}(u)u + A(u)^\top V p = 0, && (4.35a) \\ \mathcal{L}_u &= \gamma \mathrm{diag}(L^\top y)u + G(y,u)^\top V p = 0, && (4.35b) \\ \mathcal{L}_p &= V(A(u)y - q) = 0, && (4.35c) \end{aligned}$$

where p is a vector of Lagrange multipliers, and G is defined in section 4.3.3. Next, using a Gauss-Newton approximation, the KKT system is computed as

$$Hs = \begin{pmatrix} Q^\top Q & 0 & A^\top V \\ 0 & \gamma \mathrm{diag}(L^\top y) & G^\top V \\ VA & VG & 0 \end{pmatrix} \begin{pmatrix} \delta y \\ \delta u \\ \delta p \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_y \\ \mathcal{L}_u \\ \mathcal{L}_p \end{pmatrix} . \quad (4.36)$$

5 Numerical Experiments

The following experiments were performed on the three model problems using Matlab code, which is available online at (`www.mathcs.emory.edu/ haber`). To solve the optimization problem (3.7), we run a Newton iteration to find the zeros of the Euler-Lagrange equations, approximating the inverse of the reduced Hessian with a limited memory BFGS update. In particular, when solving (3.11) (or (4.36)) at each outer iteration, we use flexible GMRES preconditioned with the reduced Hessian method (as described in Section 3), applying limited memory BFGS to approximate $H_{red}^{-1} g_{red}$ in (3.12).

Whenever we require a forward problem solution (solving linear systems involving $A(u)$), we apply a Krylov method with a fixed stopping tolerance. Note that the application of the Krylov method, assuming it yields an inexact solution, leads to a nonstationary preconditioner for the KKT system. This is permissible as long as we use flexible GMRES to solve the KKT system. Also, note that it is possible to view the all-at-once method and the reduced Hessian method as identical methods except in the exactness of the forward problem solves. In other words, an exact forward and adjoint problem solves leads to the reduced Hessian method, and an inexact forward and adjoint problem solve leads to the all-at-once method.

An important issue when designing any algorithm for PDE-Optimization problem is the measure of work. It is possible to use cpu time or "flops" but such measures can be difficult to compare, especially if one compares performance on different computational hardware. Here we suggest yet another way to compare the different algorithms. Note that the main building block in all the above algorithms is the product of the forward problem Jacobians with a vector. Thus, counting the forward matrix-vector products gives a good indication of the amount of work needed to accomplish the task. Profiling our code, we found that such products consist of roughly 80% of the computational time. Below, we give some sample

results from our codes. We hope that other practitioners in the field could use these as a starting point in order to improve our algorithms and obtain substantially faster methods.

For each model problem, we present some output in which we vary the FGMRES stopping tolerance for the solution of (3.11) and the stopping tolerance for the conjugate gradient method for the forward problem solution. For PDE-optimization problems it is also interesting to see how the problem scales when different grids are used. To this extent we use 3 different grids for the solution of each problem.

To initialize the unknowns in each problem we use the reference model for u and then approximately solved the forward problem to a tolerance of 10^{-2}. We notice that when the forward problem is solved tightly in order to initialize the solution more outer iterations are needed for convergence. For all problems we chose regularization parameters that give a rough misfit of the noise level.

The results are presented in Table 1. For each problem we record the grid size, total number of unknowns (y, u, p) the tolerance of the FGMRES solver, the tolerance of the forward solver within the GMRES solver, the number of outer (nonlinear) iterations. The total number of forward problem vector products, and the total number of KKT matrix-vector products.

There are a few interesting observations

- The average number of FGMRES per iteration is more or less mesh independent. In some cases it is difficult to compare between levels and the FGMRES can behave in an unpredictable manor.

- Since the "work-horse" of the problem is the solution of the forward problem, the problem does not scale linearly with the mesh size. This can be obviously improved if an optimal method such as multigrid is used for the forward problem.

- Similar to many other problems, tightening the tolerance of the linear solver does decrease the number of (nonlinear) iteration but increase the overall work.

A more interesting observation that deserves further studying is that in some cases, using a lower tolerance for the forward problem 10^{-4} in the FGMRES (i.e. getting a better preconditioner) with a "relaxed" stopping criteria for the FGMRES 10^{-2} lead to a substantial increase in the number of *outer* iterations. To try and explain this phenomena, we observed the directions and residuals obtained at the end of each solution of the KKT system. We observe that for small tolerance of the forward solve, the residuals for the (linearized) forward and adjoint equations are very small while the residual for the model (or control) is rather large. Since the stopping criteria for the FGMRES is based on the 2-norm of the residual, this implies that the direction taken in u may not be a good direction overall compared with the directions for y and λ. We suspect that this cause the algorithm to stall.

6 Conclusions

In this paper lay-out simple model problems for PDE optimization. We described a family of algorithms based on the reduced Hessian method and its approximation. We supply the code and perform experiments with the code. It is important to note that the code we supply is not optimal. For example, the solution of the forward problem is performed by using simple iterative methods with simple preconditioners. Much better performance can be obtained by using optimal preconditioners for the forward problem. Second, the implementation of the solution to the KKT system is not optimal. We hope that better algorithms can be developed and implemented to dramatically accelerate the solution of this type of problems.

Our hope is that by releasing a publicly available, easy to modify matlab code, we will encourage practitioners in linear algebra and optimization to further improve upon current technology in PDE optimization solvers.

Acknowledgments

We thank Todd Munson of the Mathematics and Computer Science Division of Argonne National Laboratory for his support throughout this work.

References

[1] 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.

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

[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, *A multigrid method for distributed parameter estimation problems*, Electron. Trans. Numer. Anal., 15 (2003), pp. 1-17.

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

[7] R. A. Bartlett, M. Heinkenschloss, D. Ridzal, and B. Waanders, *Domain decomposition methods for advection dominated linear-quadratic elliptic optimal control problems*, Comp. Methods in Applied Mech. Eng., 195 (2006), pp. 6428-6447.

[8] 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.

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

[10] M. Berz et. al., *A collection of automatic differentiation tools*, Argonne National Laboratory, 2006. <http://www-unix.mcs.anl.gov/autodiff/AD_Tools/>.

[11] L. Biegler, O. Ghattas, M. Heinkenschloss, D. Keyes, and B. Waanders, *Real-time PDE-constrained optimization*, Comp. Sci. Eng., 3 (2007).

[12] L. Biegler, O. Ghattas, M. Heinkenschloss, and B. Waanders, *Large-scale PDE-constrained optimization*, Lecture Notes in Computational Science and Engineering, Springer-Verlag, New York, 2003.

[13] 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.

[14] A. Borzi, *High-order discretization and multigrid solution of elliptic nonlinear constrained optimal control problems*, J. Comp. Applied Math., 200 (2007), pp. 67-85.

[15] A. Borzi and K. Kunisch, *A multigrid scheme for elliptic constrained optimal control problems*, Comp. Optim. Applic., 31 (2005), pp. 309-333.

[16] 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.

[17] R. H. Byrd, F. Curtis, and J. Nocedal, *Inexact SQP methods for equality constrained optimization*, SIAM J. Opt., Submitted for publication.

[18] 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.

[19] T. F. Chan and X. C. Tai, *Identification of discontinuous coefficients from elliptic problems using total variation regularization*, SIAM J. Sci. Comput., 25 (2003), pp. 881-904.

[20] M. Cheney, D. Isaacson, and J. C. Newell, *Electrical impedance tomography*, SIAM Review, 41 (1999), pp. 85-101.

[21] B. Dupire, *Pricing with a smile*, Risk, 7 (1994), pp. 32-39.

[22] 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.

[23] G. El-Qady and K. Ushijima, *Inversion of DC resistivity data using neural networks*, Geophys. Prosp., 49 (2001), pp. 417-430.

[24] H. Elman, D. Silvester, and A. Wathen, *Finite Elements and Fast Iterative Solvers*, Oxford University Press, Oxford, 2005.

[25] R. Fletcher, S. Leyffer, and P. Toint, *A brief history of filter methods*, Preprint ANL/MCS-P1371-0906 (2006), Argonne National Laboratory.

[26] M. Gunzburger, *Perspectives in Flow Control and Optimization*, SIAM, 1987.

[27] E. Haber, *Quasi-Newton methods for large scale electromagnetic inverse problem*, Inverse Problems (21), 2004, pp. 305-317.

[28] E. Haber, *A parallel method for large scale time domain electromagnetic inverse problems*, IMACS J. Appl. Num. Math., to appear.

[29] E. Haber and U. M. Ascher, *Preconditioned all-at-once methods for large, sparse parameter estimation problems*, Inverse Problems, 17 (2001), pp. 1847-1864.

[30] E. Haber, U. M. Ascher, and D. Oldenburg, *On optimization techniques for solving nonlinear inverse problems*, Inverse Problems, 16 (2000), pp. 1263-1280.

[31] E. Haber and J. Modersitzki, *A multilevel method for image registration*, SIAM J. Sci. Comput., 27 (2006), pp. 1594-1607.

[32] S. B. Hazra and V. Schulz, *Simultaneous pseudo-timestepping for aerodynamic shape optimization problems with state constraints*, SIAM J. Sci. Comput., 28 (2006), pp. 1078-1099.

[33] M. Heinkenschloss and M. Herty, *A spatial domain decomposition method for parabolic optimal control problems*, J. Comp. Applied Math., 201 (2007), pp. 88-111.

[34] M. Heinkenschloss and L. N. Vicente, *Analysis of inexact trust region SQP algorithms*, Technical Report TR 99-18 (1999), Rice University.

[35] M. Hintermuller and L. N. Vicente, *Space mapping for optimal control of partial differential equations*, SIAM J. Opt., 15 (2005), pp. 1002-1025.

[36] M. Hinze, *A variational discretization concept in control constrained optimization: the linear-quadratic case*, J. Comp. Opt. Applic., 30 (2005), pp. 45-63.

[37] M. Hinze and R. Pinnau, *A second order approach to optimal semiconductor design*, J. Opt. Theory and Applic., to appear.

[38] C. Hogea, C. Davatzikos, and G. Biros, *An image-driven parameter estimation problem for a reaction-diffusion glioma growth model with mass effects*, Submitted for publication.

[39] A. Joshi, W. Bangerth, and E. M. Sevick-Muraca, *Non-contact fluorescence optical tomography with scanning patterned illumination*, Opt. Exp., 14 (2006), pp. 6516-6534.

[40] 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.

[41] 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.

[42] R. J. Leveque, *High-resolution conservative algorithms for advection in incompressible flow*, SIAM J. Numer. Anal., 33 (1996), pp. 627-665.

[43] R. M. Lewis and S. G. Nash, *Model problems for the multigrid optimization of systems governed by differential equations*, SIAM J. Sci. Comput., 26 (2005), pp. 1811-1837.

[44] H. Maurer and H. D. Mittleman, *Optimization techniques for solving elliptic control problems with control and state constraints. Part 1: boundary control*, Comp. Optim. Applic., 16 (2000), pp. 29-55.

[45] H. Maurer and H. D. Mittleman, *Optimization techniques for solving elliptic control problems with control and state constraints. Part 2: distributed control*, Comp. Optim. Applic., 18 (2001), pp. 141-160.

[46] S. G. Nash, *A survey of truncated-Newton methods*, J. Comp. Applied Math., 124 (2000), pp. 45-59.

[47] J. Nocedal and S. J. Wright, *Numerical Optimization*, Springer, New York, 1999.

[48] C. Orozco and O. Ghattas, *Massively parallel aerodynamic shape optimization*, Comp. Syst. Eng., 1 (1992), pp. 311-320.

[49] R. L. Parker, *Geophysical Inverse Theory*, Princeton University Press, Princeton, NJ, 1994.

[50] 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.

[51] M. Ulbrich and S. Ulbrich, *Primal-dual interior-point methods for pde-constrained optimization*, Technical Report (2006), TU Munchen.

[52] K. van den Doel and U. M. Ascher, *On level set regularization for highly ill-posed distributed parameter estimation problems*, J. Comp. Phys., 216 (2006), pp. 707-723.

[53] C. R. Vogel, *Sparse matrix computations arising in distributed parameter identification*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1027-1037.

[54] C. R. Vogel, *Computational Methods for Inverse Problems*, SIAM Frontiers in Applied Mathematics Series, 2002.

[55] S. J. Wright, *Primal-Dual Interior Point Methods*, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1997.

[56] V. M. Zavala, C. D. Laird, and L. T. Biegler, *Interior-point decomposition approaches for parallel solution of large-scale nonlinear parameter estimation problems*, Chem. Eng. Sci., Submitted for publication.

grid	# unknowns	KKT tol	Forward tol	outer iter	# for MV	# KKT MV
Elliptic						
8^3	6656	10^{-2}	10^{-2}	5	22245	82
		10^{-2}	10^{-4}	14	55919	109
		10^{-4}	10^{-2}	4	31446	127
		10^{-4}	10^{-4}	4	37063	90
16^3	53245	10^{-2}	10^{-2}	5	42611	83
		10^{-2}	10^{-4}	30	178246	161
		10^{-4}	10^{-2}	4	62051	132
		10^{-4}	10^{-4}	4	63550	76
32^3	425984	10^{-2}	10^{-2}	5	403057	516
		10^{-2}	10^{-4}	50	432570	161
		10^{-4}	10^{-2}	4	1198021	1465
		10^{-4}	10^{-4}	4	112667	79
Parabolic						
$8^3 \times 8$	8704	10^{-2}	10^{-2}	8	82489	256
		10^{-2}	10^{-4}	10	49882	85
		10^{-4}	10^{-2}	8	75534	245
		10^{-4}	10^{-4}	8	65828	127
$16^3 \times 16$	135168	10^{-2}	10^{-2}	8	119176	111
		10^{-2}	10^{-4}	20	209592	95
		10^{-4}	10^{-2}	6	120010	123
		10^{-4}	10^{-4}	7	166396	100
$32^3 \times 32$	2129920	10^{-2}	10^{-2}	7	373082	132
		10^{-2}	10^{-4}	17	419497	55
		10^{-4}	10^{-2}	6	388739	141
		10^{-4}	10^{-4}	6	440809	84
Hyperbolic						
$16^2 \times 8$	12288	10^{-2}	10^{-2}	7	7048	188
		10^{-2}	10^{-4}	7	9197	127
		10^{-4}	10^{-2}	5	10230	279
		10^{-4}	10^{-4}	5	12838	188
$32^2 \times 16$	98304	10^{-2}	10^{-2}	5	8329	135
		10^{-2}	10^{-4}	5	10650	82
		10^{-4}	10^{-2}	6	51301	954^*
		10^{-4}	10^{-4}	6	51051	438^*
$64^3 \times 32$	786432	10^{-2}	10^{-2}	5	28397	270
		10^{-2}	10^{-4}	5	32346	143
		10^{-4}	10^{-2}	4	37740	387
		10^{-4}	10^{-4}	4	42849	198

Table 1: Results for different model problems. The $*$ means that FGMRES did not converge for some of the outer iterations within 200 inner iterations.