# 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 [**?**, **?**], medical imaging [**?**] and non-destructive testing.

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

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

where $\mathbf{m}$ is the (gridded) parameter of interest, $\mathbf{u}$ is the field and $\mathbf{d}$ are the input data. The matrix $P$ restricts the predicted field to the measurement locations. 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:

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

where $^*$ denotes the complex-conjugate transpose. A necessary condition for a solution to the constrained problem (**??**) 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 [**?**]

$$\begin{pmatrix} \nabla^2_{\mathbf{m}}\mathcal{L} & \nabla^2_{\mathbf{m},\mathbf{u}}\mathcal{L} & \nabla^2_{\mathbf{m},\mathbf{v}}\mathcal{L} \\ \nabla^2_{\mathbf{u},\mathbf{m}}\mathcal{L} & \nabla^2_{\mathbf{u}}\mathcal{L} & \nabla^2_{\mathbf{u},\mathbf{v}}\mathcal{L} \\ \nabla^2_{\mathbf{v},\mathbf{m}}\mathcal{L} & \nabla^2_{\mathbf{v},\mathbf{u}}\mathcal{L} & \nabla^2_{\mathbf{v}}\mathcal{L} \end{pmatrix} \begin{pmatrix} \delta\mathbf{m} \\ \delta\mathbf{u} \\ \delta\mathbf{v} \end{pmatrix} = - \begin{pmatrix} \nabla_{\mathbf{m}}\mathcal{L} \\ \nabla_{\mathbf{u}}\mathcal{L} \\ \nabla_{\mathbf{v}}\mathcal{L} \end{pmatrix}, \tag{3}$$

and updating $\mathbf{m} := \mathbf{m} + \delta\mathbf{m}$, etc.

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

$$\min_{\mathbf{m}} \phi(\mathbf{m}) = \tfrac{1}{2}||P\overline{\mathbf{u}}(\mathbf{m})-\mathbf{d}||_2^2, \tag{4}$$

where $\overline{\mathbf{u}}(\mathbf{m}) = A(\mathbf{m})^{-1}\mathbf{q}$. 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 $\phi$ are given by

$$\nabla\phi(\mathbf{m}) = G(\mathbf{m},\overline{\mathbf{u}})^*\overline{\mathbf{v}}, \tag{5}$$

$$\begin{aligned}\nabla^2\phi(\mathbf{m}) = {} & G(\mathbf{m},\overline{\mathbf{u}})^*A(\mathbf{m})^{-*}P^*PA(\mathbf{m})^{-1}G(\mathbf{m},\overline{\mathbf{u}}) \\ & + K(\mathbf{m},\overline{\mathbf{v}})^*A(\mathbf{m})^{-*}G(\mathbf{m},\overline{\mathbf{u}}) + G(\mathbf{m},\overline{\mathbf{u}})^*A(\mathbf{m})^{-*}K(\mathbf{m},\overline{\mathbf{v}}) \\ & + R(\mathbf{m},\overline{\mathbf{u}},\overline{\mathbf{v}}).\end{aligned} \tag{6}$$

where $\overline{\mathbf{v}} = A(\mathbf{m})^{-*}P^*\left(\mathbf{d} - P\overline{\mathbf{u}}\right)$ and

$$G(\mathbf{m},\mathbf{u}) = \frac{\partial A(\mathbf{m})\mathbf{u}}{\partial \mathbf{m}}, \tag{7}$$

$$K(\mathbf{m},\mathbf{v}) = \frac{\partial A(\mathbf{m})^*\mathbf{v}}{\partial \mathbf{m}}, \tag{8}$$

$$R(\mathbf{m},\mathbf{u},\mathbf{v}) = \frac{\partial G(\mathbf{m},\mathbf{u})^*\mathbf{v}}{\partial \mathbf{m}}. \tag{9}$$

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 (**??**) 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 **??**. 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 **??**. The benefit of this approach is that it effectively eliminates these variables from the optiziatiom 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 $c(m,u)$ at each iteration and arguably leads to a less non-linear problem in $m$. It is outside the scope of the current paper to give a rigorous prove of this statement, but a case-study on a simple toy problem presented in section **??** gives an intuitive justification.

A detailed description of the proposed algorihtm is given in section **??**. 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 **??**.

Possible extensions and open problems are discussed in section **??** and section **??** 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 $\mathbf{m}$ [**?**, **?**]. Given *partial* measurements of the state $\mathbf{d} = P\mathbf{u}$, 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 (**??**) can be recast as an unconstrained problem by introducing a positive penalty function $\pi$ as follows

$$\min_{\mathbf{m},\mathbf{u}} \Phi_\lambda = \tfrac{1}{2}||P\mathbf{u} - \mathbf{d}||_2^2 + \lambda\pi(\mathbf{A}(\mathbf{m})\mathbf{u} - \mathbf{q}). \tag{10}$$

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 $\lambda$ 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 (**??**) whose minimizer $\{\overline{\mathbf{m}}_\lambda, \overline{\mathbf{u}}_\lambda\}$ coincides with the solution of the constrained optimization problem (**??**) when $\lambda \uparrow \infty$ [**?**, Thm. 17.1]. Practical algorithms rely on repeatedly solving the unconstrained problem for increasing values of $\lambda$. A common concern with this approach is that the Hessian may become increasingly ill-conditioned for large values of $\lambda$ 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 **??**.

### *2.2. Exact penalty methods*

For non-smooth penalty functions, such as $\pi(\cdot) = ||\cdot||_1$, the minimizer of $\phi_\lambda$ is a solution of the constrained problem for *any* $\lambda \geq \bar{\lambda}$ for some $\bar{\lambda}$ [**?**, Thm. 17.3]. In practice, a continuation strategy is used to find a suitable value for $\lambda$. An advantage of this approach is that $\lambda$ does not become arbritarily 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. Variational projection

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

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

The gradient and Hessian are given by

$$\nabla\phi_\lambda = \begin{pmatrix} \nabla_{\mathbf{m}}\phi_\lambda \\ \nabla_{\mathbf{u}}\phi_\lambda \end{pmatrix} = \begin{pmatrix} \lambda G(\mathbf{m},\mathbf{u})^* \left(A(\mathbf{m})\mathbf{u} - \mathbf{q}\right) \\ P^*(P\mathbf{u} - \mathbf{d}) + \lambda A(\mathbf{m})^*(A(\mathbf{m})\mathbf{u} - \mathbf{q}) \end{pmatrix}, \tag{12}$$

and

$$\nabla^2\Phi_\lambda = \begin{pmatrix} \nabla^2_{\mathbf{m}}\Phi_\lambda & \nabla^2_{\mathbf{m},\mathbf{u}}\Phi_\lambda \\ \nabla^2_{\mathbf{u},\mathbf{m}}\Phi_\lambda & \nabla^2_{\mathbf{u}}\Phi_\lambda \end{pmatrix}, \tag{13}$$

where

$$\nabla^2_{\mathbf{m}}\Phi_\lambda(\mathbf{m},\mathbf{u}) = \lambda(G(\mathbf{m},\mathbf{u})^*G(\mathbf{m},\mathbf{u}) + R(\mathbf{m},\mathbf{u},A(\mathbf{m})\mathbf{u}-\mathbf{q})), \tag{14}$$

$$\nabla^2_{\mathbf{u}}\Phi_\lambda(\mathbf{m},\mathbf{u}) = P^*P + \lambda A(\mathbf{m})^*A(\mathbf{m}), \tag{15}$$

$$\nabla^2_{\mathbf{m},\mathbf{u}}\Phi_\lambda(\mathbf{m},\mathbf{u}) = \lambda(K(\mathbf{m},A(\mathbf{m})\mathbf{u}) + A(\mathbf{m})^*G(\mathbf{m},\mathbf{u})). \tag{16}$$

$$\tag{17}$$

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

$$\min_{\mathbf{m}} \phi_\lambda(\mathbf{m}) = \Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}(\mathbf{m})), \tag{18}$$

where $\overline{\mathbf{u}}_\lambda(\mathbf{m}) = \text{argmin}_{\mathbf{u}}\Phi_\lambda(\mathbf{m},\mathbf{u})$. We interpret the argmin in a loose sence, in that $\overline{\mathbf{u}}_\lambda$ only needs to be a local minimizer.

It is readily verified that the gradient and Hessian of $\phi_\lambda$ are given by

$$\nabla\phi_\lambda(\mathbf{m}) = \nabla_{\mathbf{m}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda), \tag{19}$$

$$\nabla^2\phi_\lambda(\mathbf{m}) = \nabla^2_{\mathbf{m}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda) - \nabla^2_{\mathbf{m},\mathbf{u}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda)\left(\nabla^2_{\mathbf{u}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda)\right)^{-1}\nabla^2_{\mathbf{u},\mathbf{m}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda) \tag{20}$$

Note that $\nabla^2\phi_\lambda(\mathbf{m})$ is the Schur complement of $\nabla^2\Phi_\lambda$.

In section **??** we will use these expressions to devise an algorithm to solve the unconstrained problem (**??**), but first we will show that minimizing $\phi_\lambda$ is equivalent to minimizing $\Phi_\lambda$

**Theorem 3.1** *A local minimizer $\overline{\mathbf{m}}_\lambda$ of $\phi_\lambda$ together with the corresponding $\overline{\mathbf{u}}_\lambda = \text{argmin}_{\mathbf{u}}\Phi_\lambda(\overline{\mathbf{m}},\mathbf{u})$, are also a minimizer of $\Phi_\lambda$ and vice versa.*

**Proof** Since $\overline{\mathbf{u}}_\lambda$ is assumed to be a local minimizer of $\Phi_\lambda(\mathbf{m},\cdot)$, it satisfies the following optimality conditions:

$$\nabla_{\mathbf{u}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda) = 0,$$

and

$$\nabla^2_{\mathbf{u}}\Phi_\lambda(\mathbf{m},\overline{\mathbf{u}}_\lambda) \succ 0.$$

Likewise, since $\overline{\mathbf{m}}_\lambda$ is a local minimizer of $\phi_\lambda$, it satisfies

$$\nabla_{\mathbf{m}}\phi_\lambda(\overline{\mathbf{m}}_\lambda) = 0,$$

and

$$\nabla^2_{\mathbf{m}}\phi_\lambda(\overline{\mathbf{m}}_\lambda) \succ 0.$$

We have to show that the pair $(\overline{\mathbf{m}}_\lambda,\overline{\mathbf{u}}_\lambda)$ satisfy the first and second order optimilaty conditions of $\Phi_\lambda$. From the expression for the gradient of $\phi_\lambda$ (**??**) it follows immediately that $\nabla\phi_\lambda = 0 \iff \nabla\Phi_\lambda = 0$. Since $\nabla^2\phi_\lambda$ is the Schur complement of $\nabla^2\Phi_\lambda$ it follows that $\nabla^2\phi \succ 0, \nabla^2_{\mathbf{u}}\Phi \succ 0 \iff \nabla^2\Phi_\lambda \succ 0$ (cf. [**?**, prop. 14.1]). ∎

Finally, we show that minimizing $\phi_\lambda$ is equivalent to minimizing $\phi$ as $\lambda \uparrow \infty$ via the following Theorem.

**Theorem 3.2** *A (local) minimizer of $\phi_\lambda$ satisfies the first and second order optimality conditions of $\phi$ up to $\mathcal{O}(\lambda^{-1})$.*

**Proof** The optimzation problem for $\overline{\mathbf{u}}_\lambda$ has a closed-form solution

$$\overline{\mathbf{u}}_\lambda = \left(P^*P + \lambda A^*A\right)^{-1}\left(P^*\mathbf{d} + \lambda A^*\mathbf{q}\right), \tag{21}$$

which can be re-written as

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

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

$$\begin{aligned}\overline{\mathbf{u}}_\lambda &= A(\mathbf{m})^{-1}\mathbf{q} \\ &\quad + \lambda^{-1}\left(A^*A\right)^{-1}P^*\left(\mathbf{d} - PA^{-1}\mathbf{q}\right) \\ &\quad - \lambda^{-2}\left(A^*A\right)^{-1}P^*P\left(A^*A\right)^{-1}P^*\mathbf{d} + \mathcal{O}(\lambda^{-3}), \qquad (22)\\ &= \overline{\mathbf{u}} + \mathcal{O}(\lambda^{-1}). \qquad (23)\end{aligned}$$

If we substitute this in the expression for the gradient of $\phi_\lambda$ we get

$$\begin{aligned}\nabla\phi_\lambda(\mathbf{m}) &= \lambda G(\mathbf{m}, \overline{\mathbf{u}}_\lambda)^*\left(A(\mathbf{m})\mathbf{u}_\lambda - \mathbf{q}\right) \\ &= G(\mathbf{m}, \overline{\mathbf{u}})^*\overline{\mathbf{v}} + \lambda^{-1}G(\mathbf{m}, A^{-1}\overline{\mathbf{v}})^*\overline{\mathbf{v}} + \mathcal{O}(\lambda^{-2}) \\ &= \nabla\phi(\mathbf{m}) + \mathcal{O}(\lambda^{-1}). \qquad (24)\end{aligned}$$

For the Hessian we find

$$\nabla^2\phi_\lambda(\mathbf{m}) = \nabla^2\phi(\mathbf{m}) + \mathcal{O}(\lambda^{-1}). \tag{25}$$

Next, we need to show that

By assumption, $\overline{\mathbf{m}}_\lambda$ is a local minimizer of $\phi_\lambda$ so that $\nabla\phi(\overline{\mathbf{m}}_\lambda) = 0$ and $\nabla^2\phi(\overline{\mathbf{m}}_\lambda) \succ 0$. If follows that

$$\begin{aligned}\nabla\phi(\overline{\mathbf{m}}_\lambda) &= \mathcal{O}(\lambda^{-1}). \\ \nabla^2\phi(\overline{\mathbf{m}}_\lambda) &= \mathcal{O}(\lambda^{-1})\end{aligned}$$

## 4. Analysis of a toy problem

Before describing algorithms to minimize $\phi_{\text{pen}}$, we will illustrate the potential benefits of the penalty approach over the conventional reduced approach with the help of a toy problem with

$$A(\mathbf{m}) = \begin{pmatrix} \frac{1}{2} & \frac{1}{4} \\ \frac{1}{4} & 1 \end{pmatrix}, \quad q = \begin{pmatrix} 1\frac{3}{4} \\ 2\frac{1}{4} \end{pmatrix}.$$

The solution in this case is $\overline{m} = (1,1)$ and $\overline{u} = (1,1)$. We use a Gauss-Newton method to solve both (**??**) and (**??**), starting from $m = (2,2)$. We use $\lambda = 0.1$ for the penalty approach. Figure **??** (a) shows the solution paths and (c) shows the convergence in terms of $||r||_2$ and $||c||_2$ for both the reduced (black) and penalty approaches (red). The penalty approach gets very close to the optimal solution in one iteration while the reduced approach takes a detour because it is forced to satisfy the constraint $c(m,u) = 0$ at each iteration. Another perspective is offered by plotting the objective functions corresponding to the reduced and penalty approaches as a function of $m$. Figure **??** (c) shows the reduced objective (**??**) and (d) shows the penalty objective (**??**). The reduced objective in this case is quite non-linear while the penalty objective is quadratic. These plots illustrate that the penalty formulation can indeed lead to an objective function that is much better behaved.

## 5. Algorihtms

In this section we develop a pratical algorithm for the minimization of $\phi_\lambda$ based on the ideas discussed in section **??**. We slightly elaborate the notation by explicitly revealing the multi-experiment structure of the problem; The matrix $A$ is a block-diagonal matrix with blocks $A_i(\mathbf{m})$ for $i = 1 \dots M$, $\mathbf{u} = [\mathbf{u}_1; \dots; \mathbf{u}_M]$, etc. Of course, $\mathbf{m}$ does not depend on the experiment index.

The basic structure of

### *5.1. Computing the gradient and Hessian*

Evaluation of $\phi_\lambda$ and its derivatives requires the minimization of $\Phi_\lambda$ w.r.t. $\mathbf{u}$. Because of the block-structure of the problem and the linearity in $\mathbf{u}$ this can be achieved by solving the following overdetermined system for each $i$

$$\begin{pmatrix} \sqrt{\lambda} A_i(\mathbf{m}) \\ P \end{pmatrix} \mathbf{u}_i = \begin{pmatrix} \sqrt{\lambda} \mathbf{q}_i \\ \mathbf{d}_i \end{pmatrix}. \tag{26}$$

Since both $A_i$ and $P$ are sparse, we can efficiently solve this via direct (QR-) factorization or using an iterative method like LSQR or LSMR [**?**, **?**]. Given these solutions $\{\mathbf{u}_i\}$, the gradient and Gauss-Newton Hessian of $\phi_\lambda$ are given by (cf eq. **??**-**??**)

$$\nabla \phi_\lambda = \sum_i \lambda G_i^* \left( A_i \overline{\mathbf{u}}_i - \mathbf{q}_i \right), \tag{27}$$

$$\begin{aligned} \nabla^2 \phi_\lambda &= \sum_i \lambda G_i^* G_i - \lambda^2 G_i^* A_i \left( P^* P + \lambda A_i^* A_i \right)^{-1} A_i^* G_i \\ &= \lambda \sum_i G_i^* G_i - G_i^* \left( I + \lambda^{-1} A_i^{-*} P^* P A_i^{-1} \right)^{-1} G_i, \end{aligned} \tag{28}$$

where $G_i = \frac{\partial A_i(\mathbf{m}) \overline{\mathbf{u}}_i}{\partial \mathbf{m}}$, which is typically a sparse diagonally dominant matrix.

### *5.2. Contination of $\lambda$*

### *5.3. Complexity estimates*

Assuming we can solve equations (**??**) and (**??**) equally efficient, the penalty-based method requires a factor of 2 less computation and storage. Note, however, that for the penalty-based formulation we get an (approximate) Newton method while the reduced method only uses a Quasi-Newton approach. A Gauss-Newton method for the reduced approach would require far more PDE-solves and is not considered here. A summary of the leading order computational costs of the penalty, reduced and all-at-once approaches is given in table **??**.

## 6. Numerical experiments

For the examples, we discretize the 2D Helmholtz operator using a 5-point finite-difference stencil with absorbing boundary conditions. The Quasi-Netwon algorithm **??** is augmented with a simple back-tracking linesearch to ensure descent.

## 7. Discussion

This paper lays out the basics of an efficient implementation of the penalty method for PDE-constrained optimization. A few remaining issues and possible extensions are listed below.

**Solving large, sparse, mildly overdetermined systems** A key step in the penalty method is the solution of the augmented PDE (**??**). While we can make use of factorization techniques for small-scale applications, industry-scale applications will typically require (preconditioned) iterative solution of such systems. A promising candidate is a generic accelerated row-projected method described by [**?**, **?**] which proved successful in seismic applications and can be easily extended to deal with overdetermined systems [**?**].

**Time-domain formulation** We have described application of the penalty method to a formulation in the frequency domain, in which case we can hope to store a complete wavefield $\mathbf{u}_{kl}$ for one source and frequency. In a time-domain formulation, the PDE (after spatial discretization) can be written as

$$M(\mathbf{m})\mathbf{u}'(t) + S\mathbf{u}(t) = \mathbf{q}(t),$$

which can be solved via some form of time-stepping. The augmented wave-equation involves an extra equation of the form $P\mathbf{u}(t) = \mathbf{d}(t)$. While we can in principle form a large overdetermined system for the wavefield at all timesteps, this is not feasible and a suitable time-stepping strategy will have to be developed to solve the augmented PDE without storing the wavefields for all timesteps.

**Other PDE-constrained optimization problems** The penalty formulation can be applied to any PDE-constrained optimization problem. We can only expect an efficient implementation along the lines described in this paper, however, if the PDE is linear in the state variable. In particular, application to multi-parameter seismic inversion (e.g., visco-acoustic, variable density, visco-elastic) is straightfoward.

**Dimensionality reduction** The computational cost may be reduced by using recently proposed dimensionality reduction techniques that essentially reduce the number of right-hand-sides of the PDE by random subselection or aggregration [**?**, **?**, **?**]. These techniques can be directly included in the penalty formulation by introducing new sources and their corresponding data;

$$\widetilde{\mathbf{q}}_{kl'} = \sum_l w_{ll'}\mathbf{q}_{kl}, \quad \widetilde{\mathbf{d}}_{kl'} = \sum_l w_{ll'}\mathbf{d}_{kl},$$

where $l' = [1, 2, \ldots, N_s']$ ($N_s' << N_s$) and $w_{l'l}$ are suitably chosen random weights.

**Regularization** Regularization penalties on the model $\mathbf{m}$ can be included in a trivial manner. Also, regularization techniques on the (Gauss-Newton) subproblems can be included. This includes for example $\ell_1$ regularization as proposed by [**?**] and $\ell_2$ regularization as is used in trust-region methods.

**Penalties** Penalties other than the $\ell_2$ norm on either the data-misfit or the constraint can be included in the formulation (**??**). However, this will most likely prevent us from solving for the wavefields efficiently via system of equations (**??**). Still, most alternative penalties may be approximated by a weighted $\ell_2$ norm, in which case a system like (**??**) *can* be formed and solved via an iteratively re-weighted least-squares approach [**?**].

**Nuisance parameters** Formulations of the inverse problem may include additional parameters that need to be estimated. An example of this are the source-weights in the seismic applications. The penalty objective, in this case becomes

$$\phi_\lambda(\mathbf{m},\mathbf{u},\theta) = \tfrac{1}{2}\sum_{k,l} ||\theta_{kl} P\mathbf{u}_{kl} - \mathbf{d}_{kl}||_2^2 + \lambda^2 ||\theta_{kl} A_k(\mathbf{m})\mathbf{u}_{kl} - \mathbf{q}_{kl}||_2^2,$$

The wavefield may be solved first from (**??**) and subsequent minimization over $\theta$ for the given wavefields is trivial as a closed-form solution is available:

$$\theta_{kl} = \frac{\mathbf{u}_{kl}^* \left(P^*\mathbf{d}_{kl} + \lambda^2 A_k^*\mathbf{q}_{kl}\right)}{||P\mathbf{u}_{kl}||_2^2 + \lambda^2 ||A_k\mathbf{u}_{kl}||_2^2}.$$

This approach can be generalized to more complicated situations in which a closed-form solution is not available [**?**].

**Contination strategies for $\lambda$** The experiments presented in this paper were done for a fixed value of $\lambda$. To ensure that one really solves the original constrained problem, however, a suitable strategy for increasing $\lambda$ needs to be developed.

**Optimization strategies** In this paper we described a Gauss-Newton method to minimize the reduced penalty objective (**??**). Using the expressions for the gradient and Hessian presented in section 3.1, we can design other Newton variants.

Recently proposed stochastic algorithms aimed at dramatically reducing the cost of PDE-constrained optimization problems by working on a small subsets of the right-hand-sides [**?**, **?**, **?**, **?**] are directly applicable to this formulation.

## 8. 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 | $N_s \times N_f$ | $N_g$ | solve sparse SPSD system in $N_g$ unknowns |
| reduced | $2(N_s \times N_f)$ | $2N_g$ | solve matrix-free linear system in $N_g$ unknowns, requires $N_f \times N_s$ per mat-vec |
| all-at-once | 0 | $N_f \times N_s \times N_g$ | solve sparse symmetric, possibly indefinite system in $(N_f + N_s + 1) \times N_g$ unknowns |

**Table 1.** Leading order computation and storage costs per iteration of different methods; $N_s$ denotes the number of sources, $N_f$ denotes the number of frequencies and $N_g$ denotes the number of gridpoints. for large-scale seismic inverse problems we typically have $N_s = \mathcal{O}(10^6)$, $N_f = \mathcal{O}(10^1)$ and $N_g = \mathcal{O}(10^9)$

**Algorithm 1** Gauss-Newton algorithm based on the penalty formulation

**for** $t = 0$ to $N$ **do**
  $\mathbf{g}_t = 0$
  $H_t = 0$
  **for** $k = 1$ to $N_f$ **do**
    **for** $l = 1$ to $N_s$ **do**
      solve $\begin{pmatrix} \lambda_t A_k(\mathbf{m}_t) \\ P \end{pmatrix} \mathbf{u}_{kl} = \begin{pmatrix} \lambda_t \mathbf{q}_{kl} \\ \mathbf{d}_{kl} \end{pmatrix}$
      $\mathbf{g}_t = \mathbf{g}_t + \lambda_t^2 G_k(\mathbf{m}_t, \mathbf{u}_{kl})^* (A_k(\mathbf{m}_t)\mathbf{u}_{kl} - \mathbf{q}_{kl})$
      $H_t = H_t + (\lambda_t^2 - 1) G_k(\mathbf{m}_t, \mathbf{u}_{kl})^* G_k(\mathbf{m}_t, \mathbf{u}_{kl})$
    **end for**
  **end for**
  solve $H_t \Delta \mathbf{m}_t = -\mathbf{g}_t$
  update $\mathbf{m}_{t+1} = \mathbf{m}_t + \Delta \mathbf{m}_t$
  update $\lambda_t$.
**end for**

**Algorithm 2** Quasi-Newton algorithm based on the reduced formulation

**for** $t = 0$ to $N$ **do**
  $\mathbf{g}_t = 0$
  **for** $k = 1$ to $N_f$ **do**
    **for** $l = 1$ to $N_s$ **do**
      solve $A_k(\mathbf{m}_t)\mathbf{u}_{kl} = \mathbf{q}_{kl}$
      solve $A_k(\mathbf{m}_t)^* \mathbf{v}_{kl} = -P^*(P\mathbf{u}_{kl} - \mathbf{d}_{kl})$
      $\mathbf{g}_t = \mathbf{g}_t + G_k(\mathbf{m}_t, \mathbf{u}_{kl})^* \mathbf{v}_{kl}$
    **end for**
  **end for**
  apply L-BFGS Hessian $\Delta \mathbf{m}_t = \mathsf{lbfgs}(-\mathbf{g}_t, \{\mathbf{m}_{t-M}, \ldots, \mathbf{m}_t\}, \{\mathbf{g}_{t-M}, \ldots, \mathbf{g}_t\})$
  find a suitable steplength $\alpha$
  update $\mathbf{m}_{t+1} = \mathbf{m}_t + \alpha \Delta \mathbf{m}_t$
**end for**

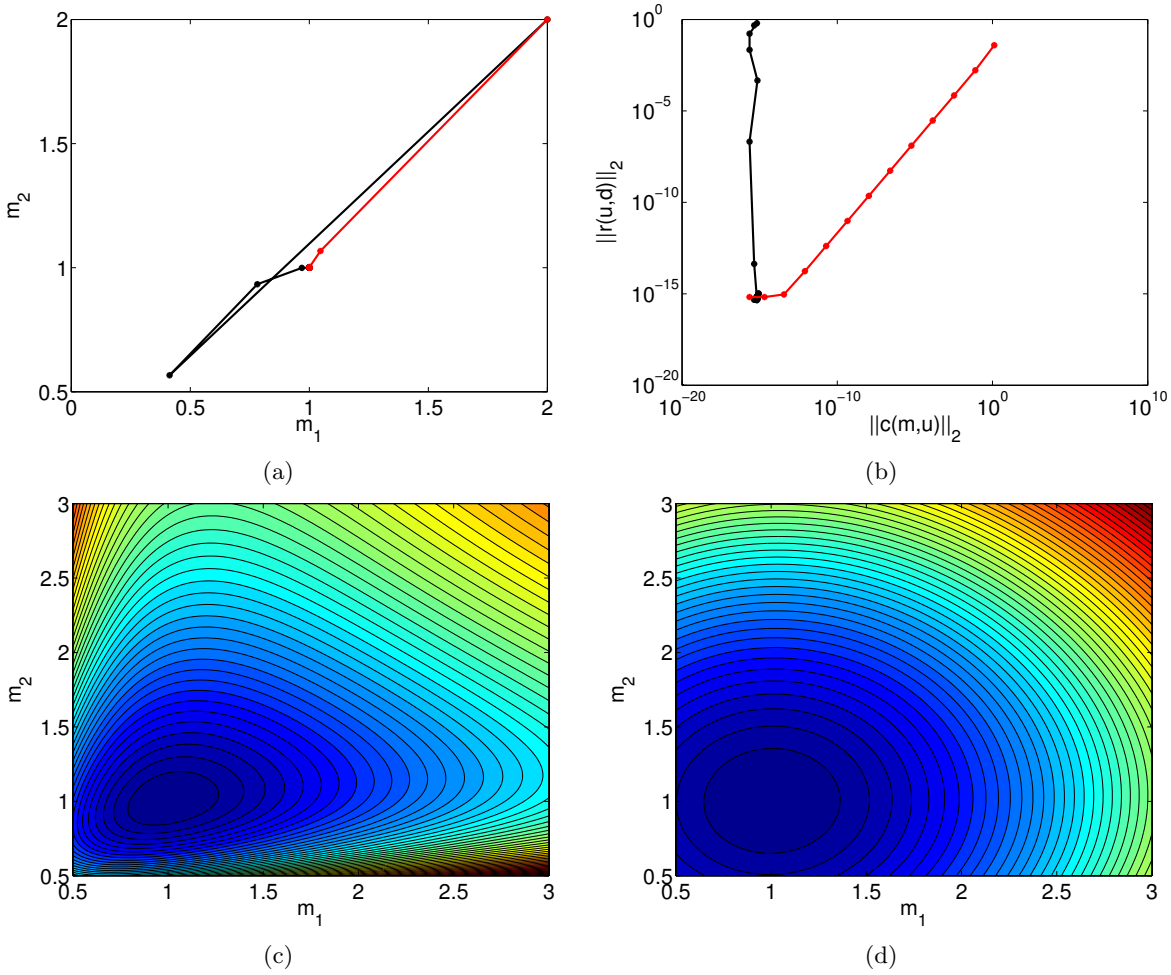


**Figure 1.** Illustration of the reduced and penalty approach on a simple 4 dimensional test problem. (a) Solution paths and (b) the convergence in terms of $||r||_2$ and $||c||_2$ for both the reduced (black) and penalty approaches (red). The objective functions corresponding to the reduced and penalty approaches are shown in (c) and (d) respectively.