A penalty method for PDE-constrained optimization

Tristan van Leeuwen[1] **and Felix J. Herrmann**[2]

[1]Centrum Wiskunde & Informatica. Science Park, Amsterdam, the Netherlands
[2] Dept. of Earth, Ocean and Atmospheric Sciences.
2207 Main Mall, Vancouver, BC Canada V6T 1Z4.

E-mail: tleeuwen@eos.ubc.ca

Abstract. We present a method for solving PDE constrained optimization problems based on a penalty formulation. This method aims to combine advantages of both full-space and reduced methods by exploiting a large search-space (consisting of both control and state variables) while allowing for an efficient implementation that avoids storing and updating the state-variables. This leads to a method that has roughly the same per-iteration complexity as conventional reduced approaches while defining an objective that is less non-linear in the control variable by implicitly relaxing the constraint. We apply the method to a seismic inverse problem where it leads to a particularly efficient implementation when compared to a conventional reduced approach as it avoids the use of adjoint state-variables. Numerical examples illustrate the approach and suggest that the proposed formulation can indeed mitigate some of the well-known problems with local minima in the seismic inverse problem.

1. Introduction

In parameter estimation, the goal is to infer physical parameters (e.g., density, soundspeed or conductivity) from partial measurements of solutions of a PDE that describes the physical process as a function of the paramater of interest (e.g., a wave-equation,). These problems arise in many applications such as geophysics [1, 2], medical imaging [3] and non-destructive testing.

For linear PDEs, the resulting optimzation problem (after discretization) can be written as

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

where $\mathbf{m}$ is the (gridded) parameter of interest, $\mathbf{u}$ is the field and $\mathbf{d}$ are the input data. The measurement process is modelled by taking inner products of the field with the columns of P. The matrix $A(\mathbf{m})\mathbf{u}=\mathbf{q}$ represents the discretized PDE and $\mathbf{q}$ is the source function.

Oftentimes, measurements are made from multiple independent experiments, in which case $\mathbf{u}$ is a block vector containing the fields for different experiments. For some applications, such as seismic inversion, $\mathbf{m}$ may represent up to $\mathcal{O}(10^9)$ unknowns while $\mathbf{u}$ may easily represent $\mathcal{O}(10^{17})$.

1.1. All-at-once approach

A popular approach to solving these constrained problems is based on the corresponding Lagrangian:

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

where ^{T} denotes the (complex-conjugate) transpose. A necessary condition for a solution to the constrained problem (1) is that it is a stationary point of the Lagrangian. Such a stationary point may be found using a Newton-like method by repeatedly solving the KKT system [4]

$$\begin{pmatrix} R & K^T & G^T \\ K & PP^T & A^T \\ G & A & \end{pmatrix} \begin{pmatrix} \delta\mathbf{m} \\ \delta\mathbf{u} \\ \delta\mathbf{v} \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_{\mathbf{m}} \\ \mathcal{L}_{\mathbf{u}} \\ \mathcal{L}_{\mathbf{v}} \end{pmatrix}, \tag{3}$$

where

$$\mathcal{L}_{\mathbf{m}} = G(\mathbf{m},\mathbf{u})^T\mathbf{v}, \tag{4}$$

$$\mathcal{L}_{\mathbf{u}} = A(\mathbf{m})^T\mathbf{v} + P(P^T\mathbf{u}-\mathbf{d}), \tag{5}$$

$$\mathcal{L}_{\mathbf{v}} = A(\mathbf{m})\mathbf{u}-\mathbf{q}, \tag{6}$$

$$\tag{7}$$

and

$$G(\mathbf{m},\mathbf{u}) = \frac{\partial A(\mathbf{m})\mathbf{u}}{\partial\mathbf{m}}, \tag{8}$$

$$K(\mathbf{m},\mathbf{v}) = \frac{\partial A(\mathbf{m})^T\mathbf{v}}{\partial\mathbf{m}}, \tag{9}$$

$$R(\mathbf{m},\mathbf{u},\mathbf{v}) = \frac{\partial G(\mathbf{m},\mathbf{u})^T\mathbf{v}}{\partial\mathbf{m}}. \tag{10}$$

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

An advantages of such an 'all-at-once' approach are that it eliminates the need to solve the PDEs explicitly. However, this approach is often unfeasible for large-scale applications we have in mind because it involves simultaneously updating (and hence storing) all the variables (up to $\mathcal{O}(10^{17})$).

1.2. Reduced approach

For large-scale applications, one usually considers a *reduced* problem

$$\min_{\mathbf{m}} \phi(\mathbf{m}) = \tfrac{1}{2}||P^T\mathbf{u}(\mathbf{m}) - \mathbf{d}||_2^2, \tag{11}$$

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. The gradient and the (Gauss-Newton) Hessian of ϕ are given by

$$\begin{aligned} \nabla\phi(\mathbf{m}) &= G(\mathbf{m},\mathbf{u})^T\mathbf{v}, && (12)\\ \nabla^2\phi(\mathbf{m}) &= G(\mathbf{m},\mathbf{u})^T A(\mathbf{m})^{-T} P P^T A(\mathbf{m})^{-1} G(\mathbf{m},\mathbf{u}) \\ &\quad + K(\mathbf{m},\mathbf{v})^T A(\mathbf{m})^{-T} G(\mathbf{m},\mathbf{u}) + G(\mathbf{m},\mathbf{u})^T A(\mathbf{m})^{-T} K(\mathbf{m},\mathbf{v}) \\ &\quad + R(\mathbf{m},\mathbf{u},\mathbf{v}). && (13) \end{aligned}$$

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

The basic (Gauss-Newton) algorithm for minimizing $\phi(\mathbf{m})$ is given in Algorithm 1. 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 additionally satisfy $\|\mathcal{L}_{\mathbf{m}}(\mathbf{m}^*, \mathbf{u}^*_{\text{red}}, \mathbf{v}^*_{\text{red}})\|_2^2 \leq \epsilon$.

Algorithm 1 Basic Gauss-Newton algorithm for find a stationary point of the Lagrangian via the reduced method

```
Require: initial guess $\mathbf{m}^0$, tolerance $\epsilon$
  $k = 0$
  repeat
    $\mathbf{u}^k_{\text{red}} = A(\mathbf{m}^k)^{-1}\mathbf{q}$
    $\mathbf{v}^k_{\text{red}} = A(\mathbf{m}^k)^{-T}P(\mathbf{d} - P^T\mathbf{u}^k_{\text{red}})$
    $\mathbf{g}^k_{\text{red}} = G(\mathbf{m}^k, \mathbf{u}^k_{\text{red}})^T\mathbf{v}^k_{\text{red}}$
    $H^k_{\text{red}} = G(\mathbf{m}^k, \mathbf{u}^k_{\text{red}})^T A(\mathbf{m}^k)^{-T} P P^T A(\mathbf{m}^k)^{-1} G(\mathbf{m}^k, \mathbf{u}^k_{\text{red}})$
    $\mathbf{m}^{k+1} = \mathbf{m}^k - \alpha^k \left(H^k_{\text{red}}\right)^{-1} \mathbf{g}^k_{\text{red}}$
  until $\|\mathbf{g}^k_{\text{red}}\|_2 \leq \epsilon$
```

The disadvantage of this approach is that it requires the solution of the PDEs at each update, making it computationally very expensive. It also strictly enforces the constraint at each iteration, which might lead to a very nonlinear problem in $\mathbf{m}$. Moreover, the corresponding Hessian is typically a dense matrix that cannot be stored and computing its action involves additional PDE solves. Practical approaches are usually based on Quasi-Newton approximations of the reduced Hessian.

1.3. Contributions and outline

In this paper we present an alternative to the reduced approach which has a roughly equivalent per-iteration complexity in terms of PDE solves and storage but retains some of the characteristics of the all-at-once approach in the sense that it exploits a larger search space by not enforcing the constraints at each iteration.

The approach is based on a *penalty* formulation of the constrained problem, the solution of which coincides with that of the constrained problem (1) for an. appropriate choice of the penalty parameter. Such a reformulations of the constrained problem are well-known, but for the sake of completeness we give a brief overview in section 2. The main contribution of this paper is a solution strategy, which is based on the elimination of the state variable u via a *variational projection* approach as detailed in section 3. The benefit of this approach is that it effectively eliminates these variables from the optizatiom problem and thus greatly reduces the dimensionality of the optimization problem. Due to the special structure of the problems under consideration, this elimination can be done efficiently, leading to a tractable algorith for large-scale problems. Contrary to the conventional *reduced* approach, the resulting algorihtm does *not* enforce the constraint 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 a case-study presented in section 5 provides some numerical evidence.

A detailed description of the proposed algorihtm is given in section 4. 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 seismic inversion using both the penalty and reduced formulations are given in section 5.

Possible extensions and open problems are discussed in section 6 and section 7 gives the conclusions.

1.4. Related work

The proposed method is related to the *equation-error* approach, which is typically used to identify the control variable in diffusion problems given a *complete* measurement of the state: $\mathbf{d} = \mathbf{u}$ by solving $A(\mathbf{m})\mathbf{u} = \mathbf{q}$ for $\mathbf{m}$ [5, 6]. Given *partial* measurements of the state $\mathbf{d} = P^T\mathbf{u}$, the proposed method can be seen as a way of bootstrapping this by first attempting to reconstruct the complete state from the partial measurements.

2. Penalty methods

A constrained optimization problems of the form (1) can be recast as an unconstrained problem by introducing a positive penalty function π as follows

$$\min_{\mathbf{m},\mathbf{u}} \Phi_\lambda = \tfrac{1}{2}||P^T\mathbf{u} - \mathbf{d}||_2^2 + \lambda\pi(\mathbf{A}(\mathbf{m})\mathbf{u} - \mathbf{q}). \tag{14}$$

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.

2.1. Quadratic penalty function

A quadratic penalty function $\pi(\cdot) = \frac{1}{2}||\cdot||_2^2$ leads to a differentiable unconstrained optimization problem (15) whose minimizer $\{\overline{\mathbf{m}}_\lambda, \overline{\mathbf{u}}_\lambda\}$ coincides with the solution of the constrained optimization problem (1) when $\lambda \uparrow \infty$ [7, 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 for large values of λ when there are fewer constraints than variables. For PDE-constrained optimziation problems in inverse problems, there are typically enough constraints to prevent this and we will discuss this in more detail in section 4.

2.2. Exact penalty methods

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* $\lambda \geq \bar{\lambda}$ for some $\bar{\lambda}$ [7, 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 arbitrarily large and this this avoids 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 do not consider exact penalty methods any further in this paper.

3. A reduced penalty method

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

$$\min_{\mathbf{m},\mathbf{u}} \mathcal{P}(\mathbf{m},\mathbf{u}) = \tfrac{1}{2}||P^T\mathbf{u}-\mathbf{d}||_2^2 + \tfrac{1}{2}\lambda||A(\mathbf{m})\mathbf{u}-\mathbf{q}||_2^2. \tag{15}$$

The gradient and Hessian are given by

$$\begin{pmatrix} \mathcal{P}_{\mathbf{m}} \\ \mathcal{P}_{\mathbf{u}} \end{pmatrix} = \begin{pmatrix} \lambda G(\mathbf{m},\mathbf{u})^T\left(A(\mathbf{m})\mathbf{u}-\mathbf{q}\right) \\ P(P^T\mathbf{u}-\mathbf{d}) + \lambda A(\mathbf{m})^T(A(\mathbf{m})\mathbf{u}-\mathbf{d}) \end{pmatrix}, \tag{16}$$

and

$$\nabla^2\mathcal{P} = \begin{pmatrix} \mathcal{P}_{\mathbf{m},\mathbf{m}} & \mathcal{P}_{\mathbf{m},\mathbf{u}} \\ \mathcal{P}_{\mathbf{u},\mathbf{m}} & \mathcal{P}_{\mathbf{u},\mathbf{u}} \end{pmatrix}, \tag{17}$$

where

$$\begin{aligned} \mathcal{P}_{\mathbf{m},\mathbf{m}} &= \lambda(G(\mathbf{m},\mathbf{u})^TG(\mathbf{m},\mathbf{u}) + R(\mathbf{m},\mathbf{u},A(\mathbf{m})\mathbf{u}-\mathbf{q})), && (18)\\ \mathcal{P}_{\mathbf{u},\mathbf{u}} &= PP^T + \lambda A(\mathbf{m})^TA(\mathbf{m}), && (19)\\ \mathcal{P}_{\mathbf{m},\mathbf{u}} &= \lambda(K(\mathbf{m},A(\mathbf{m})\mathbf{u}) + A(\mathbf{m})^TG(\mathbf{m},\mathbf{u})). && (20)\\ & && (21) \end{aligned}$$

Of course, optimization in the full $(\mathbf{m},\mathbf{u})$-space is not feasible for large-scale problems, so we eliminate $\mathbf{u}$ using a *variational projection* approach [8] to define a reduced problem:

$$\min_{\mathbf{m}} \phi_\lambda(\mathbf{m}) = \mathcal{P}(\mathbf{m},\mathbf{u}_\lambda(\mathbf{m})), \tag{22}$$

where $\mathbf{u}_\lambda(\mathbf{m}) = \mathrm{argmin}_{\mathbf{u}}\mathcal{P}(\mathbf{m},\mathbf{u})$. It is readily verified that the gradient and Hessian of ϕ_λ are given by

$$\begin{aligned} \nabla\phi_\lambda(\mathbf{m}) &= \mathcal{P}_{\mathbf{m}}(\mathbf{m},\overline{\mathbf{u}}_\lambda), && (23)\\ \nabla^2\phi_\lambda(\mathbf{m}) &= \mathcal{P}^2_{\mathbf{m}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda) \\ &\quad - \nabla^2_{\mathbf{m},\mathbf{u}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda)\left(\nabla^2_{\mathbf{u}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda)\right)^{-1}\nabla^2_{\mathbf{u},\mathbf{m}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda) && (24) \end{aligned}$$

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

The optimization problem for $\mathbf{u}_\lambda$ has a closed-form solution and the basic Gauss-Newton algorithm for minimizing ϕ_λ is shown in Algorithm 2.

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

```
**Require:** initial guess $\mathbf{m}^0$, penalty parameter $\lambda$, tolerance $\epsilon$
  $k = 0$
  **repeat**
    $\mathbf{u}_\lambda^k = \left(A^TA + \lambda^{-1}PP^T\right)^{-1}\left(A^*\mathbf{q} + \lambda^{-1}P^*\mathbf{d}\right)$
    $\mathbf{v}_\lambda^k = \lambda(A(\mathbf{m}^k)\mathbf{u}_\lambda^k - \mathbf{q})$
    $\mathbf{g}_\lambda^k = G(\mathbf{m}^k, \mathbf{u}_\lambda^k)^*\mathbf{v}_\lambda^k$
    $H_\lambda^k = \lambda G^T\left(I - A\left(A^TA + \lambda^{-1}PP^T\right)^{-1}A^T\right)G$
    $\mathbf{m}^{k+1} = \mathbf{m}^k - \alpha^k\left(H_\lambda^k\right)^{-1}\mathbf{g}_\lambda^k$
  **until** $\|\mathbf{g}_\lambda^k\|_2 \leq \epsilon$
```

Note that the modified system $A^*A+\lambda^{-1}PP^T$ 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 1); the estimate of the field is not only based on the physics and the current model, but also on the data.

Next, we show how the states generated by this algorithm $\mathbf{u}_\lambda^k$ and $\mathbf{v}_\lambda^k$ relate to the states generated by the reduced approach and subsequently if the algorithm successfully terminates the iterates satisfy $\|\nabla\mathcal{L}\|_2^2 \leq \epsilon + \mathcal{O}(\lambda^{-1})$.

Lemma 3.1 *For a fixed* $\mathbf{m}$*, the states* $\mathbf{u}_\lambda$ *and* $\mathbf{v}_\lambda$ *used in the reduced penalty approach (cf. Algorithm 2) are related to the states* $\mathbf{u}_{\mathrm{red}}$ *and* $\mathbf{v}_{\mathrm{red}}$ *used in the reduced approach (cf. Algorithm 1) as follows*

$$\mathbf{u}_\lambda = \mathbf{u}_{\mathrm{red}} + \mathcal{O}(\lambda^{-1}), \tag{25}$$

$$\mathbf{v}_\lambda = \mathbf{v}_{\mathrm{red}} + \mathcal{O}(\lambda^{-1}). \tag{26}$$

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

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

and

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

The former can be re-written as

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

For $\lambda > \sigma_{\max}(P^TA^{-1})$, we may expand the inverse as $(I + \lambda^{-1}B)^{-1} \approx I - \lambda^{-1}B + \lambda^{-2}B^2 + \ldots$ and find that

$$\begin{aligned}\mathbf{u}_\lambda &= A^{-1}\mathbf{q} \\ &\quad + \lambda^{-1}\left(A^TA\right)^{-1}P\left(\mathbf{d} - P^TA^{-1}\mathbf{q}\right) \\ &\quad - \lambda^{-2}\left(A^TA\right)^{-1}PP^T\left(A^TA\right)^{-1}P\mathbf{d} + \mathcal{O}(\lambda^{-3}) \\ &= \mathbf{u}_{\mathrm{red}} + \lambda^{-1}A\mathbf{v}_{\mathrm{red}} + \lambda^{-2}.\end{aligned} \tag{27}$$

We immediately find

$$\mathbf{v}_\lambda = \mathbf{v}_{\mathrm{red}} + \mathcal{O}(\lambda^{-1}). \tag{28}$$

Theorem 3.2 *At each iteration of algorithm 1, the iterates satisfy* $\mathcal{L}_{\mathbf{u}}(\mathbf{m}^k, \mathbf{u}^k_\lambda, \mathbf{v}^k_\lambda) = \mathcal{O}(\lambda^{-1})$ *and* $\mathcal{L}_{\mathbf{v}}(\mathbf{m}^k, \mathbf{u}^k_\lambda, \mathbf{v}^k_\lambda) = 0$ *Moreover, if algorithm 1 successfully terminates with* $\mathbf{m}^*$ *for which* $\|\mathbf{g}^*_\lambda\|_2 \leq \epsilon$ *we have* $\|\mathcal{L}_{\mathbf{m}}(\mathbf{m}^*, \mathbf{u}^*_\lambda, \mathbf{v}^*_\lambda)\|_2 \leq \epsilon 3$*.*

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

$$\begin{aligned} \mathcal{L}_{\mathbf{u}}(\mathbf{m}, \mathbf{u}_\lambda, \mathbf{v}_\lambda) &= A(\mathbf{m})^T \mathbf{v}_\lambda + P(P^T \mathbf{u}_\lambda - \mathbf{d}) \\ &= \lambda A^T (A \mathbf{u}_\lambda - \mathbf{q}) + P(P^T \mathbf{u}_\lambda - \mathbf{d}) = 0. \end{aligned} \tag{29}$$

Using the approximations for $\mathbf{u}_\lambda$ and $\mathbf{v}_\lambda$ for $\lambda > \sigma_{\max}(P^T A)$ presented in Lemma , we find

$$\begin{aligned} \mathcal{L}_{\mathbf{v}}(\mathbf{m}, \mathbf{u}_\lambda, \mathbf{v}_\lambda) &= A(\mathbf{m}) \mathbf{u}_\lambda - \mathbf{q} \\ &= \lambda^{-1} A(\mathbf{m})^{-T} P \left(\mathbf{d} - P^T A(\mathbf{m})^{-1} \mathbf{q} \right) + \mathcal{O}(\lambda^{-2}). \end{aligned} \tag{30}$$

At a stationary point $\mathbf{m}^*$, $\|\mathbf{d} - P^T A(\mathbf{m}^*)^{-1} \mathbf{q}\|_2$ can be interpreted as the noise level. Thus we find

$$\|\mathcal{L}_{\mathbf{v}}(\mathbf{m}^*, \mathbf{u}^*_\lambda, \mathbf{v}^*_\lambda)\|_2 \leq \lambda^{-1} \mu \sigma, \tag{31}$$

where σ is the noise level and μ is the largest singular value of $P^T A(\mathbf{m}^*)^{-1}$.

At a point $\mathbf{m}^*$ for which $\|\nabla \phi_\lambda(\mathbf{m}^*)\|_2 \leq \epsilon$ we immediately find that

$$\|\mathcal{L}_{\mathbf{m}}(\mathbf{m}^*, \mathbf{u}^*_\lambda, \mathbf{v}^*_\lambda)\|_2^2 \leq \epsilon. \tag{32}$$

Finally, we find that

$$\|\nabla \mathcal{L}(\mathbf{m}^*, \mathbf{u}^*_\lambda, \mathbf{v}^*_\lambda)\|_2 \leq \sqrt{\epsilon^2 + \lambda^{-2} \mu^2 \sigma^2}.e \tag{33}$$

■

4. Algorithms

In this section we discuss some practicalities of the implementation of algorithm 2. 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 M independent experiments and $\mathbf{d} = [\mathbf{d}_1; \ldots; \mathbf{d}_M]$ is a block-vector. We partition the states and sources in a simular manner and, since the experiments are independent, system matrix A is block-diagonal matrix with blocks $A_i(\mathbf{m})$. The matrix P has rank L.

4.1. Solving the augmented PDE

Due to the block structure of the problem, the linear systems can be solved independently. We assume for the moment that the state vector for a single experiment can be held in memory.

We can obtain the states by solving the following inconsistent overdetermined system

$$\left(\begin{array}{c} A_i(\mathbf{m}) \\ \lambda^{-1/2} P \end{array} \right) \mathbf{u}_i \approx \left(\begin{array}{c} \mathbf{q}_i \\ \lambda^{-1/2} \mathbf{d}_i \end{array} \right), \tag{34}$$

in a least-squares sense. If both A_i and P are sparse, we can efficiently solve this via a QR factorization or via a Cholesky factorization of the corresponding Normal equations. For simple measurements, PP^T is a diagonal matrix with ones and zeros and thus the augmented system $A^T A + \lambda^{-1} PP^T$ has the same sparsity pattern as the original system.

While we can make use of factorization techniques for small-scale applications, industry-scale applications will typically require (preconditioned) iterative solution of such systems. This allows for on-the-fly generation of the system-matrix, but still requires the state vector to fit in memory. It seems most attractive to work on the overdetermined system directly, rather than form the Normal equations. Obvious candidates are LSQR or LSMR [9, 10]. Another promising candidate is a generic accelerated row-projected method described by [11, 12] which proved successful in seismic applications and can be easily extended to deal with overdetermined systems [13].

The augmented system $A^TA + \lambda^{-1}PP^*$ is a rank L modification of the original system A^TA. It follows from [14, Thm 8.1.8] that the eigenvalues are related as

$$\mu_n(A^TA + \lambda^{-1}PP^T) = \mu_n(A^TA) + a_n\lambda^{-1},$$

where $\mu_n(B)$ denotes the n-th eigenvalue of B and $\sum_n a_n = L$.

When the state vector does not fit in memory, it is not so obvious how the overdetermined system should be solved. An example is explicit time-stepping, where the system matrix exhibits a lower-triangular block-structure.

4.2. Gradient and Hessian

Given these solutions $\mathbf{u}_i$, the gradient, $\mathbf{g}_\lambda$ and Gauss-Newton Hessian H_λ of ϕ_λ are given by (cf eq. 23-24)

$$\mathbf{g}_\lambda = \lambda\sum_{i=1}^{M} G_i^T\left(A_i\mathbf{u}_i - \mathbf{q}_i\right), \tag{35}$$

$$H_\lambda = \lambda\sum_{i=1}^{M} G_i^T\left(I - A_i\left(A_i^TA_i + \lambda^{-1}PP^T\right)^{-1}A_i^T\right)G_i, \tag{36}$$

where $G_i = G(\mathbf{m}, \mathbf{u}_i)$. We can compute the inverse of $\left(A_i^TA_i + \lambda^{-1}PP^T\right)$ 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. The Hessian is never formed explicitly, but its action is computed as required.

4.3. Complexity estimates

Assuming we can solve the overdetermined system (34) 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 1.

5. Case studies

The following experiments are done in Matlab, using `slash` to solve the PDEs. 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 K. Finally, we add a regularization term $\frac{\alpha}{2}\|D\mathbf{m}\|_2^2$, where D is the first-order finite-difference matrix, to both the reduced and penalty approaches. We measure the cost of the inversion by counting the number of PDE solves as outlined in table 1.

The code used to perform the experiments is available from `github`.

5.1. 1D DC resistivity

We consider the PDE

$$\partial_t u(t,x) = \partial_x \left(m(x)\partial_x\right) u(t,x), \tag{37}$$

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

$$A(\mathbf{m}) = \imath\omega I + D^T \mathsf{diag}(\mathbf{m}) D, \tag{38}$$

where ω is the angular frequency and D is the first-order finite-difference matrix. The Jacobian is given by $G(\mathbf{m},\mathbf{u}) = D^T\mathsf{diag}(D\mathbf{u})$.

The domain is discretized using $N = 51$ points and we let $P = Q = [\mathbf{e}_2, \mathbf{e}_{N-1}] * (N-1)$. For the inversion we use a GN method with $\epsilon = 10^{-9}$, $\eta = 10^{-3}$ and set the regularization parameter $\alpha = 10^{-6}$. The initial parameters are $\mathbf{m}^0 = \mathbf{1}$ and we set λ relative to the largest singular value of $PA(\mathbf{m}^0)^{-1}$. The results are shown in figure 1. The convergence plot, figure 1 (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 1 (b). The actual costs of the inversion are listed in table 2.

5.2. 2D seismic waveform inversion

Consider the 2D scalar wave-equation

$$m\partial_t^2 u(t,x) = \nabla^2 u(t,x), \tag{39}$$

on $x \in \Omega \subseteq \mathbb{R}^2$ with radiation boundary conditions $\sqrt{m}\partial_t - \partial_n u = 0$ on $\partial\Omega$. Discretization in the temporal Fourier domain leads to a scalar Helmholtz equation

$$A(\mathbf{m}) = \mathsf{diag}(\mathbf{s}) - D^T D, \tag{40}$$

where $D = [D_1; D_2]$ is the a finite-difference discretitation of ∇ and $s_i = \omega^2 m_i$ in the interior and $s_i = \imath\omega\sqrt{m_i}$ on the boundary. The Jacobian is given by

$$G(\mathbf{m},\mathbf{u}) = \mathsf{diag}(\mathbf{s}')\mathsf{diag}(\mathbf{u}), \tag{41}$$

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

5.2.1. Toy example. The domain $\Omega = [0,1000] \times [0,1000]$ is discretized using $N = 51 \times 51$ points. We take $P = Q$ and the points sampled by P are shown in figure 4 (a). The ground truth $\mathbf{m}^*$ is shown in figure 4 (b). For the inversion we use a GN method with $\epsilon = 10^{-9}$, $\eta = 10^{-1}$ and set the regularization parameter $\alpha = 10^1$. The initial parameters are $\mathbf{m}^0 = \frac{1}{4}\mathbf{1}$ and we set λ relative to the largest singular value of $PA(\mathbf{m}^0)^{-1}$.

First, we investigate the sensitivity of the misfit functions ϕ and ϕ_λ by plotting $\phi(\mathbf{m}^* + \delta_1\mathbf{v} + \delta_2\mathbf{w})$ and $\phi_\lambda(\mathbf{m}^* + \delta_1\mathbf{v}_\lambda + \delta_2\mathbf{w}_\lambda)$ as a function of (δ_1,δ_2). We take $\mathbf{v},\mathbf{w}$ and $\mathbf{v}_\lambda,\mathbf{w}_\lambda$ to be the first two dominant eigenvector of the GN-Hessian of ϕ and ϕ_λ respectively. These are shown in figure 2. The eigenvectors for both the reduced and penalty approach exhibit a similar behaviour; the first is basically constant while the second is a first order diagonally oscillating model. The misfit as a function of (δ_1, δ_2) is shown in figure 3. We see a radically different behaviour for the reduced and penalty methods. The first exhbits strong non-linearity and some local minima while for $\lambda = 0.1$ the misfit is much better behaved. For larger values λ the penalty misfit starts to behave more like the reduced misfit.

The results of the inversion are shown in figure 5. The convergence plot, figure 5 (top), 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 **??** (bottom). The actual costs of the inversion are listed in table 3.

5.2.2. Example 2

6. Discussion

This paper lays out the basics of an efficient implementation of the penalty method for PDE-constrained optimization.

7. Conclusions

We have presented a new method for PDE-constrained optimization based on a penalty formulation of the constrained problem. The method relies on solving for the state variables from an augmented system that is comprised of the original discretized PDE and the measurements. The resulting estimates of the state variables can be used to directly estimate the control variable from the PDE via an equation-error approach. The main benefits of this method are: *i)* The state variables for each experiment can be obtained independently and do not have to be stored or updated as part of an iterative optimization procedure, *ii)* the penalty formulation leads to a less non-linear formulation than the reduced approach where the PDE-constraint is eliminated explicitly, and *iii)* the gradient of the objective with respect to the control variable can be computed directly from the state variables, without the need to solve adjoint PDEs. We illustrate the approach on a non-linear seismic inverse problem, showing that the reduced non-linearity leads to significantly better results than the reduced approach at roughly half the computational costs (due to the fact that there is no adjoint equation to solve). Moreover, the penalty approach succesfully mitigates some of the the issues with local minima making the procedure less sensitive to the initial model.

Acknowledgments

The authors wish to thank Dan and Rachel Gordon for valuable discussions on the CARP-CG method. 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.

	# PDE's	Storage	Gauss-Newton update
penalty	M	N	solve matrix-free linear system in N unknowns, requires M (overdetermined) PDE solves per mat-vec
reduced	$2M$	$2N$	solve matrix-free linear system in N unknowns, requires $2M$ PDE solves per mat-vec
all-at-once	0	$(M+1)\times N$	solve sparse symmetric, possibly indefinite system in $(M+1)\times N3$ unknowns

Table 1. Leading order computation and storage costs per iteration of different methods; M denotes the number of experiments and N denotes the number of gridpoints. for large-scale 3D seismic inverse problems we typically have $M = \mathcal{O}(10^6)$ and $N = \mathcal{O}(10^9)$.

	reduced	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
iterations	6	5	5	6
PDE solves	222	101	95	126

Table 2. Costs of the 1D DC resistivity inversion.

	reduced	$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
iterations	10	8	9	10
PDE solves	302	81	137	148

Table 3. Costs of the 2D seismic inversion.

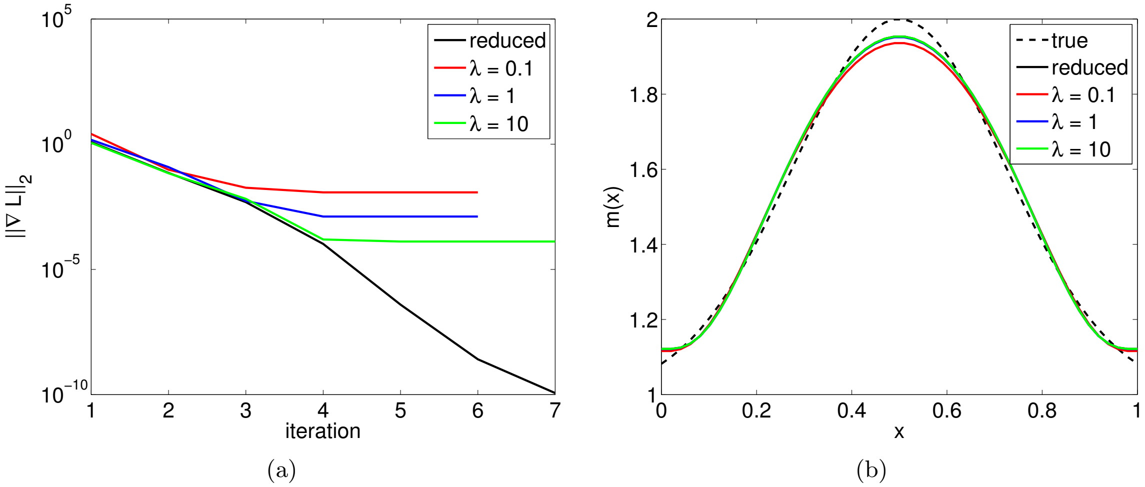


Figure 1. Solutions and convergence history for 1D resisivity problem.

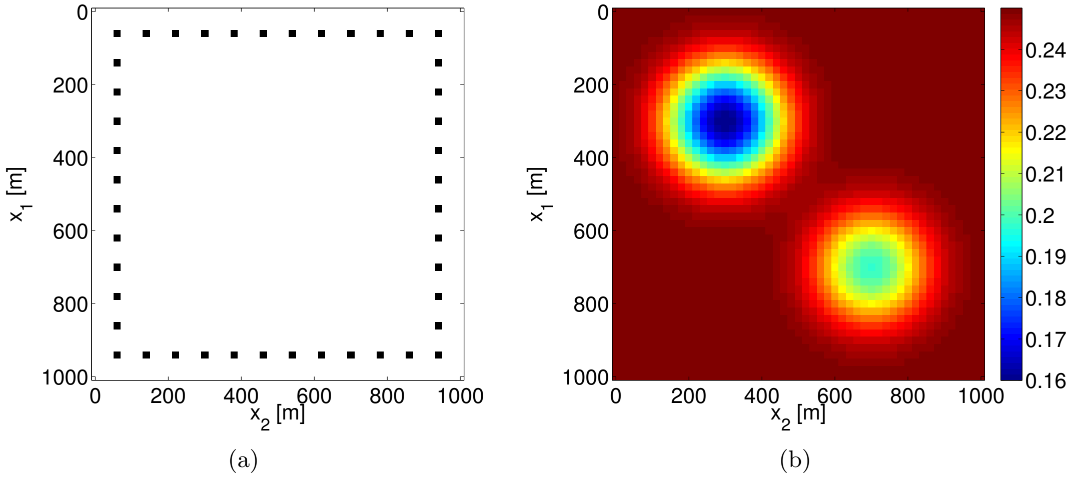
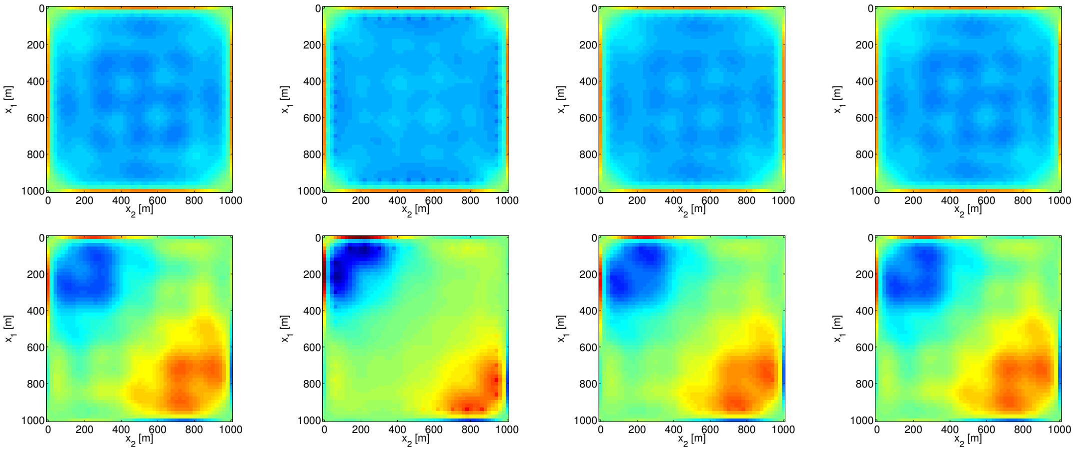
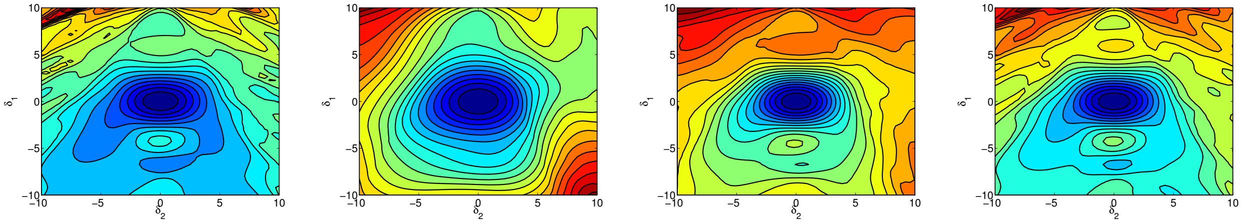


Figure 2. Sampling operator, ground truth and convergence history.



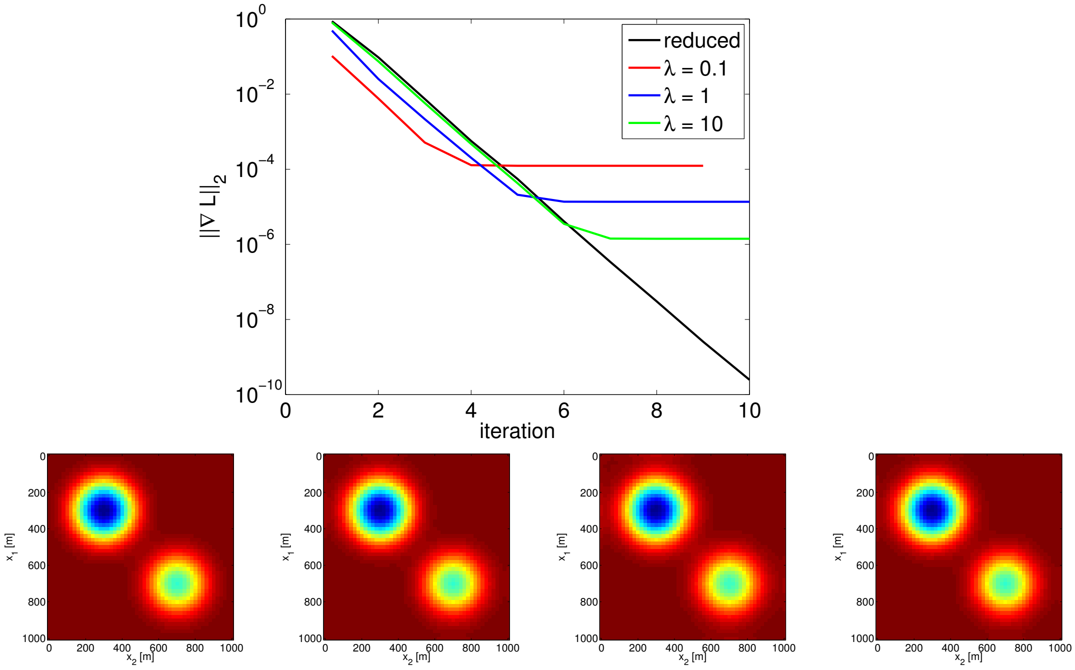
reduced $\lambda = 0.1$ $\lambda = 1$ $\lambda = 10$

Figure 3. Dominant eigenvectors of the GN Hessian.



reduced $\lambda = 0.1$ $\lambda = 1$ $\lambda = 10$

Figure 4. Misfit in the direction of the dominant eigenvector of the Hessian.



reduced

$\lambda = 0.1$

$\lambda = 1$

$\lambda = 10$

Figure 5. Convergence history and reconstructions.

[1] 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.

[2] 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.

[3] 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.

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

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

[6] 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.

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

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

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

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