A penalty method for PDE-constrained optimization in inverse problems

T. van Leeuwen[1] **and F.J. Herrmann**[2]

[1]Mathematical Institute, Utrecht University, Utrecht, the Netherlands.
[2] Dept. of Earth, Ocean and Atmospheric Sciences, University of British Columbia, Vancouver (BC), Canada.

E-mail: T.vanLeeuwen@uu.nl

Abstract. Many inverse and parameter estimation problems can be written as PDE-constrained optimization problems. The goal, then, is to infer the parameters, typically coefficients of the PDE, from partial measurements of the solutions of the PDE for several right-hand-sides. Such PDE-constrained problems can be solved by finding a stationary point of the Lagrangian, which entails simultaneously updating the parameters and the (adjoint) state variables. For large-scale problems, such an *all-at-once* approach is not feasible as it requires storing all the state variables. In this case one usually resorts to a *reduced* approach where the constraints are explicitly eliminated (at each iteration) by solving the PDEs. These two approaches, and variations thereof, are the main workhorses for solving PDE-constrained optimization problems arising from inverse problems. In this paper, we present an alternative method that aims to combine the advantages of both approaches. Our method is based on a quadratic penalty formulation of the constrained optimization problem. By eliminating the state variable, we develop an efficient algorithm that has roughly the same computational complexity as the conventional *reduced* approach while exploiting a larger search space. Numerical results show that this method indeed reduces some of the non-linearity of the problem and is less sensitive the initial iterate.

1. Introduction

In inverse problems, the goal is to infer physical parameters (e.g., density, soundspeed or conductivity) from indirect observations. When the underlying model is described by a partial differential equation (PDE) (e.g., the wave-equation or Maxwell's equations), the observed data are typically partial measurements of the solutions of the PDE for multiple right-hand-sides. The parameters typically appear as coefficients in the PDE. These problems arise in many applications such as geophysics [1, 2, 3, 4], medical imaging [5, 6] and non-destructive testing.

For linear PDEs, the inverse problem can be formulated (after discretization) as a constrained optimization problem of the form

$$\min_{\mathbf{m},\mathbf{u}} \tfrac{1}{2}||P\mathbf{u}-\mathbf{d}||_2^2 \quad \text{s.t.} \quad A(\mathbf{m})\mathbf{u}=\mathbf{q}, \tag{1}$$

where $\mathbf{m} \in \mathbb{R}^M$ represents the (gridded) parameter of interest, $A(\mathbf{m}) \in \mathbb{C}^{N\times N}$ and $\mathbf{q} \in \mathbb{C}^N$ represent the discretized PDE and source term, $\mathbf{u} \in \mathbb{C}^N$ is the state variable and $\mathbf{d} \in \mathbf{C}^L$ are the observed data. The measurement process is modelled by sampling the state with $P \in \mathbb{R}^{L\times N}$. Throughout the paper ^{T} denotes the (complex-conjugate) transpose.

Typically, measurements are made from multiple, say K, independent experiments, in which case $\mathbf{u} \in \mathbb{C}^{KN}$ is a block vector containing the state variables for all the experiments. Likewise, $\mathbf{q} \in \mathbb{C}^{KN}$ and $\mathbf{d} \in \mathbb{C}^{KL}$ are block vectors containing the right-hand-sides and observations for all experiments. The matrices $A(\mathbf{m})$ and P will be block-diagonal matrices in this case. Typical sizes for M, N, K, L for seismic inverse problems are listed in table 1.

In practice, one usually includes a regularization term in the formulation (1) to mitigate the ill-posedness of the problem. To simplify the discussion, however, we ignore such terms with the understanding that appropriate regularization terms can be added when required.

1.1. All-at-once and reduced methods

In applications arising from inverse problems, the constrained problem (1) is typically solved using the method of Lagrange multipliers [7, 8] or sequential quadratic programming (SQP) [9, 10]. This entails optimizing over both the parameters, states and the Lagrange multipliers (or adjoint-state variables) simultaneously. While such *all-at-once* approaches are often very attractive from an optimization point-of-view, they are typically not feasible for large-scale problems since we cannot afford to store the state variables for all K experiments simultaneously. Instead, the so-called *reduced* approach is based on a (block) elimination of the constraints to formulate an unconstrained optimization problem over the parameters:

$$\min_{\mathbf{m}} \tfrac{1}{2}||PA(\mathbf{m})^{-1}\mathbf{q}-\mathbf{d}||_2^2. \tag{2}$$

While this eliminates the need to store the full state variables for all K experiments, evaluation of the objective and its gradient requires PDE-solves. Moreover, by eliminating the constraints we have dramatically reduced the search-space, thus arguably making it more difficult to find an appropriate minimizer. Note also that the dependency of the objective on $\mathbf{m}$ is now through $A(\mathbf{m})^{-1}\mathbf{q}$ in stead of through $A(\mathbf{m})\mathbf{u}$. For linear PDEs, the latter can often be made to depend linearly on $\mathbf{m}$ while the dependency of $A(\mathbf{m})^{-1}\mathbf{q}$ is much more complicated.

1.2. Motivation

The main motivation for this work is the observation that the stated inverse problem would be much easier to solve if we had a *complete* measurement of the state (i.e., P is invertible). In this case, we could reconstruct the state from the data as $\mathbf{u} = P^{-1}\mathbf{d}$ and subsequently recover the parameter by solving

$$\min_{\mathbf{m}} \tfrac{1}{2}\|A(\mathbf{m})\mathbf{u} - \mathbf{q}\|_2^2, \tag{3}$$

which for many linear PDEs would lead to a linear least-squares problem. This approach is known in the literature as the *equation-error approach* [11, 12]. When we do not have complete measurements, this method does not apply directly since we cannot invert P. We can, however, aim to recover the state by solving the following (inconsistent) overdetermined system

$$\begin{pmatrix} A(\mathbf{m}) \\ P \end{pmatrix} \mathbf{u} \approx \begin{pmatrix} \mathbf{q} \\ \mathbf{d} \end{pmatrix},$$

which combines the physics and the data. We can subsequently use the obtained estimate of the state to estimate $\mathbf{m}$ by solving (3). These steps can be repeated in an alternating fashion as needed. In a previous paper [13], we proposed this methodology for seismic inversion and coined it Wavefield Reconstruction Inversion (WRI). We showed – via numerical experiments – that this approach can mitigate some of the (notorious) non-linearity of the seismic inverse problem. In the current paper we seek to analyse this approach in more detail and broaden the scope of its application to inverse problems involving PDEs.

1.3. Contributions and outline

We give the above sketched method a sound theoretical basis by showing that it can be derived from a *penalty* formulation of the constrained problem:

$$\min_{\mathbf{m},\mathbf{u}} \tfrac{1}{2}||P\mathbf{u} - \mathbf{d}||_2^2 + \frac{\lambda}{2}\|A(\mathbf{m})\mathbf{u} - \mathbf{q}\|_2^2, \tag{4}$$

the solution of which (theoretically) satisfies the optimality conditions of the constrained problem (1) as $\lambda \to \infty$. Such reformulations of the constrained problem are well-known but are, to our best knowledge, not being applied to large-scale inverse problems.

The main contribution of this paper is the development of an efficient algorithm based on the penalty formulation for large-scale inverse problems and the insight that we can approximate the solution of the constrained problem up to arbitrary finite precision with a finite λ. The numerical experiments suggest that a single fixed value of λ is typically sufficient.

Our approach is based on the elimination of the state variable, $\mathbf{u}$, from (4) via a *variational projection* approach as detailed in section (4). This reduces the dimensionality of the optimization problem in a similar fashion as the *reduced* approach (2) does for the constrained formulation (1) by solving K systems of equations. This elimination leads to a cost function $\phi_\lambda(\mathbf{m})$ whose gradient and Hessian can be readily computed. The main difference is that the state $\mathbf{u}$ in this case is not defined by solving the PDE, but instead is solved from an overdetermined system that involves both the PDE *and* the data. Due to the special block-structure of the problems under consideration, this elimination can be done efficiently, leading to a tractable algorithm

for large-scale problems. Contrary to the conventional *reduced* approach, the resulting algorihtm does *not* enforce the constraints at each iteration and arguably leads to a less non-linear problem in $\mathbf{m}$. It is outside the scope of the current paper to give a rigorous prove of this statement, but we present some numerical evidence to support this conjecture.

The outline of the paper is as follows. First, we give a brief overview of the constrained and penalty formulations in sections 2 and 3. The main theoretical results are presented in section 4 while a detailed description of the proposed algorihtm is given in section 5. Here, we also compare the penalty approach to both the all-at-once and the reduced approaches in terms of algorithmic complexity. Numerical examples on a 1D DC-resistivity and 2D seismic inversion problem are given in section 6. Possible extensions and open problems are discussed in section 7 and section 8 gives the conclusions.

2. All-at-once and reduced methods

A popular approach to solving constrained problems of the form (1) is based on the corresponding Lagrangian:

$$\mathcal{L}(\mathbf{m},\mathbf{u},\mathbf{v}) = \tfrac{1}{2}||P\mathbf{u}-\mathbf{d}||_2^2 + \mathbf{v}^T(A(\mathbf{m})\mathbf{u}-\mathbf{q})\,, \tag{5}$$

where $\mathbf{v}\in\mathbb{C}^{KN}$ is the Lagrange multiplier or adjoint-state variable [14, 7]. A necessary condition for a solution $(\mathbf{m}^*,\mathbf{u}^*,\mathbf{v}^*)$ of the constrained problem (1) is that it is a stationary point of the Lagrangian, i.e. $\nabla\mathcal{L}(\mathbf{m}^*,\mathbf{u}^*,\mathbf{v}^*)=0$. The gradient and Hessian of the Lagrangian are given by

$$\nabla\mathcal{L}(\mathbf{m},\mathbf{u},\mathbf{v}) = \begin{pmatrix}\mathcal{L}_{\mathbf{m}}\\ \mathcal{L}_{\mathbf{u}}\\ \mathcal{L}_{\mathbf{v}}\end{pmatrix} = \begin{pmatrix} G(\mathbf{m},\mathbf{u})^T\mathbf{v},\\ A(\mathbf{m})^T\mathbf{v}+P(P\mathbf{u}-\mathbf{d}),\\ A(\mathbf{m})\mathbf{u}-\mathbf{q},\end{pmatrix}, \tag{6}$$

and

$$\nabla^2\mathcal{L}(\mathbf{m},\mathbf{u},\mathbf{v}) = \begin{pmatrix} R(\mathbf{m},\mathbf{u},\mathbf{v}) & K(\mathbf{m},\mathbf{v})^T & G(\mathbf{m},\mathbf{u})^T\\ K(\mathbf{m},\mathbf{v}) & P^TP & A(\mathbf{m})^T\\ G(\mathbf{m},\mathbf{u}) & A(\mathbf{m}) & 0\end{pmatrix}, \tag{7}$$

where

$$G(\mathbf{m},\mathbf{u}) = \frac{\partial A(\mathbf{m})\mathbf{u}}{\partial\mathbf{m}},\ K(\mathbf{m},\mathbf{v}) = \frac{\partial A(\mathbf{m})^T\mathbf{v}}{\partial\mathbf{m}},$$
$$R(\mathbf{m},\mathbf{u},\mathbf{v}) = \frac{\partial G(\mathbf{m},\mathbf{u})^T\mathbf{v}}{\partial\mathbf{m}}.$$

These Jacobian matrices are typically sparse when A is sparse and can be computed analytically.

In practice we are usually interested in satisfying the optimality conditions up to some tolerance, i.e. $\|\nabla\mathcal{L}(\mathbf{m}^*,\mathbf{u}^*,\mathbf{v}^*)\|_2\leq\epsilon$.

2.1. All-at-once approach

So-called all-at-once approaches find such a stationary point by applying a Newton-like method to the Lagrangian [7]. A basic algorithm is given in Algorithm 1. Many variants of Algorithm 1 exist and may include preconditioning, inexact solves of the KKT system $\left(\nabla^2\mathcal{L}\right)^{-1}\nabla\mathcal{L}$ and a linesearch to ensure global convergence. For an extensive overview we refer to [15].

Algorithm 1 Basic Newton algorithm for find a stationary point of the Lagrangian via the all-at-once method

Require: initial guess $\mathbf{m}^0, \mathbf{u}^0, \mathbf{v}^0$, tolerance ϵ
  $k = 0$
  while $\|\nabla\mathcal{L}(\mathbf{m}^k, \mathbf{u}^k, \mathbf{v}^k)\|_2 \geq \epsilon$ **do**
    $$\begin{pmatrix} \delta\mathbf{m}^k \\ \delta\mathbf{u}^k \\ \delta\mathbf{v}^k \end{pmatrix} = -\left(\nabla^2\mathcal{L}(\mathbf{m}^k, \mathbf{u}^k, \mathbf{v}^k)\right)^{-1} \nabla\mathcal{L}(\mathbf{m}^k, \mathbf{u}^k, \mathbf{v}^k)$$
    determine steplength $\alpha^k \in (0, 1]$
    $\mathbf{m}^{k+1} = \mathbf{m}^k + \alpha^k\delta\mathbf{m}^k$
    $\mathbf{u}^{k+1} = \mathbf{u}^k + \alpha^k\delta\mathbf{u}^k$
    $\mathbf{v}^{k+1} = \mathbf{v}^k + \alpha^k\delta\mathbf{v}^k$
    $k = k + 1$
  end while

An advantage of such an all-at-once approach is that it eliminates the need to solve the PDEs explicitly; the constraints are only (approximately) satisfied upon convergence. However, such an approach is often unfeasible for the large-scale applications we have in mind because it involves simultaneously updating (and hence storing) all the variables.

2.2. Reduced approach

Instead, one usually considers a *reduced* formulation that is obtained by eliminating the constraints from (1). This results in an unconstrained optimization problem:

$$\min_{\mathbf{m}} \left\{\phi(\mathbf{m}) = \tfrac{1}{2}||P\mathbf{u}(\mathbf{m}) - \mathbf{d}||_2^2\right\}, \tag{8}$$

where $\mathbf{u}(\mathbf{m}) = A(\mathbf{m})^{-1}\mathbf{q}$. The resulting optimization problem has a much smaller dimension and can be solved using black-box non-linear optimization methods. In contrast to the *all-at-once* method, the constraints are satisfied at each iteration.

The gradient and the Hessian of ϕ are given by

$$\nabla\phi(\mathbf{m}) = G(\mathbf{m}, \mathbf{u})^T\mathbf{v}, \tag{9}$$

$$\begin{aligned}\nabla^2\phi(\mathbf{m}) = {} & G(\mathbf{m}, \mathbf{u})^T A(\mathbf{m})^{-T} P^T P A(\mathbf{m})^{-1} G(\mathbf{m}, \mathbf{u}) \\ & - K(\mathbf{m}, \mathbf{v})^T A(\mathbf{m})^{-1} G(\mathbf{m}, \mathbf{u}) - G(\mathbf{m}, \mathbf{u})^T A(\mathbf{m})^{-T} K(\mathbf{m}, \mathbf{v}) \\ & + R(\mathbf{m}, \mathbf{u}, \mathbf{v}),\end{aligned} \tag{10}$$

where $\mathbf{v} = A(\mathbf{m})^{-T} P\left(\mathbf{d} - P\mathbf{u}\right)$.

A basic (Gauss-Newton) algorithm for minimizing $\phi(\mathbf{m})$ is given in Algorithm 2. Note that this corresponds to a block-elimination of the KKT system and the iterates automatically satisfy $\mathcal{L}_{\mathbf{u}}(\mathbf{m}^k, \mathbf{u}^k_{\text{red}}, \mathbf{v}^k_{\text{red}}) = \mathcal{L}_{\mathbf{v}}(\mathbf{m}^k, \mathbf{u}^k_{\text{red}}, \mathbf{v}^k_{\text{red}}) = 0$. If the algorithm terminates successfully, the final iterates $(\mathbf{m}^*, \mathbf{u}^*_{\text{red}}, \mathbf{v}^*_{\text{red}})$ additionally satisfy $\|\mathcal{L}_{\mathbf{m}}(\mathbf{m}^*, \mathbf{u}^*_{\text{red}}, \mathbf{v}^*_{\text{red}})\|_2 \leq \epsilon$, so that $\|\nabla\mathcal{L}(\mathbf{m}^*, \mathbf{u}^*_{\text{red}}, \mathbf{v}^*_{\text{red}})\|_2 \leq \epsilon$.

A disadvantage of this approach is that it requires the solution of the PDEs at each update, making it computationally expensive. It also strictly enforces the constraint at each iteration, possibly leading to a very non-linear problem in $\mathbf{m}$.

Algorithm 2 Basic Gauss-Newton algorithm for find a stationary point of the Lagrangian via the reduced method

Require: initial guess $\mathbf{m}^0$, tolerance ϵ
  $k = 0$
  $\mathbf{u}_{\text{red}}^0 = A(\mathbf{m}^0)^{-1}\mathbf{q}$
  $\mathbf{v}_{\text{red}}^0 = A(\mathbf{m}^0)^{-T}P(\mathbf{d} - P\mathbf{u}_{\text{red}}^0)$
  while $\|\mathcal{L}_{\mathbf{m}}(\mathbf{m}^k, \mathbf{u}_{\text{red}}^k, \mathbf{v}_{\text{red}}^k)\|_2 \geq \epsilon$ **do**
    $\mathbf{g}_{\text{red}}^k = G(\mathbf{m}^k, \mathbf{u}_{\text{red}}^k)^T\mathbf{v}_{\text{red}}^k$
    $H_{\text{red}}^k = G(\mathbf{m}^k, \mathbf{u}_{\text{red}}^k)^T A(\mathbf{m}^k)^{-T}P^T P A(\mathbf{m}^k)^{-1}G(\mathbf{m}^k, \mathbf{u}_{\text{red}}^k)$
    determine steplength $\alpha^k \in (0, 1]$
    $\mathbf{m}^{k+1} = \mathbf{m}^k - \alpha^k \left(H_{\text{red}}^k\right)^{-1}\mathbf{g}_{\text{red}}^k$
    $\mathbf{u}_{\text{red}}^{k+1} = A(\mathbf{m}^{k+1})^{-1}\mathbf{q}$
    $\mathbf{v}_{\text{red}}^{k+1} = A(\mathbf{m}^{k+1})^{-T}P(\mathbf{d} - P\mathbf{u}_{\text{red}}^{k+1})$
    $k = k + 1$
  end while

3. Penalty and augmented Lagrangian methods

It is impossible to do justice to the wealth of research that has been done on penalty and augmented Lagrangian methods here but we give a brief overview, high-lighting the main characteristics of a few basic approaches and their limitations when applied to large-scale inverse problems.

A constrained optimization problem of the form (1) can be recast as an unconstrained problem by introducing a positive penalty function $\pi : \mathbb{C}^N \to \mathbb{R}$ and a penalty parameter $\lambda > 0$ as follows

$$\min_{\mathbf{m},\mathbf{u}} \tfrac{1}{2}||P\mathbf{u} - \mathbf{d}||_2^2 + \lambda\pi(\mathbf{A}(\mathbf{m})\mathbf{u} - \mathbf{q}). \tag{11}$$

The idea is that any departure from the constraint is penalized so that the solution of this unconstrained problem will coincide with that of the constrained problem when λ is large enough.

A quadratic penalty function $\pi(\cdot) = \frac{1}{2}||\cdot||_2^2$ leads to a differentiable unconstrained optimization problem (12) whose minimizer coincides with the solution of the constrained optimization problem (1) when $\lambda \to \infty$ [14, Thm. 17.1]. Practical algorithms rely on repeatedly solving the unconstrained problem for increasing values of λ. A common concern with this approach is that the Hessian may become increasingly ill-conditioned as $\lambda \to \infty$ when there are fewer constraints than variables. For PDE-constrained optimization in inverse problems, there are enough constraints ($A(\mathbf{m})$ is invertible) to prevent this. We discuss this limiting case in more detail in section 5.

For certain non-smooth penalty functions, such as $\pi(\cdot) = ||\cdot||_1$, the minimizer of ϕ_λ is a solution of the constrained problem for *any* $\lambda \geq \lambda^*$ for some λ^* [14, Thm. 17.3]. In practice, a continuation strategy is used to find a suitable value for λ. An advantage of this approach is that λ does not become arbritarily large, thus avoiding the ill-conditioning problems mentioned above. A disadvantage is that the resulting unconstrained problem is no longer differentiable. With large-scale applications in

mind, we therefore do not consider exact penalty methods any further in this paper.

Another approach that avoids having to increase λ to infinity is the *augmented Lagrangian* approach (cf. [14]). In this approach, a quadractic penalty $\lambda\|\mathbf{A}(\mathbf{m})\mathbf{u}-\mathbf{q}\|_2^2$ is added to the Lagrangian (5). A standard approach to solve the constrained problem based on the augmented Lagrangian is the *Alternating direction method of multipliers* (ADMM). In its most basic form it relies on minimizing the augmented Lagrangian w.r.t. $(\mathbf{m}, \mathbf{u})$ and subsequently updating the multiplier $\mathbf{v}$ and the penalty parameter λ [16, 17]. This would require us to store the multipliers, which is not feasible for the large-scale problems we have in mind.

In the next two sections, we discuss a computationally efficient algorithm for solving the constrained optimization problem (1) based on a quadratic penalty formulation. This formulation is attractive because it leads to a differentiable, unconstrained, optimization problem. Moreover, the optimization in $\mathbf{u}$ has a closed-form solution which can be computed efficiently, making it an ideal canditate for large-scale problems.

4. A reduced penalty method

Using a quadratic penalty function, the constrained problem (1) is reformulated as

$$\min_{\mathbf{m},\mathbf{u}} \mathcal{P}(\mathbf{m},\mathbf{u}) = \tfrac{1}{2}||P\mathbf{u}-\mathbf{d}||_2^2 + \tfrac{1}{2}\lambda||A(\mathbf{m})\mathbf{u}-\mathbf{q}||_2^2. \tag{12}$$

The gradient and Hessian of $\mathcal{P}$ are given by

$$\nabla\mathcal{P} = \begin{pmatrix} \mathcal{P}_{\mathbf{m}} \\ \mathcal{P}_{\mathbf{u}} \end{pmatrix} = \begin{pmatrix} \lambda G(\mathbf{m},\mathbf{u})^T\left(A(\mathbf{m})\mathbf{u}-\mathbf{q}\right) \\ P(P\mathbf{u}-\mathbf{d}) + \lambda A(\mathbf{m})^T(A(\mathbf{m})\mathbf{u}-\mathbf{q}) \end{pmatrix}, \tag{13}$$

and

$$\nabla^2\mathcal{P} = \begin{pmatrix} \mathcal{P}_{\mathbf{m},\mathbf{m}} & \mathcal{P}_{\mathbf{m},\mathbf{u}} \\ \mathcal{P}_{\mathbf{u},\mathbf{m}} & \mathcal{P}_{\mathbf{u},\mathbf{u}} \end{pmatrix}, \tag{14}$$

where

$$\mathcal{P}_{\mathbf{m},\mathbf{m}} = \lambda(G(\mathbf{m},\mathbf{u})^T G(\mathbf{m},\mathbf{u}) + R(\mathbf{m},\mathbf{u},A(\mathbf{m})\mathbf{u}-\mathbf{q})), \tag{15}$$

$$\mathcal{P}_{\mathbf{u},\mathbf{u}} = P^TP + \lambda A(\mathbf{m})^T A(\mathbf{m}), \tag{16}$$

$$\mathcal{P}_{\mathbf{m},\mathbf{u}} = \lambda(K(\mathbf{m},A(\mathbf{m})\mathbf{u}-\mathbf{q}) + A(\mathbf{m})^T G(\mathbf{m},\mathbf{u})). \tag{17}$$

Of course, optimization in the full $(\mathbf{m},\mathbf{u})$-space is not feasible for large-scale problems, so we eliminate $\mathbf{u}$ by introducing

$$\mathbf{u}_\lambda(\mathbf{m}) = \operatorname*{argmin}_{\mathbf{u}} \mathcal{P}(\mathbf{m},\mathbf{u}), \tag{18}$$

and defining a reduced objective:

$$\phi_\lambda(\mathbf{m}) = \mathcal{P}(\mathbf{m},\mathbf{u}_\lambda(\mathbf{m})). \tag{19}$$

The optimization problem for the state (18) has a closed-form solution:

$$\mathbf{u}_\lambda = \left(A(\mathbf{m})^TA(\mathbf{m}) + \lambda^{-1}P^TP\right)^{-1}\left(A(\mathbf{m})^T\mathbf{q} + \lambda^{-1}P\mathbf{d}\right).$$

The modified system $A^TA + \lambda^{-1}P^TP$ is a low-rank update of the original PDE and incorporates the measurements in the PDE solve. This is the main difference with the conventional reduced approach (cf. Algorithm 2); the estimate of the state is not only based on the physics and the current model, but also on the data.

Following [18, Thm. 1], it is readily verified that the gradient and Hessian of ϕ_λ are given by

$$\nabla\phi_\lambda(\mathbf{m}) = \mathcal{P}_{\mathbf{m}}(\mathbf{m},\mathbf{u}_\lambda), \tag{20}$$

$$\nabla^2\phi_\lambda(\mathbf{m}) = \mathcal{P}_{\mathbf{m},\mathbf{m}}\Phi_\lambda(\mathbf{m},\mathbf{u}_\lambda) - \mathcal{P}_{\mathbf{m},\mathbf{u}}(\mathbf{m},\mathbf{u}_\lambda)\left(\mathcal{P}_{\mathbf{u},\mathbf{u}}(\mathbf{m},\mathbf{u}_\lambda)\right)^{-1}\mathcal{P}_{\mathbf{u},\mathbf{m}}(\mathbf{m},\mathbf{u}_\lambda). \tag{21}$$

Note that $\nabla^2\phi_\lambda$ is the Schur complement of $\nabla^2\mathcal{P}$.

A basic Gauss-Newton algorithm for minimizing ϕ_λ is shown in Algorithm 3. Note

Algorithm 3 Basic Gauss-Newton algorithm for find a stationary point of the Lagrangian via the penalty method

Require: initial guess $\mathbf{m}^0$, penalty parameter λ, tolerance ϵ

$k = 0$

$\mathbf{u}_\lambda^0 = \left(A(\mathbf{m}^0)^T A(\mathbf{m}^0) + \lambda^{-1}P^TP\right)^{-1}\left(A(\mathbf{m}^0)^T\mathbf{q} + \lambda^{-1}P\mathbf{d}\right)$

$\mathbf{v}_\lambda^0 = \lambda(A(\mathbf{m}^0)\mathbf{u}_\lambda^0 - \mathbf{q})$

while $\|\mathcal{L}_{\mathbf{m}}(\mathbf{m}^k,\mathbf{u}_\lambda^k,\mathbf{v}_\lambda^k)\|_2 \geq \epsilon$ **do**

$\mathbf{g}_\lambda^k = G(\mathbf{m}^k,\mathbf{u}_\lambda^k)^T\mathbf{v}_\lambda^k$

$H_\lambda^k = \lambda G^T\left(I - A\left(A^TA + \lambda^{-1}P^TP\right)^{-1}A^T\right)G$

determine steplength $\alpha^k \in (0,1]$

$\mathbf{m}^{k+1} = \mathbf{m}^k - \alpha^k\left(H_\lambda^k\right)^{-1}\mathbf{g}_\lambda^k$

$\mathbf{u}_\lambda^{k+1} = \left(A(\mathbf{m}^{k+1})^TA(\mathbf{m}^{k+1}) + \lambda^{-1}P^TP\right)^{-1}\left(A(\mathbf{m}^{k+1})^T\mathbf{q} + \lambda^{-1}P\mathbf{d}\right)$

$\mathbf{v}_\lambda^{k+1} = \lambda(A(\mathbf{m}^{k+1})\mathbf{u}_\lambda^{k+1} - \mathbf{q})$

$k = k + 1$

end while

that the computation of the adjoint-state $\mathbf{v}_\lambda$ does *not* require an additional PDE-solves in this algorithm. Instead, the forward and adjoint solve are done simulatenously via the normal equations.

Next, we show how the states, $\mathbf{u}_\lambda^k$ and $\mathbf{v}_\lambda^k$, generated by this algorithm relate to the states generated by the reduced approach and subsequently that if the algorithm successfully terminates the iterates $(\mathbf{m}^*,\mathbf{u}_\lambda^*,\mathbf{v}_\lambda^*)$ satisfy $\|\nabla\mathcal{L}(\mathbf{m}^*,\mathbf{u}_\lambda^*,\mathbf{v}_\lambda^*)\|_2 \leq \epsilon + \mathcal{O}(\lambda^{-1})$.

Lemma 4.1 *For a fixed* $\mathbf{m}$*, the states* $\mathbf{u}_\lambda$ *and* $\mathbf{v}_\lambda$ *used in the reduced penalty approach (Algorithm 3) are related to the states* $\mathbf{u}_{\text{red}}$ *and* $\mathbf{v}_{\text{red}}$ *used in the reduced approach (Algorithm 2) as follows*

$$\mathbf{u}_\lambda = \mathbf{u}_{\text{red}} + \mathcal{O}(\lambda^{-1}), \tag{22}$$

$$\mathbf{v}_\lambda = \mathbf{v}_{\text{red}} + \mathcal{O}(\lambda^{-1}). \tag{23}$$

Proof The state variables used in the penalty approach are given by

$$\mathbf{u}_\lambda = \left(A^TA + \lambda^{-1}P^TP\right)^{-1}\left(A^T\mathbf{q} + \lambda^{-1}P\mathbf{d}\right),$$

and

$$\mathbf{v}_\lambda = \lambda(A\mathbf{u}_\lambda - \mathbf{q}).$$

The former can be re-written as

$$\mathbf{u}_\lambda = A^{-1}\left(I + \lambda^{-1}A^{-T}P^TPA^{-1}\right)^{-1}\left(\mathbf{q} + \lambda^{-1}A^{-T}P\mathbf{d}\right).$$

Let $\mu_1(A^{-T}P^TPA^{-1})$ denote the largest eigenvalue of $A^{-T}P^TPA^{-1}$. For $\lambda > \mu_1\mu_1(A^{-T}P^TPA^{-1})$ we may expand the inverse as $(I + \lambda^{-1}B)^{-1} \approx I - \lambda^{-1}B + \lambda^{-2}B^2 + \ldots$, and find that

$$\begin{aligned}\mathbf{u}_\lambda &= A^{-1}\mathbf{q} \\ &\quad + \lambda^{-1}\left(A^TA\right)^{-1}P\left(\mathbf{d} - PA^{-1}\mathbf{q}\right) \\ &\quad - \lambda^{-2}\left(A^TA\right)^{-1}P^TP\left(A^TA\right)^{-1}P\mathbf{d} + \mathcal{O}(\lambda^{-3}) \\ &= \mathbf{u}_{\text{red}} + \lambda^{-1}A\mathbf{v}_{\text{red}} + \mathcal{O}(\lambda^{-2}). \end{aligned} \tag{24}$$

We immediately find

$$\mathbf{v}_\lambda = \mathbf{v}_{\text{red}} + \mathcal{O}(\lambda^{-1}). \tag{25}$$

∎

Remark Lemma 4.1 suggests a natural scaling for the penalty parameter, $\lambda > \mu_1(A^{-T}P^TPA^{-1})$ can be considered large, while $\lambda < \mu_1(A^{-T}P^TPA^{-1})$ can be considered small.

Theorem 4.2 *At each iteration of algorithm 3, the iterates satisfy* $\|\mathcal{L}_{\mathbf{u}}(\mathbf{m}^k, \mathbf{u}_\lambda^k, \mathbf{v}_\lambda^k)\|_2 = \mathcal{O}(\lambda^{-1})$ *and* $\|\mathcal{L}_{\mathbf{v}}(\mathbf{m}^k, \mathbf{u}_\lambda^k, \mathbf{v}_\lambda^k)\|_2 = 0$ *Moreover, if algorithm 3 terminates successfully at* $\mathbf{m}^*$ *for which* $\|\mathcal{L}_{\mathbf{m}}(\mathbf{m}^*, \mathbf{u}_\lambda^*, \mathbf{v}_\lambda^*)\|_2 \leq \epsilon$*, we have* $\|\nabla\mathcal{L}(\mathbf{m}^*, \mathbf{u}_\lambda^*, \mathbf{v}_\lambda^*)\|_2 \leq \epsilon + \mathcal{O}(\lambda^{-1})$.

Proof Using the definitions of $\mathbf{u}_\lambda$ and $\mathbf{v}_\lambda$ we find for any $\mathbf{m}$

$$\begin{aligned}\mathcal{L}_{\mathbf{u}}(\mathbf{m}, \mathbf{u}_\lambda, \mathbf{v}_\lambda) &= A(\mathbf{m})^T\mathbf{v}_\lambda + P(P\mathbf{u}_\lambda - \mathbf{d}) \\ &= \lambda A^T(A\mathbf{u}_\lambda - \mathbf{q}) + P(P\mathbf{u}_\lambda - \mathbf{d}) = 0. \end{aligned} \tag{26}$$

Using the approximations for $\mathbf{u}_\lambda$ and $\mathbf{v}_\lambda$ for $\lambda > \mu_1(A^{-T}P^TPA^{-1})$ presented in Lemma 4.1, we find

$$\begin{aligned}\mathcal{L}_{\mathbf{v}}(\mathbf{m}, \mathbf{u}_\lambda, \mathbf{v}_\lambda) &= A(\mathbf{m})\mathbf{u}_\lambda - \mathbf{q} \\ &= \lambda^{-1}A(\mathbf{m})^{-T}P\left(\mathbf{d} - PA(\mathbf{m})^{-1}\mathbf{q}\right) + \mathcal{O}(\lambda^{-2}). \end{aligned} \tag{27}$$

Thus we find

$$\|\mathcal{L}_{\mathbf{v}}(\mathbf{m}^*, \mathbf{u}_\lambda^*, \mathbf{v}_\lambda^*)\|_2 = \mathcal{O}(\lambda^{-1}). \tag{28}$$

At a point $\mathbf{m}^*$ for which $\|\mathcal{L}_{\mathbf{m}}(\mathbf{m}^*, \mathbf{u}_\lambda^*, \mathbf{v}_\lambda^*)\|_2 \leq \epsilon$ we immediately find that

$$\begin{aligned}&\|\nabla\mathcal{L}(\mathbf{m}^*, \mathbf{u}_\lambda^*, \mathbf{v}_\lambda^*)\|_2^2 = \\ &\|\mathcal{L}_{\mathbf{m}}(\mathbf{m}^*, \mathbf{u}_\lambda^*, \mathbf{v}_\lambda^*)\|_2^2 + \|\mathcal{L}_{\mathbf{u}}(\mathbf{m}^*, \mathbf{u}_\lambda^*, \mathbf{v}_\lambda^*)\|_2^2 + \|\mathcal{L}_{\mathbf{v}}(\mathbf{m}^*, \mathbf{u}_\lambda^*, \mathbf{v}_\lambda^*)\|_2^2 \\ &\leq \epsilon^2 + \mathcal{O}(\lambda^{-2}), \end{aligned} \tag{29}$$

and hence that

$$\|\nabla\mathcal{L}(\mathbf{m}^*, \mathbf{u}_\lambda^*, \mathbf{v}_\lambda^*)\|_2 \leq \epsilon + \mathcal{O}(\lambda^{-1}). \tag{30}$$

∎

5. Algorithm

In this section, we discuss some practicalities of the implementation of algorithm 3. We slightly elaborate the notation to explicitly reveal the multi-experiment structure of the problem. In this case, the data are acquired in a series of K independent experiments and $\mathbf{d} = [\mathbf{d}_1; \ldots; \mathbf{d}_K]$ is a block-vector. We partition the states and sources in a simular manner and, since the experiments are independent, the system matrix A is block-diagonal matrix with K blocks $A_i(\mathbf{m})$ of size $N \times N$. Similarly, the matrix P consists of blocks P_i. Recall that we collect L independent measurements for each experiment, so the matrices $P_i \in \mathbb{R}^{N\times L}$ have full rank.

5.1. Solving the augmented PDE

Due to the block structure of the problem, the linear systems can be solved independently. We can obtain the state $\mathbf{u}_i$ by solving the following inconsistent overdetermined system

$$\begin{pmatrix} A_i(\mathbf{m}) \\ \lambda^{-1/2} P_i \end{pmatrix} \mathbf{u}_i \approx \begin{pmatrix} \mathbf{q}_i \\ \lambda^{-1/2} \mathbf{d}_i \end{pmatrix}, \tag{31}$$

in a least-squares sense. Assuming that all the blocks A_i and P_i are identical, we will drop the subscript i for the remainder of this subsection. Next, we will discuss various approaches to solving the overdetermined system (31).

Factorization: If both A and P are sparse, we can efficiently solve the system via a QR factorization or via a Cholesky factorization of the corresponding Normal equations. In many applications, P^TP is a (nearly) diagonal matrix and thus the augmented system $A^TA+\lambda^{-1}P^TP$ has a similar sparsity pattern as the original system. Thus, the fill-in will not be worse than when factorizing the original system.

Iterative methods: While we can make use of factorization techniques for small-scale applications, industry-scale applications will typically require (preconditioned) iterative methods. Obviously, we can apply any preconditioned iterative method that is suitable for solving least-squares problems, such as LSQR, LSMR or CGLS [19, 20, 21]. Another promising candidate is a generic accelerated row-projected method described by [22, 23] which proved usefull for solving PDEs and can be easily extended to deal with overdetermined systems [24].

To get an idea of how such iterative methods will perform we explore some of the properties of the augmented system. The augmented system $A^TA + \lambda^{-1}P^TP$ is a rank L modification of the original system A^TA. It follows from [25, Thm 8.1.8] that the eigenvalues are related as

$$\mu_n(A^TA + \lambda^{-1}P^TP) = \mu_n(A^TA) + a_n\lambda^{-1}, n = 1, 2, \ldots, N \tag{32}$$

where $\mu_1(B) > \mu_2(B) > \ldots > \mu_N(B)$ denote the eigenvalues of B and the coefficients a_n satisfy $\sum_{n=1}^{N} a_n = L$. This means that at worst, 1 eigenvalue is shifted by $L\lambda^{-1}$ while at best, all the eigenvalues are shifted by $LN^{-1}\lambda^{-1}$. For the condition numbers $\kappa(B) = \mu_1(B)/\mu_N(b)$ we find

$$C_N^{-1}\kappa(A^TA) \leq \kappa(A^TA + \lambda^{-1}P^TP) \leq C_1\kappa(A^TA), \tag{33}$$

where $C_i = \left(1 + \frac{L}{\lambda\mu_i(A^TA)}\right)$.

To illustrate this, we show a few examples for a 1D (time-harmonic) parabolic PDE $\left(\imath\omega - \partial_x^2\right) u = 0$ and the 1D Helmholtz equation $\left(\omega^2 + \partial_x^2\right) u = 0$, both with

Neumann boundary conditions. Both are discretized using first-order finite-differences on $x \in [0,1]$ with $N = 51$ points for $\omega = 10\pi$. The sampling matrix P consists of L columns of the identity matrix (regularly sampled). The ratio of the condition numbers of A^TA and $A^TA + \lambda^{-1}P^TP$ for the parabolic and Helmholtz equation are shown in tables 2 and 3. For these examples, the condition number of the augmented system is actually lower than that of the original system. The eigenvalues are shown in figures 1 and 2. These show that the actual eigenvalues distributions do not change significantly. We expect that iterative methods will perform similarly on the augmented system as they would on the original system. *How* to effectively precondition the augmented system given a good preconditioner for the original system is a different matter which is outside the scope of this paper.

Direct methods: When the matrix has additional structure we might actually prefer a direct method over an iterative method. An example is explicit time-stepping, where the system matrix A exhibits a lower-triangular block-structure. In this case the action of A^{-1} can be computed efficiently via forward substitution, requiring storage of only a few time-slices of the state. The adjoint system A^{-T} can be solved by backward substitution, however, the full time-history of the state variable is needed to compute the gradient. For the penalty method, the augmented system $A^TA + P^TP$ will have a banded structure and the system can be solved using a block version of the Thomas algorithm, which again would require storage of the full time-history. So even in this setting it seems possible to apply the penalty method at roughly the same complexity as the reduced method.

5.2. Gradient and Hessian computation

Given these solutions $\mathbf{u}_k$ of (31), the gradient, $\mathbf{g}_\lambda$ and Gauss-Newton Hessian H_λ of ϕ_λ are given by (cf eqs. 20-21)

$$\mathbf{g}_\lambda = \lambda \sum_{k=1}^{K} G_k^T \left(A_k \mathbf{u}_k - \mathbf{q}_k \right), \tag{34}$$

$$H_\lambda = \lambda \sum_{k=1}^{K} G_k^T \left(I - A_k \left(A_k^T A_k + \lambda^{-1} P_k P_k^T \right)^{-1} A_k^T \right) G_k, \tag{35}$$

where $G_k = G(\mathbf{m}, \mathbf{u}_k)$. We can compute the inverse of $\left(A_k^T A_k + \lambda^{-1} P_k P_k^T \right)$ in the same way as used when solving for the states. In practice, we would solve for one state at a time and aggregate the gradient on the fly.

5.3. Complexity estimates

Assuming we can solve the overdetermined system (31) as efficiently as the original PDE, the evaluation of the gradient requires a factor of 2 less computation and storage as the gradient in the reduced approach. A summary of the leading order computational costs of the penalty, reduced and all-at-once approaches is given in table 4.

6. Case studies

The following experiments are done in Matlab, using direct factorization to solve the PDEs (with Matlab `slash`). We consider both a Gauss-Newton (GN) and a Quasi-

Newton (QN) variant of the algorithms and use a weak Wolfe linesearch to determine the steplength. In the GN method the Hessian is inverted using conjugate gradients (`pcg`) up to a relative tolerance of η. The matrix-vector products are computed on the fly. For the QN method we use the L-BFGS inverse Hessian with a history size of 5 [14]. We measure the cost of the inversion by counting the number of PDE solves as outlined in table 4. In all experiments, we set λ relative to the largest eigenvalue of $A^{-T}P^TPA^{-1}$ at the initial iterate. This scaling is justified by remark 4.

To avoid the inverse crime, we compute the data for the ground truth model on a finer grid than used for the inversion.

In these experiments, we illustrate that the penalty method:

- converges to a stationary point of the Lagrangian within the predicted tolerance of $\mathcal{O}(\lambda^{-1})$;
- gives practically the same or a better results as the reduced method at a lower computational cost;
- is not overly sensitive to noise;
- leads to a less non-linear optimization problem than the conventional reduced approach.

The Matlab code used to perform the experiments is available from `https://github.com/tleeuwen/Penalty-Method`.

6.1. 1D DC resistivity

We consider the PDE

$$\partial_t u(t,x) = \partial_x \left(m(x)\partial_x\right) u(t,x), \tag{36}$$

on the domain $x = [0,1]$ with Neumann boundary conditions. A finite-difference discretization in the temporal Fourier domain gives

$$A(\mathbf{m}) = \imath\omega \mathsf{diag}(\mathbf{w}) + D^T \mathsf{diag}(\mathbf{m}) D, \tag{37}$$

where ω is the angular frequency, $\mathbf{w} = [\frac{1}{2}; 1, \ldots, 1; \frac{1}{2}]$, $\mathbf{m}$ represents the medium parameter in the cell-centres and D is the $N-1 \times N$ finite-difference matrix

$$D = \frac{1}{h}\begin{pmatrix} -1 & 1 & & & \\ & -1 & 1 & & \\ & & \ddots & \ddots & \\ & & & -1 & 1 \end{pmatrix},$$

with $h = 1/(N-1)$. The Jacobian is given by

$$G(\mathbf{m}, \mathbf{u}) = D^T \mathsf{diag}(D\mathbf{u}).$$

The ground-truth model is $m(x) = 1 + e^{-10(x-1/2)^2}$ and we locate two sources and receivers on either end of the domain. The data are generated on a grid with $N = 201$ points and we have $K = L = 2$.

For the inversion we use $N = 101$ points. We use a GN method with $\epsilon = 10^{-9}$, $\eta = 10^{-3}$ and include a regularization term $\frac{\alpha}{2}\|D\mathbf{m}\|_2^2$ with $\alpha = 10^{-6}$. The initial parameters are $\mathbf{m}^0 = \mathbf{1}$.

The results are shown in figure 3. The convergence plot, figure 3 (a), shows the predicted behaviour of the penalty method; the norm of the gradient of the Laplacian stalls at $\mathcal{O}(\lambda^{-1})$. The resulting parameter estimates are very similar as can be seen in

figure 3 (b). The actual costs of the inversion are listed in table 5. The computational cost for the various approaches are of the same order of magnitude, except for $\lambda = 10$, where more than twice as many iterations are required.

6.2. 2D Acoustic tomography

Consider the 2D scalar wave-equation

$$m(x)\partial_t^2 u(t,x) = \nabla^2 u(t,x), \tag{38}$$

on $x \in \Omega \subseteq \mathbb{R}^2$ with radiation boundary conditions $\sqrt{m}(x)\partial_t u(t,x) - n(x)\cdot\nabla u(t,x) = 0$ on $\partial\Omega$ where $n(x)$ is the outward normal vector.

Discretization in the temporal Fourier domain leads to a scalar Helmholtz equation

$$A(\mathbf{m}) = \mathsf{diag}(\mathbf{s}) - D^T D, \tag{39}$$

where $D = [I_2 \otimes D_1; D_2 \otimes I_1]$ with D_i the $(N_i - 1) \times N_i$ finite-difference matrix, I_i the $N_i \times N_i$ identity matrix and $s_i = \omega^2 m_i$ in the interior and $s_i = \omega^2 m_i/2 + \imath\omega\sqrt{m_i}/h$ on the boundary. The Jacobian is given by

$$G(\mathbf{m}, \mathbf{u}) = \mathsf{diag}(\mathbf{s}')\mathsf{diag}(\mathbf{u}), \tag{40}$$

where $s_i' = \omega^2$ in the interior and $s_i' = (\omega^2 + \imath\omega/\sqrt{m_i})/2$ on the boundary.

The observation matrix P is samples the solution at the receiver locations using 2D linear interpolation while the point sources are defined using adjoint 2D linear interpolation.

6.2.1. Ultrasound tomography The domain $\Omega = [0,1] \times [0,1]$ m is discretized using $N_1 \times N_2$ points. The ground-truth $\mathbf{m}^*$ as well as the source and receiver locations are shown in figure 4. We use a single frequency of $5kHz$ (i.e., $\omega = 10^4\pi$). The data for the ground-truth model are generated using $N_1 = N_2 = 101$ while the following experiments are done with $N_1 = N_2 = 51$.

Non-linearity: First, we investigate the sensitivity of the misfit functions ϕ and ϕ_λ by plotting $\phi(\mathbf{m}^* + \delta_1\mathbf{v}_1 + \delta_2\mathbf{v}_2)$ and $\phi_\lambda(\mathbf{m}^* + \delta_1\mathbf{v}_1 + \delta_2\mathbf{v}_2)$ as a function of (δ_1, δ_2). We take $\mathbf{v}_1, \mathbf{v}_2$ to be slowly oscillatory modes as shown in figure 5. The misfit as a function of (δ_1, δ_2) is shown in figure 6. We see a radically different behaviour for the reduced and penalty methods. The first exhbits strong non-linearity and some spurious stationary points while for small λ the penalty misfit is much better behaved. For larger values λ the penalty misfit starts to behave more like the reduced misfit as expected.

Inversion: For the inversion, we include a regularization term $\frac{\alpha}{2}\|D\mathbf{m}\|_2^2$ with $\alpha = 2$ and compare the GN method ($\epsilon = 10^{-6}$, $\eta = 10^{-1}$) to the QN method ($\epsilon = 10^{-6}$). The initial parameter $\mathbf{m}_0$ is constant at $\frac{1}{4}\ s^2/m^2$.

The results for the GN method are shown in figure 7. The convergence history, figure 7 (top, left), shows the predicted behaviour of the penalty method; the norm of the gradient of the Lagrangian stalls at $\mathcal{O}(\lambda^{-1})$ when using the penalty method. Figure 7 (top, right) shows that the methods perform similarly in terms of reconstruction error. The resulting parameter estimates are very similar as can be seen in figure 7 (bottom). The actual costs of the inversion are listed in table 6. The penalty method converges in less iterations and uses less PDE solves per iterations. Note that all methods start overfitting after a few iterations. This can be countered by including

more appropriate regularization or stopping the iterations early. The point here is to show that the penalty method gives simular results as the reduced method.

The results for the QN method are shown in figure 8. The convergenge history shows the same behaviour as the previous experiment. The costs of the inversion, shown in table 7, are slightly less than those of the GN method. As with the GN method, the penalty method converges in less iterations and uses less PDE solves per iterations.

Sensitivity to noise: Results for the QN method on data with 10% Gaussian noise are shown in figure 9. Figure 10 shows the results on data with 20% Gaussian noise. These results show that the penalty approach is not overly sensitive to noise and gives very similar –even slightly better– results compared to the reduced approach.

6.2.2. Seismic tomography Here, the domain $\Omega = [0, 5] \times [0, 20]$ km is discretized using $N_1 \times N_2$ points. The ground-truth $\mathbf{m}^*$ as well as the source and receiver locations are shown in figure 11. We use a frequency of $2Hz$ (i.e., $\omega = 4\pi$). The data for the ground-truth are generated using $N_1 = 101, N_2 = 401$ while the following experiments are done with $N_1 = 51, N_2 = 201$.

Sensitivity to the initial guess For the inversion, we include a regularization term $\frac{\alpha}{2}\|D\mathbf{m}\|_2^2$ with $\alpha = 5$ and use the QN method ($\epsilon = 10^{-6}$). We will use two different initial guesses, I and II, depicted in figure 11, and see whether the methods converge to the same final iterate. Initial iterate I is much closer to the ground truth than the initial iterate II. This can also be observed when looking at the data produced by these iterates. The first initial iterate produces data that differs only slightly from the observed data and inversion is considered to be easy. The second initial iterate produces data that is shifted significantly with respect to the observed data and inversion is considered to be difficult.

The results for initial guess I (figure 11, middle) are shown in figure 12. We see that both the reduced and penalty methods converge to roughly the same final iterate and are able to fit the data equally well. Starting from initial guess II (figure 11, bottom), however, we see that the reduced and penalty methods converge to different final iterates. For small λ, however, the penalty method converges to roughly the same iterate as when starting from a better initial guess. Looking at the data-fit, we observe that the penalty method for small λ is still able to fit the data perfectly while the reduced method is not.

Figure 14 shows the convergence of the methods in terms of the data misfit $\|P\mathbf{u} - \mathbf{d}\|_2$ and the distance to the constraint $\|A(\mathbf{m})\mathbf{u} - \mathbf{q}\|_2$. We observe that, when starting from initial guess I, both the penalty and reduced methods converge to approximately the same point. For initial guess II the penalty method for $\lambda = 0.1$ and $\lambda = 1$ needs a few more iterations, but still converges to the same point as for initial guess I. For $\lambda = 10$ and the reduced method, however, the iterations stall at a relatively high data misfit.

These experiments suggests that the penalty indeed mitigates some of the non-linearity of the problem, allowing the optimization to converge to the same final iterate, even when the initial guess is further away from the ground truth.

7. Discussion

This paper lays out the basics of an efficient implementation of the penalty method for PDE-constrained optimization problems arising in inverse problems. While the

initial results are promising, some aspects of the proposed method warrant further investigation.

While the theoretical results suggest that the penalty approach can find a stationary point of the Lagrangian with finite precision with a finite λ, it is not clear how to choose a suitable value for λ a priori. Our analysis and results suggest that choosing λ to be a small fraction of the largest eigenvalue of $A^{-T}P^TPA^{-1}$ at the initial iterate yields good results. A more solid justification of this observation is needed to design robust algorithms.

Similarly, a continuation strategy for λ is needed if we want to guarantee finding a stationary point of the Lagrangian with preset tolerance. A natural way to do this seems to by detecting when the penalty method stalls and subsequently reducing λ.

Finally, the Hessian of the penalty objective exhibits additional structure that could potentially be exploited. In particular, the penalty-method GN Hessian is full rank and allows for a natural sparse approximation $H_\lambda \approx \lambda G^TG$ (cf. equation 17). The reduced GN Hessian, on the other hand, has rank of at most ML and does not permit such a natural sparse approximation.

8. Conclusions

We have presented a penalty method for PDE-constrained optimization with linear PDEs with applications to inverse problems. The method is based on a quadratic penalty formulation of the constrained problem. This reformulation results in a an unconstrained optimization problem in both the parameters and the state variables. To avoid having to store and update the state variables as part of the optimization, we explicitly eliminate the state variables by solving an overdetermined linear system. The proposed method combines features from both the *all-at-once* approach, in which the states and parameters are updated simultaneously, and the conventional *reduced* approach, in which the PDE-constraints are eliminated explicitly. While having a similar computational complexity as the conventional reduced approach, the penalty approach explores a larger search space by not satisfying the PDE-constraints exactly.

We show that we can (theoretically) find a stationary point of the Lagrangian of the constrained problem within a given tolerance as long as the penalty parameter, λ, is chosen large enough. While theoretically we need $\lambda \uparrow \infty$, we can suffice with solving the problem for a finite λ to reach the stationary point within finite precision.

The main algorithmic difference with the conventional reduced approach is the way the states are eliminated from the problem. Instead of solving the PDEs, we formulate an overdetermined system of equations that consists of the discretized PDE and the measurements. We discuss the properties of this augmented system and show with a few numerical examples that both the structure of the system as well as the eigenvalues are not altered dramatically as compared the original PDE. Thus, it is plausible that the augmented system can be solved as efficiently using the same approach as is used for the original PDE.

The numerical examples show that very good results can be obtained by using even a single, relatively small, value of λ. The numerical examples further show that by enlarging the search space, the optimzation problem may actually be less non-linear and that in some cases a better parameter reconstruction is obtained as compared to the conventional reduced approach. In particular, the results show that the penalty method is not overly sensitive to noise and less sensitive to the initial model than the conventional reduced approach.

Thus, the proposed approach is a viable alternative to the conventional reduced approach for solving inverse problems with PDE-constraints.

Acknowledgments

This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada Discovery Grant (22R81254) and the Collaborative Research and Development Grant DNOISE II (375142-08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BGP, BP, Chevron, CGG, ConocoPhillips, ION, Petrobras, PGS, Total SA, WesternGeco and Woodside.

	small 2D	large 2D	industrial 3D
K	10^2	10^3	10^6
L	10^2	10^3	10^6
M	10^6	10^9	10^{12}
N	10^3	10^6	10^9

Table 1. Typical size of seismic inverse problem in terms K: of the number of experiments, L: the number of measurements per experiment, N: the number of discretization points and M: the number of parameters.

	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
L = 1	9.83e-01	9.84e-01	9.89e-01
L = 10	9.64e-02	5.06e-01	9.10e-01
L = 20	9.16e-02	5.00e-01	9.09e-01

Table 2. Ratio of the condition numbers of $A^TA + \lambda P_L^T P_L$ and A^TA for various λ and L, where A is a finite-difference discritization of $i\omega - \partial_x\left(m(x)\partial_x\right)$ and P_L is a restricted identify matrix of rank L.

	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
L = 1	1.80e-01	5.44e-01	9.17e-01
L = 10	1.03e-01	5.06e-01	9.10e-01
L = 20	9.10e-02	5.00e-01	9.09e-01

Table 3. Ratio of the condition numbers of $A^TA + \lambda P_L^T P_L$ and A^TA for various λ and L, where A is a finite-difference discritization of $\omega^2 m + \partial_x^2$ and P_L is a restricted identify matrix of rank L.

	# PDE's	Storage	Gauss-Newton update
penalty	K	$N + M$	solve matrix-free linear system in M unknowns, requires K (overdetermined) PDE solves per mat-vec
reduced	$2K$	$2N + M$	solve matrix-free linear system in M unknowns, requires $2K$ PDE solves per mat-vec
all-at-once	0	$2KN + M$	solve sparse symmetric, possibly indefinite system in $(2KN + M) \times (2KN + M)$ unknowns

Table 4. Leading order computation and storage costs per iteration of different methods; K denotes the number of experiments and N denotes the number of gridpoints and M denotes the number of parameters.

	reduced	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
iterations	6	6	6	15
PDE solves	368	206	193	682

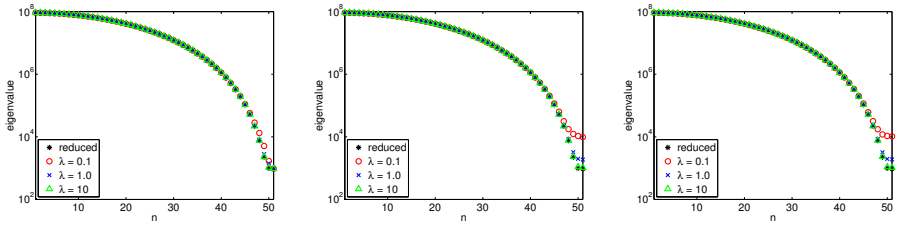
Table 5. Costs of the 1D DC resistivity inversion.

	reduced	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
iterations	6	4	5	6
PDE solves	172	38	55	82

Table 6. Costs of the 2D ultrasound inversion with a GN method.

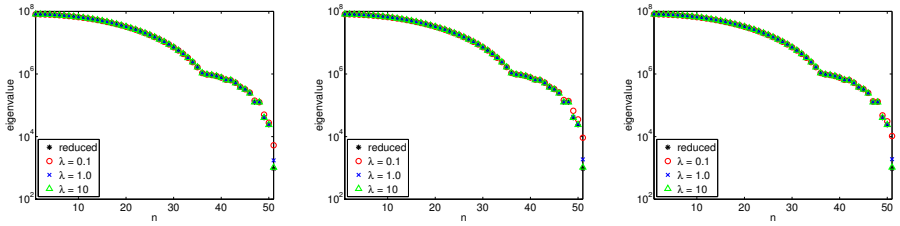
	reduced	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
iterations	67	32	48	60
PDE solves	148	34	51	67

Table 7. Costs of the 2D ultrasound inversion with a QN method.



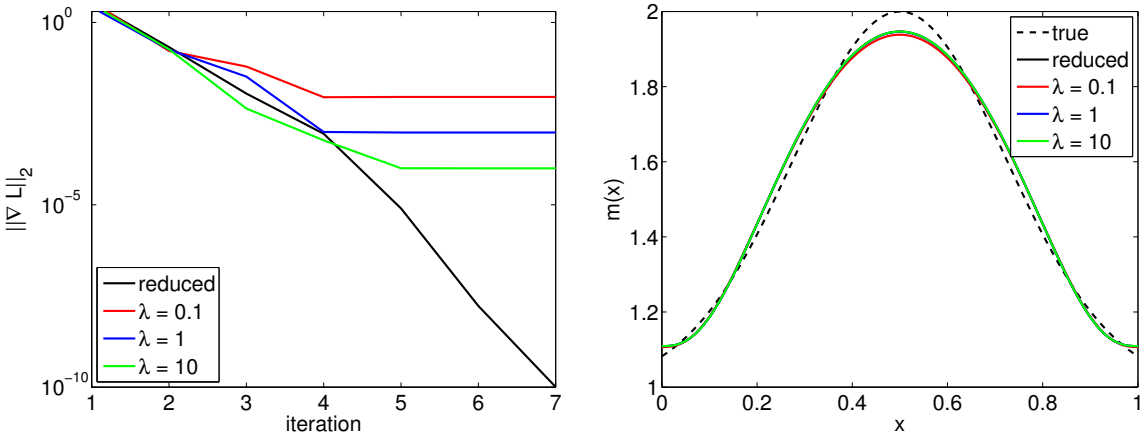
$L = 1$ $L = 10$ $L = 20$

Figure 1. Eigenvalues of the augmented system, $A^TA + \lambda P_L^T P_L$, for various λ and L, where A is a finite-difference discritization of $\imath\omega - \partial_x\,(m(x)\partial_x)$ and P_L is a restricted identify matrix of rank L. For comparison, the eigenvalues of the original system A^TA are also shown.



$L = 1$ $L = 10$ $L = 20$

Figure 2. Eigenvalues of the augmented system, $A^TA + \lambda P_L^T P_L$, for various λ and L, where A is a finite-difference discritization of $\omega^2 + \partial_x^2$ and P_L is a restricted identify matrix of rank L. For comparison, the eigenvalues of the original system A^TA are also shown.



(a) (b)

Figure 3. Solutions and convergence history for 1D resisivity problem. Even though the penalty method does not converge to same tolerance as the reduced method in terms of the gradient of the Lagrangian, the resulting parameter estimates are almost the same.

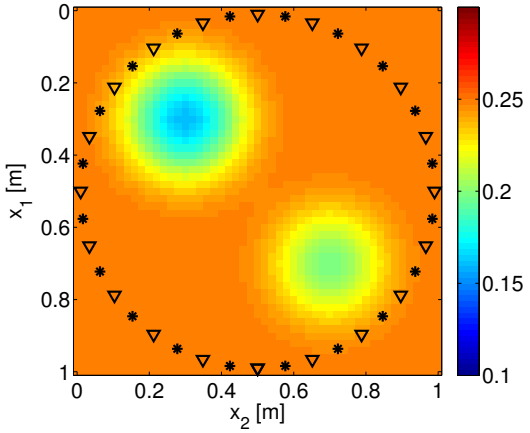


Figure 4. Ground truth model (s^2/km^2) and locations of the sources ($*$) and receivers ($\triangledown$)

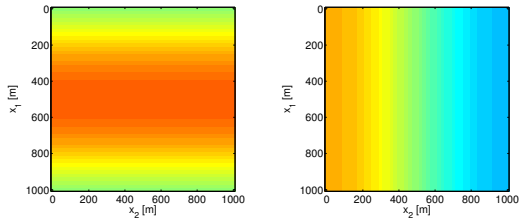


Figure 5. Perturbations $\mathbf{v}_1$ and $\mathbf{v}_2$ used to plot the misfit $\phi(\mathbf{m}^* + \delta_1\mathbf{v}_1 + \delta_2\mathbf{v}_2)$ and $\phi_\lambda(\mathbf{m}^* + \delta_1\mathbf{v}_1 + \delta_2\mathbf{v}_2)$ in figure 6.

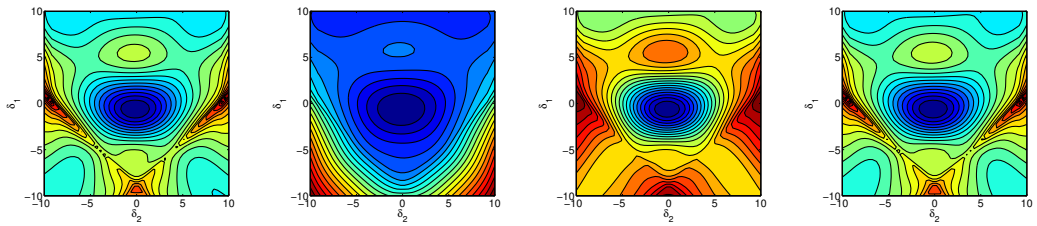


Figure 6. Misfit in the direction of the perturbations shown in figure 5. For small λ, the reduced penalty objective ϕ_λ is less non-linear than the reduced objective ϕ.

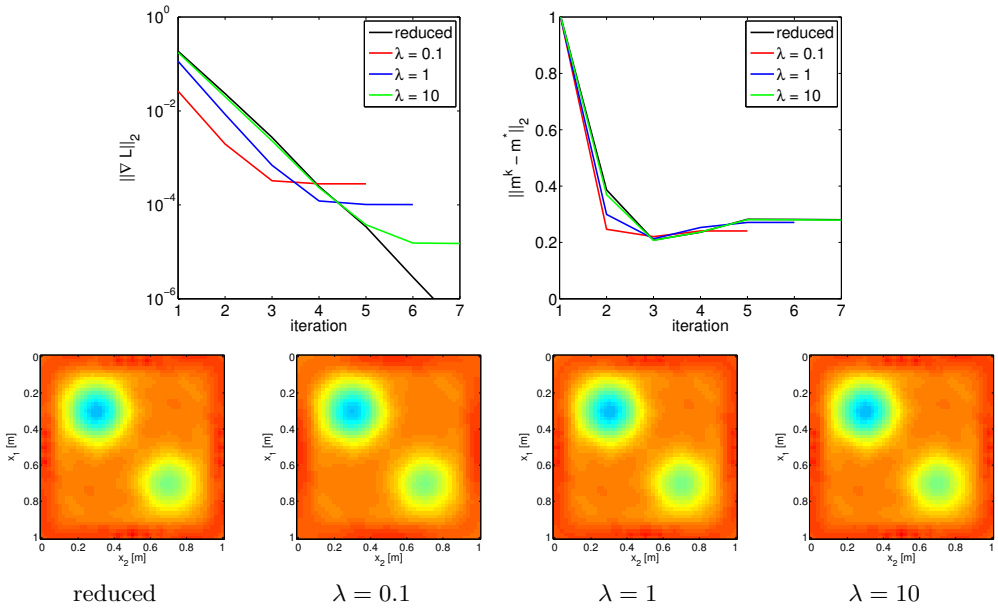


Figure 7. Convergence history, GN reconstruction error and reconstructions for data without noise. Even though the penalty method does not converge to same tolerance as the reduced method in terms of the gradient of the Lagrangian, the resulting parameter estimates are almost the same.

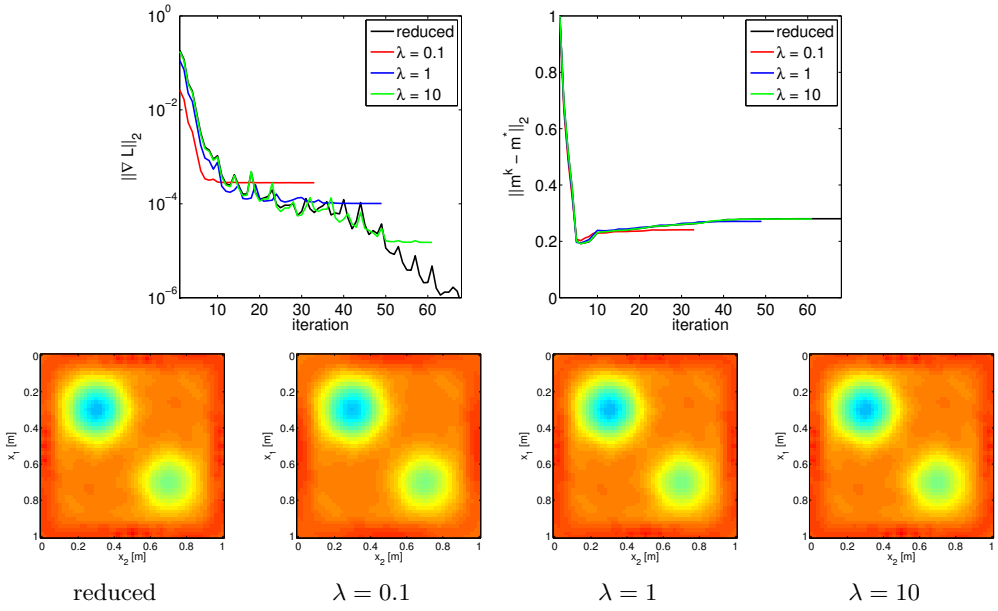


Figure 8. Convergence history, QN reconstruction error and reconstructions for data without noise. Even though the penalty method does not converge to same tolerance as the reduced method in terms of the gradient of the Lagrangian, the resulting parameter estimates are almost the same.

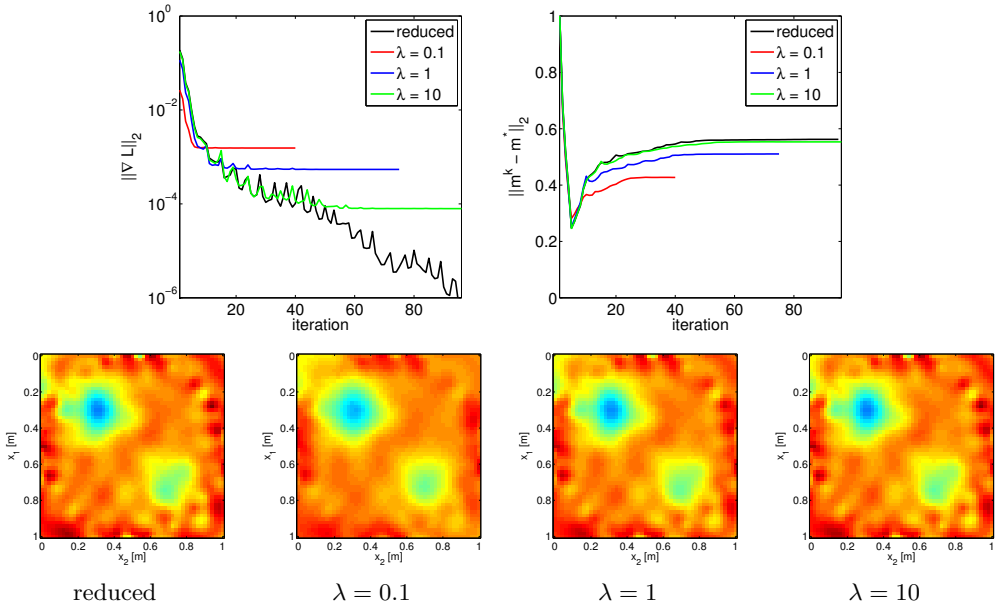


Figure 9. Convergence history, QN reconstruction error and reconstructions for data with 10% Gaussian noise. Even though the penalty method does not converge to same tolerance as the reduced method in terms of the gradient of the Lagrangian, the resulting parameter estimates are almost the same. In fact, for small λ, result is even a little better.

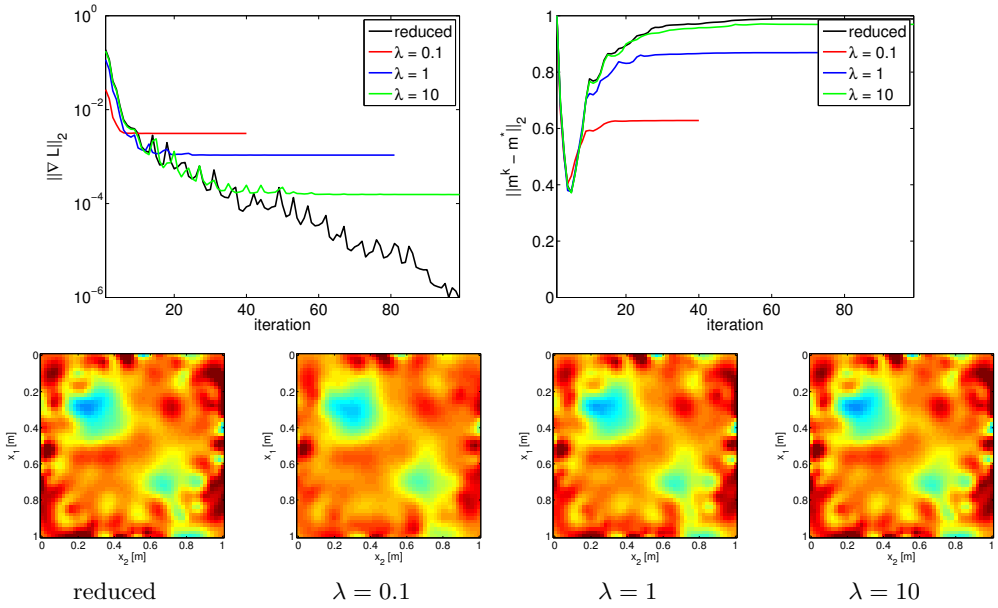


Figure 10. Convergence history, QN reconstruction error and reconstructions for data with 20% Gaussian noise. Even though the penalty method does not converge to same tolerance as the reduced method in terms of the gradient of the Lagrangian, the resulting parameter estimates are almost the same. In fact, for small λ, result is even a little better.

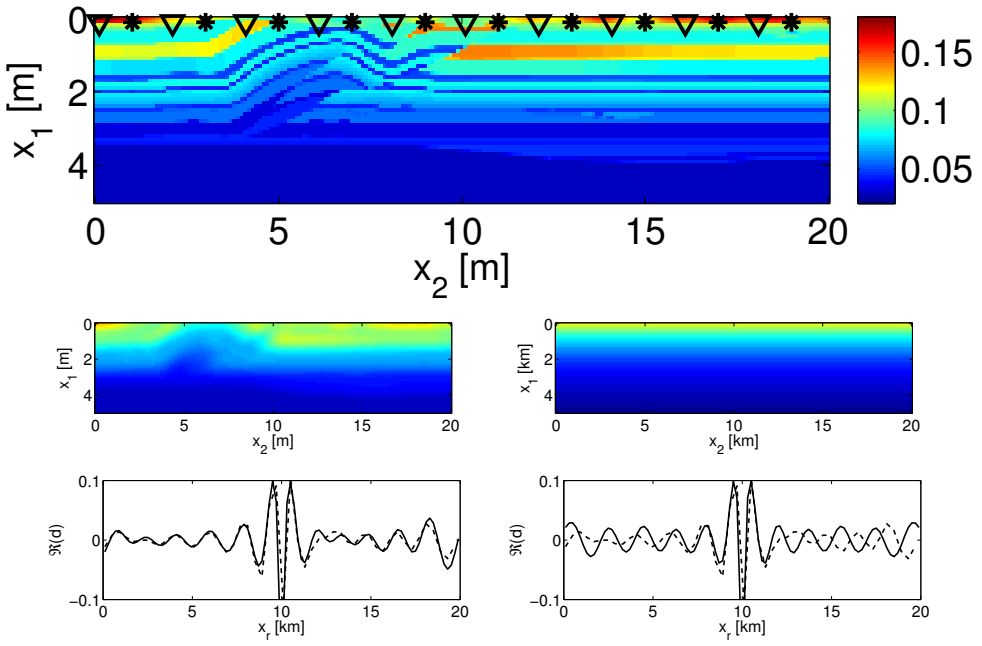


Figure 11. Ground truth (s^2/km^2) (top) with locations of the sources ($*$) and receivers ($\bigtriangledown$) and initial iterates I (middle, left) and II (bottom, right). The bottom row shows the data for a source in the centre for the ground truth (dashed line) as well as the data for the two initial iterates. The first initial iterate produces data that differs only slightly from the observed data and inversion is considered to be easy. The second initial iterate produces data that is shifted significantly with respect to the observed data and inversion is considered to be difficult.

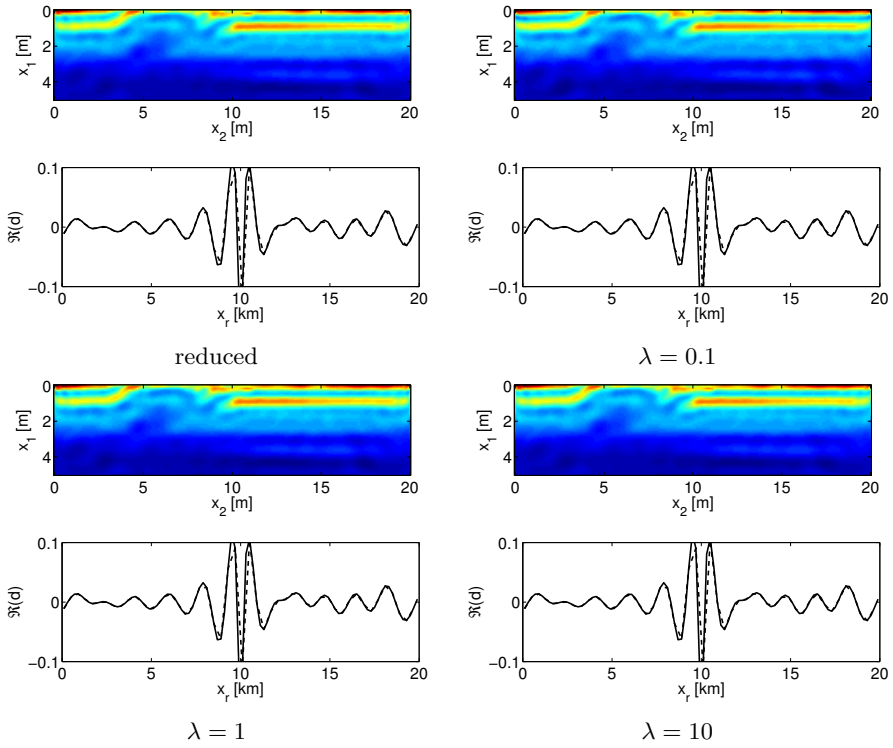


Figure 12. QN reconstructions after 50 iterations and corresponding data for a source in the center, starting from the initial iterate I. Both the penalty and reduced methods converge to the same final iterate when starting from this initial guess and are able to fit the data equally well.

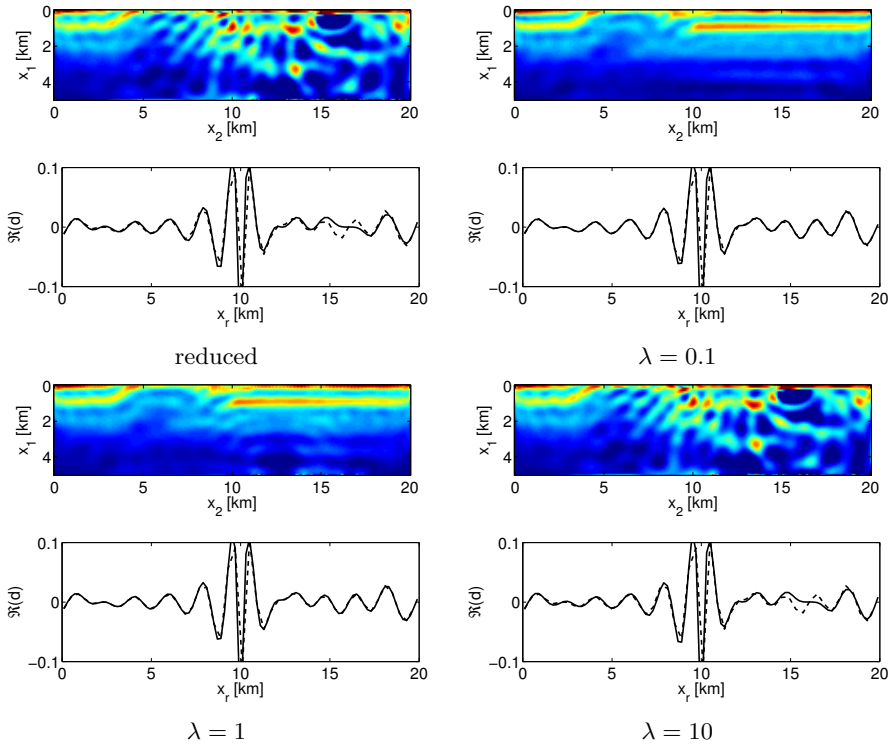


Figure 13. QN reconstructions after 50 iterations and corresponding data for a source in the center, starting from the initial iterate II. For small λ, the penalty method converges to the same final iterate as when starting from initial guess II, showing stability against changes in the initial guess. The reduced method converges to a completely different model, suggesting that the optimization method is stuck in a local minimum. This is confirmed when looking at the data-fit.

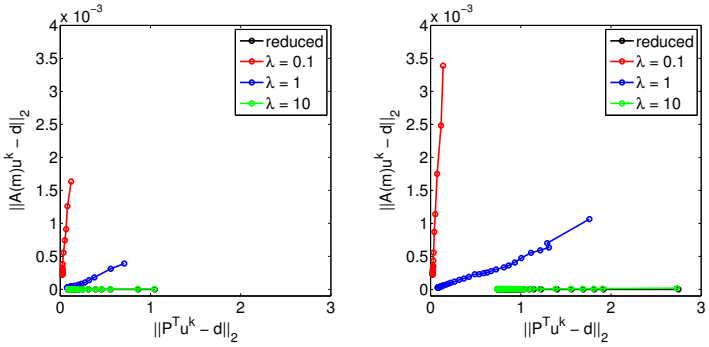


Figure 14. Convergence history in terms of the data-fit and distance to the constraint, starting from initial iterate I (left) and starting from initial iterate II (right). These plots show that, for small λ, the penalty method is able to reduce both the data and PDE misfit to the same level when starting from either initial guess. The reduced method, however, cannot reduce the data misfit to the same level when starting from initial guess II, suggesting that it got stuck in a local minimum.

[1] A Tarantola and A Valette. Generalized nonlinear inverse problems solved using the least squares criterion. *Reviews of Geophysics and Space Physics*, 20(2):129–232, 1982.

[2] Eldad Haber, Uri M. Ascher, and Douglas W. Oldenburg. Inversion of 3D electromagnetic data in frequency and time domain using an inexact all-at-once approach. *Geophysics*, 69(5):1216, 2004.

[3] I Epanomeritakis, V Akçelik, O Ghattas, and J Bielak. A Newton-CG method for large-scale three-dimensional elastic full-waveform seismic inversion. *Inverse Problems*, 24(3):034015, June 2008.

[4] Tristan van Leeuwen and Felix J Herrmann. 3D Frequency-Domain Seismic Inversion with Controlled Sloppiness. *SIAM Journal on Scientific Computing*, 36(5):S192–S217, October 2014.

[5] Gassan S Abdoulaev, Kui Ren, and Andreas H Hielscher. Optical tomography as a PDE-constrained optimization problem. *Inverse Problems*, 21(5):1507–1530, October 2005.

[6] Kun Wang, Thomas Matthews, Fatima Anis, Cuiping Li, Neb Duric, and Mark A. Anastasio. Breast ultrasound computed tomography using waveform inversion with source encoding. In *Proc. SPIE*, page 94190C, 2015.

[7] Eldad Haber, Uri M Ascher, and Doug Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems*, 16(5):1263–1280, October 2000.

[8] Marcus J. Grote, Johannes Huber, Drosos Kourounis, and Olaf Schenk. Inexact Interior-Point Method for PDE-Constrained Nonlinear Optimization. *SIAM Journal on Scientific Computing*, 36:A1251–A1276, 2014.

[9] JE Dennis, Matthias Heinkenschloss, and L.N. Vicente. Trust-region interior-point SQP algorithms for a class of nonlinear programming problems. *SIAM Journal on Control and Optimization*, 36(5):1750–1794, 1998.

[10] M Heinkenschloss and D Ridzal. An inexact trust-region SQP method with applications to PDE-constrained optimization. *Numerical Mathematics and Advanced Applications*, pages 613–620, 2008.

[11] R.G. Richter. Numerical Identification of a Spatially Varying Diffusion Coefficient. *Mathematics of Computation*, 36(154):375–386, 1981.

[12] Biswanath Banerjee, Timothy F Walsh, Wilkins Aquino, and Marc Bonnet. Large Scale Parameter Estimation Problems in Frequency-Domain Elastodynamics Using an Error in Constitutive Equation Functional. *Computer methods in applied mechanics and engineering*, 253:60–72, January 2013.

[13] Tristan van Leeuwen and Felix J Herrmann. Mitigating local minima in full-waveform inversion by expanding the search space. *Geophysical Journal International*, 195(1):661–667, July 2013.

[14] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer Series in Operations Research. Springer, 2006.

[15] R. Herzog. Lectures Notes Algorithms and Preconditioning in PDE-Constrained Optimization. Technical Report July, Chemnitz University of Technology, 2010.

[16] Jonathan Eckstein. Augmented Lagrangian and Alternating Direction Methods for Convex Optimization : A Tutorial and Some Illustrative Computational Results. Technical Report RRR 32-2012, Rutgers University, 2012.

[17] Frank E. Curtis, Hao Jiang, and Daniel P. Robinson. An adaptive augmented Lagrangian method for large-scale constrained optimization. *Mathematical Programming*, 2014.

[18] Aleksandr Y Aravkin and Tristan van Leeuwen. Estimating nuisance parameters in inverse problems. *Inverse Problems*, 28(11):115016, 2012.

[19] Christopher C. Paige and Michael A. Saunders. LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares. *ACM Transactions on Mathematical Software*, 8(1):43–71, March 1982.

[20] David Chin-Lung Fong and Michael Saunders. LSMR: An Iterative Algorithm for Sparse Least-Squares Problems. *SIAM Journal on Scientific Computing*, 33(5):2950–2971, January 2011.

[21] Rafael Bru, José Marín, José Mas, and Miroslav Tma. Preconditioned Iterative Methods for Solving Linear Least Squares Problems. *SIAM Journal on Scientific Computing*, 36(4):A2002–A2022, August 2014.

[22] Å. Björck and T. Elfving. Accelerated projection methods for computing pseudoinverse solutions of systems of linear equations. *BIT*, 19(2):145–163, June 1979.

[23] Dan Gordon and Rachel Gordon. Robust and highly scalable parallel solution of the Helmholtz equation with large wave numbers. *Journal of Computational and Applied Mathematics*, 237(1):182–196, January 2013.

[24] Yair Censor, Paul P. B. Eggermont, and Dan Gordon. Strong underrelaxation in Kaczmarz's method for inconsistent systems. *Numerische Mathematik*, 41(1):83–92, February 1983.

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