LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-TR-819385

Bound Constrained Partial DifferentialEquation Inverse Problem Solution by theSemi-Smooth Newton Method

T. Hartland, C. G. Petra, N. Petra, J. Wang

February 10, 2021

Disclaimer

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

LLNL IM Release number: LLNL-TR-819385

Bound Constrained Partial Differential Equation Inverse Problem Solution by the Semi-Smooth Newton Method

Tucker Hartland[1], Cosmin G. Petra[2], Noémi Petra[1], and Jingyi Wang[2]

[1]Department of Applied Mathematics, University of California, Merced
[2]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory

February 12, 2021

Abstract

We present the mathematical derivation, software implementation details, and computational results for a semi-smooth Newton method applied to two inverse problems governed by partial differential equations with bound constraints. The two problems share mathematical structural similarities to density-based topology optimization problems. The semi-smooth Newton method provides a mesh independent solution computation for the two test problems. A key step is that the complementarity part of the necessary optimality conditions are reformulated with the use of a complementarity function ϕ such that the complementarity conditions are satisfied if and only if a zero of a nonsmooth function has been obtained. The modular finite element package `MFEM` is utilized for the software implementation. In addition we constructed a matrix-free `Operator` to enable the use of efficient Krylov subspace `IterativeSolver` of `MFEM` for the solution of our two target problems.

1 Introduction

We consider the problem of inverting for a distributed parameter in a partial differential equation (PDE) such that existing observational data for the state of the PDE is well matched in a L^2 norm sense. A complicating feature of our formulation of the problem stems from the requirement that the inversion parameter should satisfy hard bound constraints. The goal of this manuscript is to describe a semi-smooth Newton (ssN) numerical method for the solution of the inversion problem. The choice of ssN method is primarily motivated by the need to apply Hessian-based optimization algorithms to reduce the number of iterations and, thus, lessen the overall computation associated with first-order, gradient-based methods. For problems in topology optimization, a density field, constrained to be in $[0, 1]$, and other additional potential constraints that determine a PDE solution is to be estimated. The inverse problems with bound constraints studied here share a sufficient amount of mathematical structure with density-based topology optimization to afford an assessment of the efficacy of ssN for topology optimization.

Mathematically, the inversion problem considered in this work can be formulated as

$$\min_{(u,m)\in H^1(\Omega)\times L^2(\Omega)} J(u,m) = \frac{1}{2}||u-u_d||^2_{L^2(\Omega)} + \frac{1}{2}||m||^2_{\mathcal{R}} \tag{1}$$

such that

$$F(u,m) = 0 \quad \text{weakly on } \overline{\Omega}, \tag{2}$$

$$m_\ell \leq m \leq m_u \quad \text{a.e. in } \Omega. \tag{3}$$

Here m is a field that parametrizes an elliptic PDE, with appropriate boundary conditions, mathematically specified by $F(u, m)$, and u is the 'state' or solution of the PDE for a given m. Examples of such problems are presented in Section 3. The (semi)norm $||\cdot||_{\mathcal{R}}$ will be defined by means of a symmetric bilinear function R, namely,

$$||m||^2_{\mathcal{R}} = \int_\Omega R(m,m)\mathrm{d}V, \tag{4}$$

and has the role of regularizing the problem. In this report, we use an L^2 regularization for which one takes $R(m_1, m_2) = \gamma\, m_1\, m_2$. To regularize the highly oscillatory modes of m, one could take $R(m_1, m_2) = \gamma \boldsymbol{\nabla} m_1 \cdot \boldsymbol{\nabla} m_2$, which is known as H^1 regularization. In either case, $\gamma > 0$ is a regularization

parameter which controls the strength of the regularization. An alternative to regularizing the objective functional would be to determine the PDE solution u by $\tilde{m}$ obtained through the application of a smoothing filter to m [9].

We follow the optimize-then-discretize approach as a principle to derive mesh independent ssN schemes. The `MFEM` finite element library [1] is used to implement a ssN method for elliptic PDE-constrained optimization problems with bound constraints. Using this computational setup we demonstrate the mesh independence and superlinear rate of convergence of the semi-smooth Newton method. Finally, we comment on the software needs from `MFEM` for judicious implementation of ssN for infinite-dimensional problems and discuss limitations of the ssN method for topology optimization problems.

2 The Semi-Smooth Newton method for Inequality Constrained Optimization

Here we outline the ssN method in the context of Karush-Kuhn-Tucker (KKT) point estimation for finite-dimensional problems, consider

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad J(\boldsymbol{x}) \tag{5}$$

$$\text{such that } c_i(\boldsymbol{x}) \geq 0, \quad i = 1, 2, \ldots, k. \tag{6}$$

A necessary condition for a point $\boldsymbol{x}^*$ to be optimal [10] consists of the existence of k Lagrange multipliers $\lambda_1^*, \lambda_2^*, \ldots, \lambda_k^* \in \mathbb{R}$ such that

$$\boldsymbol{\nabla}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}, \tag{7}$$

$$\lambda_i^* \geq 0, \quad c_i(\boldsymbol{x}^*) \geq 0, \quad \lambda_i^* \, c_i(\boldsymbol{x}^*) = 0, \quad i = 1, 2, \ldots, k. \tag{8}$$

The Lagrangian function $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is defined as

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) := J(\boldsymbol{x}) - \sum_{i=1}^{m} \lambda_i \, c_i(\boldsymbol{x}). \tag{9}$$

Estimating a KKT point is numerically challenging, in part due to the lack of differentiability of complementarity equations (8). The complementarity conditions can be equivalently reformulated as a system of nonsmooth

equality conditions by utilizing a complementarity function, that is a function $\phi : \mathbb{R}^2 \to \mathbb{R}$ such that $c \geq 0$, $\lambda \geq 0$, $c\,\lambda = 0$ if and only if $\phi(c, \lambda) = 0$. For example, such functions are the min-complementarity function

$$\phi_{\min}(c, \lambda) = \min(c, \lambda),$$

and the Fischer-Burmeister complementarity function

$$\phi_{\mathrm{FB}}(c, \lambda) = c + \lambda - \sqrt{c^2 + \lambda^2}.$$

In the present work we use the complementarity function

$$\phi(c, \lambda) = \lambda - \min(0, \lambda - \sigma\, c), \tag{10}$$

following [7]. Above the constant $\sigma > 0$ parametrizes ϕ. Using the above function, a KKT point $(\boldsymbol{x}^*, \boldsymbol{\lambda}^*)$ must necessarily satisfy

$$\boldsymbol{\nabla}_{\boldsymbol{x}} \mathcal{L}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}, \tag{11a}$$
$$\boldsymbol{\Phi}(\boldsymbol{x}^*, \boldsymbol{\lambda}^*) = \mathbf{0}, \tag{11b}$$

where $\boldsymbol{\Phi} : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^m$ is defined by $(\boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{\lambda}))_i := \phi(c_i(\boldsymbol{x}), \lambda_i)$, $i = \{1, 2, \ldots, m\}$.

We note that $\boldsymbol{\Phi}$ is not everywhere continuously differentiable on $\mathbb{R}^n \times \mathbb{R}^m$, however it does improve upon the mathematical properties of the original complementarity equations (8); namely $\boldsymbol{\Phi}$ is Lipschitz, directionally differentiable, and slantly differentiable [7]. Slant differentiability of $\boldsymbol{\Phi}$ guarantees the existence of a slanting function $\boldsymbol{G}$ and of a generalized Jacobian of $\boldsymbol{\Phi}$ that enable semi-smooth Newton algorithm to converges to a KKT point at a superlinear rate.

3 Motivation: Example Problems

Here we outline two inverse problems with bound inequality and elliptic partial differential equality constraints. For both examples the first and second order adjoints are derived. For the first problem we will outline how the ssN method can be set up and will discuss nontrivial issues arising in the discretization of the systems of equations used in the ssN algorithm.

3.1 Example 1: Determination of a source term in Poisson's equation

In this problem we seek to determine the source term m, of Poisson's equation, within the lower and upper bounds m_ℓ, m_u, such that the solution of Poisson's equation u, matches a given data set u_d, namely,

$$\min_{(u,m)\in H^1(\Omega)\times L^2(\Omega)} J(u,m) := \frac{1}{2}||u-u_d||^2_{L^2(\Omega)} + \frac{\gamma}{2}||m||^2_{L_2(\Omega)} \tag{12}$$

such that

$$-\Delta u = m \quad \text{weakly in } \Omega, \tag{13}$$

$$u = u_0 \quad \text{on } \partial\Omega, \tag{14}$$

$$m_\ell \leq m \leq m_u \quad \text{a.e. in } \Omega. \tag{15}$$

The reader is refered to [12] for a detailed discussion of the above problem.

3.1.1 Forming the Lagrangian

We first introduce the following subsets of $H^1(\Omega)$:

$$\begin{aligned} \mathcal{V}_{u_0} &= \mathcal{V}(u_0) \text{ and} \\ \mathcal{V}_0 &= \mathcal{V}(0), \end{aligned}$$

where $\mathcal{V}(q) = \{v \in H^1(\Omega) \text{ such that } v = q,\ x \in \partial\Omega\}$. Following standard optimization theory, the Lagrangian functional $\mathcal{L} : \mathcal{V}_{u_0}\times\mathcal{V}_0\times L^2(\Omega)\times L^2(\Omega)\times L^2(\Omega) \to \mathbb{R}$ is

$$\begin{aligned} \mathcal{L}(u,p,m,z_\ell,z_u) :=& J(u,m) - \int_\Omega (\boldsymbol{\nabla} u\cdot\boldsymbol{\nabla} p - m\,p)\,\mathrm{d}V \\ &- \int_\Omega (z_\ell\,(m-m_\ell) + z_u\,(m_u-m))\,\mathrm{d}V. \end{aligned}$$

Above we introduced z_ℓ and z_u from $L^2(\Omega)$ to be the Lagrange multiplier functions associated to the lower and upper bound inequality constraints, respectively. The Lagrange multiplier associated with the partial differential equality constraint is denoted by p and hereafter referred to as the adjoint variable.

3.1.2 Optimality system

Having formed the Lagrangian functional, we then introduce the characterization of KKT point (u, p, m, z_ℓ, z_u) in the form of

$$\mathcal{L}_p\tilde{p} = 0, \ \ \forall \tilde{p} \in \mathcal{V}_0, \tag{16a}$$
$$\mathcal{L}_u\tilde{u} = 0, \ \ \forall \tilde{u} \in \mathcal{V}_0, \tag{16b}$$
$$\mathcal{L}_m\tilde{m} = 0, \ \ \forall \tilde{m} \in L^2(\Omega), \tag{16c}$$
$$z_\ell \geq 0, \ \ m \geq m_\ell, \ \ z_\ell (m - m_\ell) = 0 \quad \text{a.e. in } \Omega, \tag{16d}$$
$$z_u \geq 0, \ \ m_u \geq m, \ \ z_u (m_u - m) = 0 \quad \text{a.e. in } \Omega, \tag{16e}$$

where $\mathcal{L}_p\tilde{p}$, denotes the first variation [4] of $\mathcal{L}$ with respect to the adjoint variable p in direction $\tilde{p}$. Moreover $\mathcal{L}_u\tilde{u}$ and $\mathcal{L}_m\tilde{m}$ are similarly defined in the form of

$$\begin{aligned}
\mathcal{L}_p\tilde{p} &= -\int_\Omega (\boldsymbol{\nabla} u \cdot \boldsymbol{\nabla}\tilde{p} - m\,\tilde{p})\,\mathrm{d}V, \\
\mathcal{L}_u\tilde{u} &= J_u, \tilde{u} - \int_\Omega \boldsymbol{\nabla}\tilde{u} \cdot \boldsymbol{\nabla} p\,\mathrm{d}V, \\
\mathcal{L}_m\tilde{m} &= J_m\tilde{m} + \int_\Omega \tilde{m}\,p\,\mathrm{d}V - \int_\Omega \tilde{m}\,(z_\ell - z_u)\,\mathrm{d}V, \\
J_u\tilde{u} &= \int_\Omega \tilde{u}\,(u - u_d)\,\mathrm{d}V, \\
J_m\tilde{m} &= \int_\Omega R\,(m, \tilde{m})\,\mathrm{d}V.
\end{aligned}$$

3.1.3 The semismooth reformulation of the optimality system

The mathematical character of equations (16d), (16e) is significantly different from (16a), (16b), (16c) because one is a pointwise constraint and the other is an orthogonality relation between elements of a Hilbert space. Assuming that all the fields are continuous we can recast (16d), (16e) as

$$\begin{aligned}
\Phi^{(\ell)}\tilde{z}_\ell = 0, \ \ \forall \tilde{z}_\ell \in L^2(\Omega), \\
\Phi^{(u)}\tilde{z}_u = 0, \ \ \forall \tilde{z}_u \in L^2(\Omega),
\end{aligned}$$

where $\Phi^{(\ell)}$, $\Phi^{(u)}$ are defined by

$$\Phi^{(\ell)}(m, z_\ell)\,\tilde{z}_\ell := \int_\Omega \tilde{z}_\ell \phi(m - m_\ell, z_\ell)\,\mathrm{d}V, \tag{17a}$$

$$\Phi^{(u)}(m, z_u)\,\tilde{z}_u := \int_\Omega \tilde{z}_u \phi(m_u - m, z_u)\,\mathrm{d}V, \tag{17b}$$

and ϕ is the complementarity function defined in (10). Equation (17) allows us to obtain the following equivalent optimality system

$$\mathcal{L}_p \tilde{p} = 0, \ \ \forall \tilde{p} \in \mathcal{V}_0, \tag{18a}$$

$$\mathcal{L}_u \tilde{u} = 0, \ \ \forall \tilde{u} \in \mathcal{V}_0, \tag{18b}$$

$$\mathcal{L}_m \tilde{m} = 0, \ \ \forall \tilde{m} \in L^2(\Omega), \tag{18c}$$

$$\Phi^{(\ell)} \tilde{z}_\ell = 0, \ \ \forall \tilde{z}_\ell \in L^2(\Omega), \tag{18d}$$

$$\Phi^{(u)} \tilde{z}_u = 0, \ \ \forall \tilde{z}_u \in L^2(\Omega), \tag{18e}$$

which should be interpreted as a generalization of the finite dimensional ssN system (11).

3.1.4 The ssN system

In order to determine the solution of (18), we apply a semi-smooth Newton method whose updates are found by linearizing (18) and solving the linearized system, that is, we seek to find updates $(\hat{p}, \hat{u}, \hat{m}, \hat{z}_\ell, \hat{z}_u)$ that satisfy

$$\mathcal{L}_{p,u}(\tilde{p}, \hat{u}) + \mathcal{L}_{p,m}(\tilde{p}, \hat{m}) = -\mathcal{L}_p \tilde{p}, \ \ \forall \tilde{p} \in \mathcal{V}_0, \tag{19a}$$

$$\mathcal{L}_{u,p}(\tilde{u}, \hat{p}) + \mathcal{L}_{u,u}(\tilde{u}, \hat{u}) = -\mathcal{L}_u \tilde{u}, \ \ \forall \tilde{u} \in \mathcal{V}_0, \tag{19b}$$

$$\begin{aligned} &\mathcal{L}_{m,p}(\tilde{m}, \hat{p}) + \mathcal{L}_{m,m}(\tilde{m}, \hat{m}) + \\ &\mathcal{L}_{m,z_\ell}(\tilde{m}, \hat{z}_\ell) + \mathcal{L}_{m,z_u}(\tilde{m}, \hat{z}_u) = -\mathcal{L}_m \tilde{m}, \ \ \forall \tilde{m} \in L^2(\Omega), \end{aligned} \tag{19c}$$

$$\Phi_m^{(\ell)}(\tilde{z}_\ell, \hat{m}) + \Phi_{z_\ell}^{(\ell)}(\tilde{z}_\ell, \hat{z}_\ell) = -\Phi^{(\ell)} \tilde{z}_\ell, \ \ \forall \tilde{z}_\ell \in L^2(\Omega), \tag{19d}$$

$$\Phi_m^{(u)}(\tilde{z}_u, \hat{m}) + \Phi_{z_u}^{(u)}(\tilde{z}_u, \hat{z}_u) = -\Phi^{(u)} \tilde{z}_u, \ \ \forall \tilde{z}_u \in L^2(\Omega), \tag{19e}$$

$$
\begin{aligned}
\mathcal{L}_{p,u}\left(\tilde{p},\hat{u}\right) &= -\int_{\Omega} \boldsymbol{\nabla}\hat{u}\cdot\boldsymbol{\nabla}\tilde{p}\,\mathrm{d}V,\\
\mathcal{L}_{p,m}\left(\tilde{p},\hat{m}\right) &= \int_{\Omega} \tilde{p}\,\hat{m}\,\mathrm{d}V,\\
\mathcal{L}_{u,u}\left(\tilde{u},\hat{u}\right) &= \int_{\Omega} \tilde{u}\,\hat{u}\,\mathrm{d}V,\\
\mathcal{L}_{m,m}\left(\tilde{m},\hat{m}\right) &= \int_{\Omega} R\left(\tilde{m},\hat{m}\right)\mathrm{d}V,\\
\mathcal{L}_{m,z_\ell}\left(\tilde{m},\hat{z}_\ell\right) &= -\int_{\Omega} \tilde{m}\,\hat{z}_\ell\,\mathrm{d}V,\\
\mathcal{L}_{m,z_u}\left(\tilde{m},\hat{z}_u\right) &= \int_{\Omega} \tilde{m}\,\hat{z}_u\,\mathrm{d}V.
\end{aligned}
$$

To facilitate the determination of variations of the functionals $\Phi^{(\ell)}, \Phi^{(u)}$, that are defined in terms of the piecewise complementarity function ϕ, we describe the following sets

$$
\begin{aligned}
\Omega_{\mathcal{I}^{(\ell)}} &:= \{\boldsymbol{x}\in\Omega \text{ such that } z_\ell - \sigma\,(m-m_\ell) < 0\}, && \text{(20a)}\\
\Omega_{\mathcal{I}^{(u)}} &:= \{\boldsymbol{x}\in\Omega \text{ such that } z_u - \sigma\,(m_u-m) < 0\}, && \text{(20b)}\\
\Omega_{\mathcal{A}^{(\ell)}} &:= \Omega\setminus\Omega_{\mathcal{I}^{(\ell)}}, && \text{(20c)}\\
\Omega_{\mathcal{A}^{(u)}} &:= \Omega\setminus\Omega_{\mathcal{I}^{(u)}}, && \text{(20d)}\\
\Omega_{\mathcal{A}} &:= \Omega_{\mathcal{A}^{(\ell)}}\cup\Omega_{\mathcal{A}^{(u)}}, && \text{(20e)}\\
\Omega_{\mathcal{I}} &:= \Omega_{\mathcal{I}^{(\ell)}}\cap\Omega_{\mathcal{I}^{(u)}}. && \text{(20f)}
\end{aligned}
$$

Given (20) and how the complementarity function ϕ is defined in (10), it quickly follows that

$$
\begin{aligned}
\boldsymbol{x}\in\Omega_{\mathcal{I}^{(\ell)}} &\implies \phi\left(m-m_\ell, z_\ell\right) = z_\ell,\\
\boldsymbol{x}\in\Omega_{\mathcal{A}^{(\ell)}} &\implies \phi\left(m-m_\ell, z_\ell\right) = \sigma\,\left(m-m_\ell\right).
\end{aligned}
$$

The above implications allow for the functional $\Phi^{(\ell)}$ to be expressed as

$$
\Phi^{(\ell)}\tilde{z}_\ell = \int_{\Omega_{\mathcal{I}^{(\ell)}}} \tilde{z}_\ell\,z_\ell\,\mathrm{d}V + \sigma\int_{\Omega_{\mathcal{A}^{(\ell)}}} \tilde{z}_\ell\,\left(m-m_\ell\right)\,\mathrm{d}V.
$$

Furthermore, we obtain the variations $\Phi^{(\ell)}_{z_\ell}$ and $\Phi^{(\ell)}_{m}$ of $\Phi^{(\ell)}$ given by the above equation in the following form:

$$\Phi^{(\ell)}_{z_\ell}\left(\tilde{z}_\ell, \hat{z}_\ell\right) = \int_{\Omega_{\mathcal{I}^{(\ell)}}} \tilde{z}_\ell\, \hat{z}_\ell \,\mathrm{d}V,$$
$$\Phi^{(\ell)}_{m}\left(\tilde{z}_\ell, \hat{m}\right) = \sigma \int_{\Omega_{\mathcal{A}^{(\ell)}}} \tilde{z}_\ell\, \hat{m} \,\mathrm{d}V.$$

Similar expressions for $\Phi^{(\ell)}$ and its variations can be derived in the form of

$$\Phi^{(u)}\tilde{z}_u = \int_{\Omega_{\mathcal{I}^{(u)}}} \tilde{z}_u\, z_u \,\mathrm{d}V + \sigma \int_{\Omega_{\mathcal{A}^{(u)}}} \tilde{z}_u\, \left(m_u - m\right) \,\mathrm{d}V,$$
$$\Phi^{(u)}_{z_u}\left(\tilde{z}_u, \hat{z}_u\right) = \int_{\Omega_{\mathcal{I}^{(u)}}} \tilde{z}_u\, \hat{z}_u \,\mathrm{d}V,$$
$$\Phi^{(u)}_{m}\left(\tilde{z}_u, \hat{m}\right) = -\sigma \int_{\Omega_{\mathcal{A}^{(u)}}} \tilde{z}_u\, \hat{m} \,\mathrm{d}V.$$

We now show how the semi-smooth Newton method can be regarded as an active set method for this problem. We first observe that

$$\Phi^{(\ell)}_{m}\left(\tilde{z}_\ell, \hat{m}\right) + \Phi^{(\ell)}_{z_\ell}\left(\tilde{z}_\ell, \hat{z}_\ell\right) = -\Phi^{(\ell)}\tilde{z}_\ell,$$

holds provided the following two identities are true

$$\sigma \int_{\Omega_{\mathcal{A}^{(\ell)}}} \tilde{z}_\ell\, \hat{m} \,\mathrm{d}V = -\sigma \int_{\Omega_{\mathcal{A}^{(\ell)}}} \tilde{z}_\ell\, \left(m - m_\ell\right) \mathrm{d}V, \tag{21}$$
$$\int_{\Omega_{\mathcal{I}^{(\ell)}}} \tilde{z}_\ell\, \hat{z}_\ell \,\mathrm{d}V = -\int_{\Omega_{\mathcal{I}^{(\ell)}}} \tilde{z}_\ell\, z \,\mathrm{d}V. \tag{22}$$

That is, the ssN method parameter update $m + \hat{m}$, will satisfy

$$m + \hat{m} = m_\ell, \;\; \boldsymbol{x} \in \Omega_{\mathcal{A}^{(\ell)}},$$
$$z_\ell + \hat{z}_\ell = 0, \;\; \boldsymbol{x} \in \Omega_{\mathcal{I}^{(\ell)}}.$$

Similarly, $\Phi^{(u)}_{m}\left(\tilde{z}_u, \hat{m}\right) + \Phi^{(u)}_{z_u}\left(\tilde{z}_u, \hat{z}_u\right) = -\Phi^{(u)}\tilde{z}_u$, whenever

$$m + \hat{m} = m_u, \;\; \boldsymbol{x} \in \Omega_{\mathcal{A}^{(u)}}, \tag{23}$$
$$z_u + \hat{z}_u = 0, \;\; \boldsymbol{x} \in \Omega_{\mathcal{I}^{(u)}}. \tag{24}$$

Finally we observe that in order that the ssN parameter update $\hat{m}$ to be well defined, it is necessary that $\Omega_{\mathcal{A}^{(\ell)}} \cap \Omega_{\mathcal{A}^{(u)}} = \emptyset$, which coincides with the update of an active-set method.

3.1.5 Method of numerical solution

Given $m(\boldsymbol{x}), z_\ell(\boldsymbol{x}), z_u(\boldsymbol{x})$, we seek to compute the associated ssN update $\hat{m}(\boldsymbol{x}), \hat{z}_\ell(\boldsymbol{x}), \hat{z}_u(\boldsymbol{x})$. We proceed by presenting this infinite-dimensional consistent algorithm by means of a discretization conducted by finite elements. For that purpose, let us define

- $\mathcal{V}_h \subset H^1(\Omega)$, subspace with basis elements $\{\varphi_j(\boldsymbol{x})\}_{j=1}^n$.
- $\mathcal{M}_h \subset L^2(\Omega)$, subspace with basis elements $\{\psi_j(\boldsymbol{x})\}_{j=1}^m$.
- $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ – $\mathcal{V}_h$ discretization of the bilinear form $\int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} p \,\mathrm{d}V$.
- $\boldsymbol{M} \in \mathbb{R}^{m\times m}$ – $\mathcal{M}_h$ discretization of the bilinear form $\int_\Omega m_1\, m_2 \,\mathrm{d}V$.
- $\boldsymbol{M}_{\mathcal{V}} \in \mathbb{R}^{n\times n}$ – $\mathcal{V}_h$ discretization of the bilinear form $\int_\Omega u\, p \,\mathrm{d}V$.
- $\boldsymbol{C} \in \mathbb{R}^{m\times n}$ – $\mathcal{V}_h, \mathcal{M}_h$ mixed discretization of the bilinear form $\int_\Omega p\, m \,\mathrm{d}V$.
- $\boldsymbol{R} \in \mathbb{R}^{m\times m}$ – $\mathcal{M}_h$ discretization of the bilinear form $\int_\Omega R(m_1, m_2) \,\mathrm{d}V$.

Vectors of finite element expansion coefficients $\boldsymbol{m}, \boldsymbol{u}$ are used to represent functions m, u, contained in the finite dimensional spaces $\mathcal{V}_h, \mathcal{M}_h$, respectively.

In order to reduce the amount of notation, in the remainder of this section the Dirichlet conditions that are used to obtain a discretized operator are not explicitly stated and should be inferred from the context. For instance, the state and adjoint variables are both solutions of a Poisson equation with distinct Dirichlet conditions; also we use the same symbol for the discretized Laplacian operators for the two Poisson equations.

For given discretized parameter and bound constraint Lagrange multipliers $\boldsymbol{m}, \boldsymbol{z}_\ell, \boldsymbol{z}_u$, one can determine the state and adjoint variables by solving the following linear systems in sequence

$$\boldsymbol{A}\boldsymbol{u} = \boldsymbol{C}^\top \boldsymbol{m}, \tag{25}$$

$$\boldsymbol{A}\boldsymbol{p} = \boldsymbol{M}_{\mathcal{V}}(\boldsymbol{u} - \boldsymbol{u}_d), \tag{26}$$

by, for instance, the preconditioned conjugate gradient method. The state and adjoint variables allow for the adjoint-based evaluation of the gradient

$$\boldsymbol{g} = \boldsymbol{R}\boldsymbol{m} + \boldsymbol{C}\boldsymbol{p} - \boldsymbol{M}(\boldsymbol{z}_\ell - \boldsymbol{z}_u). \tag{27}$$

The gradient $\boldsymbol{g}$, is with respect to the Euclidean inner product on $\mathbb{R}^m$ and is consistent with the infinite dimensional inverse problem, that is, for each $\tilde{m} \in L^2(\Omega)$, the discrepancy $|\tilde{\boldsymbol{m}}^\top \boldsymbol{g} - \mathcal{L}_m \tilde{m}|$, vanishes as the mesh is refined.

We now discuss solving the ssN system, whose solution determines updates of the parameter and bound constraint Lagrange multipliers. We follow a similar approach to that of [11] that employs the Newton's method to solve PDE inverse problems with log barrier terms for problems that contain pointwise inequality bound constraints. The ssN system in the absence of terms that determine the active and inactive sets is

$$-\boldsymbol{A}\hat{\boldsymbol{u}} + \boldsymbol{C}^\top \hat{\boldsymbol{m}} = \boldsymbol{0}, \tag{28}$$

$$-\boldsymbol{A}\hat{\boldsymbol{p}} + \boldsymbol{M}_{\mathcal{V}}\hat{\boldsymbol{u}} = \boldsymbol{0}, \tag{29}$$

$$\boldsymbol{C}\hat{\boldsymbol{p}} + \boldsymbol{R}\hat{\boldsymbol{m}} + \boldsymbol{M}\left(\hat{\boldsymbol{z}}_u - \hat{\boldsymbol{z}}_\ell\right) = -\boldsymbol{g}. \tag{30}$$

By eliminating the second-order state and the adjoint variables $\hat{\boldsymbol{u}}, \hat{\boldsymbol{p}}$, the following reduced Newton system is formed:

$$\boldsymbol{H}\hat{\boldsymbol{m}} + \boldsymbol{M}\left(\hat{\boldsymbol{z}}_u - \hat{\boldsymbol{z}}_\ell\right) = -\boldsymbol{g}, \tag{31}$$

$$\boldsymbol{H} = \boldsymbol{C}\boldsymbol{A}^{-1}\boldsymbol{M}_{\mathcal{V}}\boldsymbol{A}^{-1}\boldsymbol{C}^\top + \boldsymbol{R}. \tag{32}$$

The following computational strategy is employed to compute the active and inactive sets for the discretized problem and the subsequent active set constraint impostion on the Newton system. We assume a finite element discretization such that the components of the vector representation corresponds to the evaluation of the finite element function at a set of points $\{\boldsymbol{x}_i\}_i \subset \Omega$. Finite-dimensional representations of the sets $\Omega_{\mathcal{I}^{(\ell)}}, \Omega_{\mathcal{A}^{(\ell)}}, \Omega_{\mathcal{I}^{(u)}}, \Omega_{\mathcal{A}^{(u)}}$ are given by the index sets

$$\boldsymbol{\mathcal{I}}^{(\ell)} = \{i \text{ s.t. } \left(\boldsymbol{z}_\ell - \sigma\left(\boldsymbol{m} - \boldsymbol{m}_\ell\right)\right)_i \leq 0\}, \tag{33a}$$

$$\boldsymbol{\mathcal{A}}^{(\ell)} = \{i \text{ s.t. } \left(\boldsymbol{z}_\ell - \sigma\left(\boldsymbol{m} - \boldsymbol{m}_\ell\right)\right)_i > 0\}, \tag{33b}$$

$$\boldsymbol{\mathcal{I}}^{(u)} = \{i \text{ s.t. } \left(\boldsymbol{z}_u - \sigma\left(\boldsymbol{m}_u - \boldsymbol{m}\right)\right)_i \leq 0\}, \tag{33c}$$

$$\boldsymbol{\mathcal{A}}^{(u)} = \{i \text{ s.t. } \left(\boldsymbol{z}_u - \sigma\left(\boldsymbol{m}_u - \boldsymbol{m}\right)\right)_i > 0\}. \tag{33d}$$

The discretized parameter components $\boldsymbol{\mathcal{A}} = \boldsymbol{\mathcal{A}}^{(\ell)} \cup \boldsymbol{\mathcal{A}}^{(u)}$ are considered active, that is, the components of $\hat{\boldsymbol{m}}$ are not determined by (31) but rather that $\left(\boldsymbol{m} + \hat{\boldsymbol{m}}\right)_i = \left(\boldsymbol{m}_\ell\right)_i$, for each $i \in \boldsymbol{\mathcal{A}}^{(\ell)}$ and $\left(\boldsymbol{m} + \hat{\boldsymbol{m}}\right)_i = \left(\boldsymbol{m}_u\right)_i$ for

each $i \in \boldsymbol{\mathcal{A}}^{(u)}$. To facilitate an incorporation of the active set constraints, we define the operator $\boldsymbol{P}_{\boldsymbol{\mathcal{A}}}$ that projects vectors such as $\boldsymbol{m}$, to the $|\boldsymbol{\mathcal{A}}|$-dimensional vector space of active degrees of freedom. The projector $\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}$ to the space of inactive degrees of freedom can be similarly defined. We utilize the aforementioned projection operators to explicitly expose the components of $\hat{\boldsymbol{m}}$ and $\hat{\boldsymbol{\lambda}}$ that are not determined by active or inactive set constraints in (31); for this we apply $\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}$ to obtain

$$\begin{aligned}
\boldsymbol{H}\hat{\boldsymbol{m}} - \boldsymbol{M}\hat{\boldsymbol{\lambda}} &= -\boldsymbol{g}, \\
\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}^{\top}\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{I}}} - \boldsymbol{M}\boldsymbol{P}_{\boldsymbol{\mathcal{A}}}^{\top}\hat{\boldsymbol{\lambda}}_{\boldsymbol{\mathcal{A}}} &= -\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{\mathcal{A}}}^{\top}\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{A}}} + \boldsymbol{M}\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}^{\top}\hat{\boldsymbol{\lambda}}_{\boldsymbol{\mathcal{I}}} - \boldsymbol{g}, \\
\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}^{\top}\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{I}}} - \boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{M}\boldsymbol{P}_{\boldsymbol{\mathcal{A}}}^{\top}\hat{\boldsymbol{\lambda}}_{\boldsymbol{\mathcal{A}}} &= -\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{\mathcal{A}}}^{\top}\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{A}}} + \boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{M}\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}^{\top}\hat{\boldsymbol{\lambda}}_{\boldsymbol{\mathcal{I}}} - \boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{g},
\end{aligned}$$

where $\hat{\boldsymbol{\lambda}} = \hat{\boldsymbol{z}}_{\ell} - \hat{\boldsymbol{z}}_{u}$ is used to reduce notation. For a piecewise-constant finite element discretization of m, the mass matrix $\boldsymbol{M}$ is diagonal, which implies $\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{M}\boldsymbol{P}_{\boldsymbol{\mathcal{A}}}^{\top} = \boldsymbol{0}$. For such mass matrices, we observe that $\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{I}}}$ and $\hat{\boldsymbol{\lambda}}_{\boldsymbol{\mathcal{A}}}$ are algebraically decoupled and, as a consequence, $\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{I}}}$ can then be written as the solution of the symmetric linear system

$$\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}^{\top}\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{I}}} = -\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{\mathcal{A}}}^{\top}\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{A}}} + \boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{M}\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}^{\top}\hat{\boldsymbol{\lambda}}_{\boldsymbol{\mathcal{I}}} - \boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{g}.$$

The above-mentioned algebraic decoupling is preferable and thus we approximate $\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{M}\boldsymbol{P}_{\boldsymbol{\mathcal{A}}}^{\top}$ by the zero matrix even for those finite element discretizations in which the mass matrix is not diagonal. We now summarize the steps needed to determine $\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{I}}}$:

$$\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}^{\top}\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{I}}} = -\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{\mathcal{A}}}^{\top}\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{A}}} + \boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{M}\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}^{\top}\hat{\boldsymbol{\lambda}}_{\boldsymbol{\mathcal{I}}} - \boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{g}, \tag{34a}$$

$$\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{A}}} = \boldsymbol{P}_{\boldsymbol{\mathcal{A}}}\left(\boldsymbol{m}_{\ell} - \boldsymbol{m}\right), \tag{34b}$$

$$\hat{\boldsymbol{\lambda}}_{\boldsymbol{\mathcal{I}}} = \hat{\boldsymbol{z}}_{\ell,\boldsymbol{\mathcal{I}}} - \hat{\boldsymbol{z}}_{u,\boldsymbol{\mathcal{I}}}, \tag{34c}$$

$$\hat{\boldsymbol{z}}_{\ell,\boldsymbol{\mathcal{I}}} = -\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{z}_{\ell}, \tag{34d}$$

$$\hat{\boldsymbol{z}}_{u,\boldsymbol{\mathcal{I}}} = -\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{z}_{u}. \tag{34e}$$

For the given convex optimization problem, the Hessian will be symmetric positive definite as will be the submatrix $\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}^{\top}$. This property can be exploited to solve the linear system (34a) efficiently utilizing the preconditoned conjugate-gradient algorithm. Had $\boldsymbol{P}_{\boldsymbol{\mathcal{I}}}\boldsymbol{M}\boldsymbol{P}_{\boldsymbol{\mathcal{A}}}^{\top}$ not been nullified, a nonsymmetric linear system would need to be solved in order to simultaneously determine $\hat{\boldsymbol{m}}_{\boldsymbol{\mathcal{A}}}$ and $\hat{\boldsymbol{\lambda}}_{\boldsymbol{\mathcal{I}}}$.

Having infeasible iterates can cause the optimization algorithm to fail, such as for those problems for which the state equation is not well defined outside of the feasible region, e.g., subsection 3.2. For that reason, we choose to project $\hat{\boldsymbol{m}}$ to ensure that the subsequent iterate $\boldsymbol{m} + \hat{\boldsymbol{m}}$ is in the feasible region and, thus, constitutes a primal feasible ssN iterate update. Once $\hat{\boldsymbol{m}}$ is known, we need only determine $\hat{\boldsymbol{z}}_\ell$ and $\hat{\boldsymbol{z}}_u$, the steps needed to compute them are summarized below:

$$\boldsymbol{M}\hat{\boldsymbol{\lambda}} = \boldsymbol{H}\hat{\boldsymbol{m}} + \boldsymbol{g}, \tag{35a}$$

$$(\hat{\boldsymbol{z}}_\ell)_i = -(\boldsymbol{z}_\ell)_i, \quad i \in \boldsymbol{\mathcal{I}}_\ell, \tag{35b}$$

$$(\hat{\boldsymbol{z}}_u)_i = -(\boldsymbol{z}_u)_i, \quad i \in \boldsymbol{\mathcal{I}}_u, \tag{35c}$$

$$(\hat{\boldsymbol{z}}_\ell)_i = \left(\hat{\boldsymbol{z}}_u + \hat{\boldsymbol{\lambda}}\right)_i, \quad i \in \boldsymbol{\mathcal{A}}_\ell, \tag{35d}$$

$$(\hat{\boldsymbol{z}}_u)_i = \left(\hat{\boldsymbol{z}}_\ell - \hat{\boldsymbol{\lambda}}\right)_i, \quad i \in \boldsymbol{\mathcal{A}}_u. \tag{35e}$$

3.2 Example 2: Determination of a diffusion coefficient in Poisson's equation

In this section, we seek to determine the spatially varying diffusion coefficient m of Poisson's equation within given lower and upper bounds m_ℓ, m_u such that the PDE solution u matches a given data set u_d. This can be mathematically written as:

$$\min_{(u,m)\in H^1(\Omega)\times L^2(\Omega)} J(u,m) := \frac{1}{2}||u - u_d||^2_{L^2(\Omega)} + \frac{1}{2}||m||^2_{\mathcal{R}} \tag{36}$$

such that

$$-\boldsymbol{\nabla}\cdot(m\,\boldsymbol{\nabla}u) = f \quad \text{weakly in } \Omega, \tag{37}$$

$$u = u_0 \quad \text{on } \partial\Omega, \tag{38}$$

$$m_\ell \le m \le m_u \quad \text{a.e. in } \Omega. \tag{39}$$

As before we follow the Lagrangian formalism by introducing a Lagrangian functional $\mathcal{L} : \mathcal{V}_{u_0} \times \mathcal{V}_0 \times L^2(\Omega) \times L^2(\Omega) \times L^2(\Omega) \to \mathbb{R}$ in the form of

$$\begin{aligned}\mathcal{L}(u,p,m,z_\ell,z_u) :=& J(u,m) - \int_\Omega (m\boldsymbol{\nabla}u\cdot\boldsymbol{\nabla}p - f\,p)\,\mathrm{d}V \\ &- \int_\Omega (z_\ell\,(m-m_\ell) + z_u\,(m_u - m))\,\mathrm{d}V.\end{aligned}$$

In order to determine a KKT point (u, p, m, z_ℓ, z_u) we require that

$$\mathcal{L}_p \tilde{p} = 0, \ \ \forall \tilde{p} \in \mathcal{V}_0, \tag{40a}$$
$$\mathcal{L}_u \tilde{u} = 0, \ \ \forall \tilde{u} \in \mathcal{V}_0, \tag{40b}$$
$$\mathcal{L}_m \tilde{m} = 0, \ \ \forall \tilde{m} \in L^2(\Omega), \tag{40c}$$
$$z_\ell \geq 0, \ \ m \geq m_\ell, \ \ z_\ell (m - m_\ell) = 0 \quad \text{a.e. in } \Omega, \tag{40d}$$
$$z_u \geq 0, \ \ m_u \geq m, \ \ z_u (m_u - m) = 0 \quad \text{a.e. in } \Omega. \tag{40e}$$

Above we introduced the following quantities:

$$\begin{aligned}
\mathcal{L}_p \tilde{p} &= -\int_\Omega (m \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} \tilde{p} - f \tilde{p}) \, \mathrm{d}V, \\
\mathcal{L}_u \tilde{u} &= J_u \tilde{u} - \int_\Omega m \boldsymbol{\nabla} \tilde{u} \cdot \boldsymbol{\nabla} p \, \mathrm{d}V, \\
\mathcal{L}_m \tilde{m} &= J_m \tilde{m} - \int_\Omega \tilde{m} \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} p \, \mathrm{d}V - \int_\Omega \tilde{m} (z_\ell - z_u) \, \mathrm{d}V, \\
J_u \tilde{u} &= \int_\Omega \tilde{u} (u - u_d) \, \mathrm{d}V, \\
J_m \tilde{m} &= \int_\Omega R(m, \tilde{m}) \, \mathrm{d}V.
\end{aligned}$$

3.2.1 The Semismooth reformulation of the optimality system

Again we utilize the functionals $\Phi^{(\ell)}$, $\Phi^{(u)}$ defined by

$$\Phi^{(\ell)}(m, z_\ell) \, \tilde{z}_\ell := \int_\Omega \tilde{z}_\ell \phi(m - m_\ell, z_\ell) \, \mathrm{d}V, \tag{41a}$$
$$\Phi^{(u)}(m, z_u) \, \tilde{z}_u := \int_\Omega \tilde{z}_u \phi(m_u - m, z_u) \, \mathrm{d}V. \tag{41b}$$

This allows for the following equivalent reformulation of the optimality system

$$\mathcal{L}_p \tilde{p} = 0, \ \ \forall \tilde{p} \in \mathcal{V}_0, \tag{42a}$$
$$\mathcal{L}_u \tilde{u} = 0, \ \ \forall \tilde{u} \in \mathcal{V}_0, \tag{42b}$$
$$\mathcal{L}_m \tilde{m} = 0, \ \ \forall \tilde{m} \in L^2(\Omega), \tag{42c}$$
$$\Phi^{(\ell)} \tilde{z}_\ell = 0, \ \ \forall \tilde{z}_\ell \in L^2(\Omega), \tag{42d}$$
$$\Phi^{(u)} \tilde{z}_u = 0, \ \ \forall \tilde{z}_u \in L^2(\Omega). \tag{42e}$$

The semi-smooth Newton method is then applied to iteratively obtain updates $(\hat{p}, \hat{u}, \hat{m}, \hat{z}_\ell, \hat{z}_u)$ which solve the following linear system

$$\mathcal{L}_{p,u}(\tilde{p}, \hat{u}) + \mathcal{L}_{p,m}(\tilde{p}, \hat{m}) = -\mathcal{L}_p\tilde{p}, \quad \forall \tilde{p} \in \mathcal{V}_0, \tag{43a}$$

$$\mathcal{L}_{u,p}(\tilde{u}, \hat{p}) + \mathcal{L}_{u,u}(\tilde{u}, \hat{u}) + \mathcal{L}_{u,m}(\tilde{u}, \hat{m}) = -\mathcal{L}_u\tilde{u}, \quad \forall \tilde{u} \in \mathcal{V}_0, \tag{43b}$$

$$\begin{aligned}\mathcal{L}_{m,p}(\tilde{m}, \hat{p}) + \mathcal{L}_{m,u}(\tilde{m}, \hat{u}) + \mathcal{L}_{m,m}(\tilde{m}, \hat{m}) + \\ \mathcal{L}_{m,z_\ell}(\tilde{m}, \hat{z}_\ell) + \mathcal{L}_{m,z_u}(\tilde{m}, \hat{z}_u) = -\mathcal{L}_m\tilde{m}, \quad \forall \tilde{m} \in L^2(\Omega),\end{aligned} \tag{43c}$$

$$\Phi^{(\ell)}_m(\tilde{z}_\ell, \hat{m}) + \Phi^{(\ell)}_{z_\ell}(\tilde{z}_\ell, \hat{z}_\ell) = -\Phi^{(\ell)}\tilde{z}_\ell, \quad \forall \tilde{z}_\ell \in L^2(\Omega), \tag{43d}$$

$$\Phi^{(u)}_m(\tilde{z}_u, \hat{m}) + \Phi^{(u)}_{z_u}(\tilde{z}_u, \hat{z}_u) = -\Phi^{(u)}\tilde{z}_u, \quad \forall \tilde{z}_u \in L^2(\Omega). \tag{43e}$$

Above we have introduced the variations

$$\begin{aligned}
\mathcal{L}_{p,u}(\tilde{p}, \hat{u}) &= -\int_\Omega m\, \boldsymbol{\nabla}\hat{u} \cdot \boldsymbol{\nabla}\tilde{p}\, \mathrm{d}V, \\
\mathcal{L}_{p,m}(\tilde{p}, \hat{m}) &= -\int_\Omega \hat{m}\, \boldsymbol{\nabla}u \cdot \boldsymbol{\nabla}\tilde{p}\, \mathrm{d}V, \\
\mathcal{L}_{u,u}(\tilde{u}, \hat{u}) &= \int_\Omega \tilde{u}\, \hat{u}\, \mathrm{d}V, \\
\mathcal{L}_{u,m}(\tilde{u}, \hat{m}) &= -\int_\Omega \hat{m}\, \boldsymbol{\nabla}\tilde{u} \cdot \boldsymbol{\nabla}p\, \mathrm{d}V, \\
\mathcal{L}_{m,m}(\tilde{m}, \hat{m}) &= \int_\Omega R(\tilde{m}, \hat{m})\, \mathrm{d}V, \\
\mathcal{L}_{m,z_\ell}(\tilde{m}, \hat{z}_\ell) &= -\int_\Omega \tilde{m}\, \hat{z}_\ell\, \mathrm{d}V, \\
\mathcal{L}_{m,z_u}(\tilde{m}, \hat{z}_u) &= \int_\Omega \tilde{m}\, \hat{z}_u\, \mathrm{d}V.
\end{aligned}$$

We do not proceed with the the development of the ssN framework for this example as it is identical to that of the first example. We do note that the second order adjoint equations are not independent of u, m, p and as such the elements of the ssN system matrix $\boldsymbol{H}$, need to be updated at each ssN iterate.

4 Software Implementation

The goal of this section is to describe how one can utilize the modular finite element methods `MFEM` library in order solve the example problems by the ssN

method as previously detailed. A `ReducedHessianOperator` class is developed to inherit from the `MFEM Operator` class. Furthermore, our application code specifies:

- `MFEM FiniteElementSpace` – pointer data for the discretization of both the state and parameter fields,
- `MFEM GridFunction` – data for u_d, f, m_ℓ, m_u, the 'observation' data, rhs state equation forcing term, lower and upper bounds for the parameter field, respectively,
- `MFEM Array<int>` – which specifies which discretized state boundary degrees of freedom will a Dirichlet condition apply to,
- `double` – value to specify γ the strength of the L^2 regularization,

and contains member functions

- `void set_forward_parameters` – which sets the parameter `GridFunction` member data from an application code `GridFunction`, and then updates the `LinearForm` state equation member data,
- `void state_solve` – solves the state equation, using `PCG` with a `GSSmoother` and writes the solution data to a state `GridFunction` member data,
- `double eval_cost()` – returns the value of the cost functional for the given state and parameter `GridFunction` member data,
- `void set_adj_parameters` – which sets the `LinearForm` adjoint equation member data from the state and parameter `GridFunction` member data,
- `void adj_solve` – solves the adjoint equation, using `PCG` with a `GSSmoother` and writes the solution data to an adjoint `GridFunction` member data,
- `void eval_grad` – for the given state, parameter and adjoint `GridFunction` member data, evaluate the gradient as a `GridFunction` over the parameter `FiniteElementSpace`,
- `void set_incremental_parameters` – set `MixedBilinear` form and associated `SparseMatrix` member data that implicitly define the Hessian,

- `void set_active_set` – compute `Array <int>` member data that will mark which discretized parameter degrees of freedom are active and inactive,
- `void set_rhs` – writes $-\left(\boldsymbol{g} + \boldsymbol{M}\boldsymbol{P}_{\mathcal{I}}^{\top}\boldsymbol{\lambda}_{\mathcal{I}}\right)$ to a pointer to `Vector` passed as argument from the application code,
- `void reduce_rhs` – alter the application Newton system rhs `Vector` in a manner similar to that which is employed in the `MFEM::ConstrainedOperator` class,
- `void set_mhat_initial` – alter the application code initial `Vector` pointer iterate for the Newton conjugate-gradient solve, again in a manner similar to that which is employed in the `MFEM::ConstrianedOperator`,
- `void project_update` – alter $\hat{m}$ in order that when it is added to the parameter `GridFunction` member data, the result is feasible,
- `void z_solve` – backsolve to update the z_ℓ, z_u `GridFunction` Lagrange multipliers for the lower and upper bound constraints,
- `bool is_primal_feasible` – check if the parameter member data is feasible,
- `bool is_dual_feasible` – check if the Lagrange multipliers for the lower and upper bound constraints are dual feasible.

5 Numerical Results

5.1 Example 1: Determination of a source term in Poisson's equation

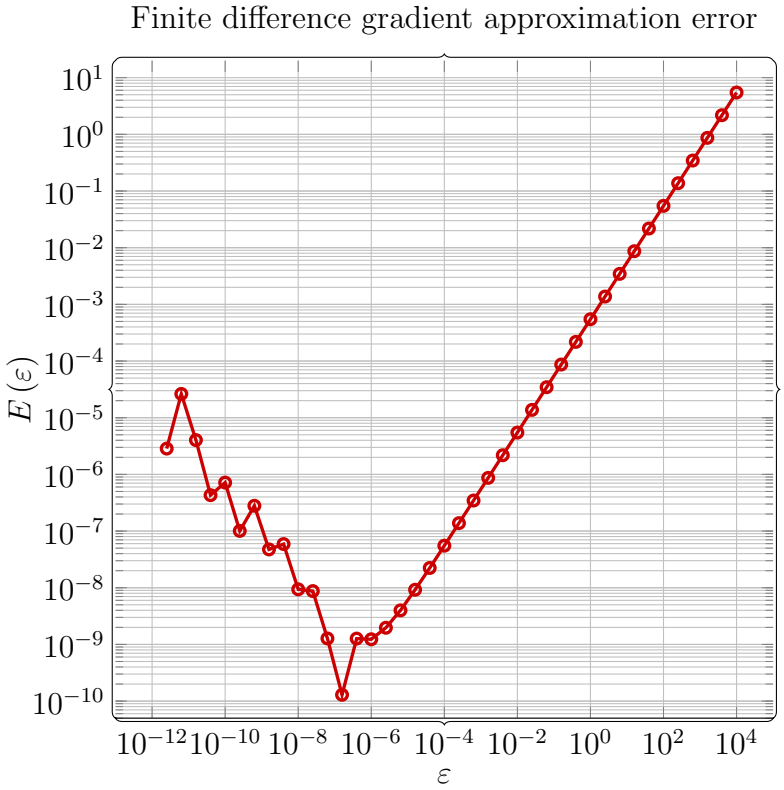
We first verify that the adjoint-based gradient and Hessian apply routines are correctly implemented, by means of a finite difference check.

5.1.1 Adjoint-based Gradient and Hessian Apply Verification

The consistency of the discretized cost functional evaluation and adjoint-based gradient are verified by computing the finite difference approximation error of the gradient $\boldsymbol{g}$ at $\boldsymbol{m}_0$ in direction $\boldsymbol{m}_1$

$$E(\varepsilon) := \left| \frac{J(\boldsymbol{m}_0 + \varepsilon \boldsymbol{m}_1) - J(\boldsymbol{m}_0)}{\varepsilon} - \boldsymbol{m}_1^\top \boldsymbol{g}(\boldsymbol{m}_0) \right|, \tag{44}$$

which with infinite-precision arithmetic is $\mathcal{O}(\varepsilon)$.



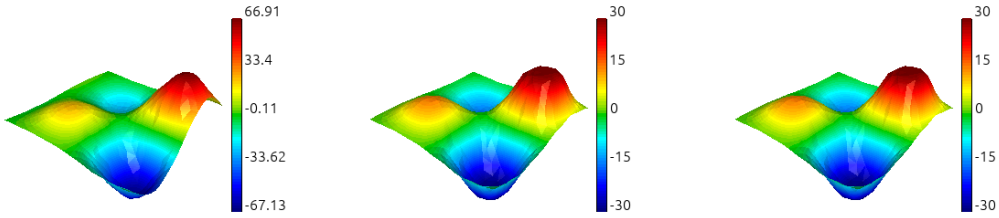
Similar tests were performed to verify the Hessian apply implemented.

5.1.2 Problem setup and solution estimation

The state and adjoint variables u, p, are here discretized by linear $H^1(\Omega)$ elements and the parameter m, by piecewise constant $L^2(\Omega)$ elements. For simplicity, the Dirichlet condition is chosen to be consistent with the data set u_d. The PDE inverse problem is regularized with the inclusion of a L^2 regularization term in the cost functional, with $\gamma = 10^{-4}$ as the regularization parameter. The lower and upper bounds m_ℓ, m_u are taken

to be the constants $m_\ell = -30$, $m_u = 30$ and observation data set of $u_d(x, y) = \sin(2\pi x)\sin(2\pi y)\exp(2x)/6$.

On a coarse unit square mesh, with 968 degrees of freedom for the discretized parameter field and 525 degrees of freedom for the discretized state the following solution estimate to the inverse with bound constraints is obtained



(1) True Parameter | (2) Projected True Parameter | (3) Parameter Reconstruction

The true parameter $-\Delta u_d$, is that which gives rise to the data u_d. As illustrated above the true parameter is not contained within the lower and upper bounds m_ℓ and m_u. To assess the quality of the parameter estimate, the true parameter is projected onto the feasible region and compared to the estimate. The parameter reconstruction agrees well with the projected true parameter.

5.1.3 Convergence Results

Table 1 demonstrates that the optimization algorithm is consistent with the infinite-dimensional formulation, in that the number of ssN updates needed to converge to a KKT point is largely independent of the level of mesh refinement.

$\dim(\boldsymbol{m})$	ssN iters	$\dim(\boldsymbol{\mathcal{A}}_\ell)$	$\dim(\boldsymbol{\mathcal{A}}_\ell)/\dim(\boldsymbol{m})$
968	4	28	0.0289
3872	3	117	0.0302
15488	4	480	0.0310
61952	4	1930	0.0312

Table 1: ssN iterations needed to reach KKT point and relative size of active set at KKT point upon mesh refinement.

Table 2 provides numerical evidence that the ssN iterates converge superlinearly to a KKT point.

k	$\|\|m^{(k)} - \bar{m}\|\|_{L^2(\Omega)}$	$\|\|m^{(k)} - \bar{m}\|\|_{L^2(\Omega)} / \|\|m^{(k-1)} - \bar{m}\|\|_{L^2(\Omega)}$
0	1.360×10^{1}	-
1	3.503×10^{-1}	2.572×10^{-2}
2	8.299×10^{-3}	2.369×10^{-2}
3	7.446×10^{-6}	8.973×10^{-4}
4	5.327×10^{-11}	7.153×10^{-6}

Table 2: Convergence behavior of the ssN iterates, 15488 degrees of freedom of the discretized parameter field m.

5.2 Example 2: Determination of a diffusion coefficient in Poisson's equation

This problem has features that make solution computation by the ssN method more challenging than in the former example problem. Nonconvexity demands that we use methods such as Armijo backtracking linesearch in order that the scheme be globalized. The guarantee that the solution of the ssN system is a descent direction, is not present in this nonconvex problem and thus the ssN solution is not suitable for use in linesearch. Yet another challenging feature is that the forward state equation is not well-posed for each $m \in L^2(\Omega)$, which is precisely what motivated the study of how to ensure primal feasibility in the former problem. The traditional ssN method will

in general, create infeasible iterates and hence the algorithm was modified by means of projections onto the feasible set. We took note of more sophisticated projection techniques [13], but were concerned of an increase of iterations caused by using a search direction that is a linear combination of the ssN and gradient directions and potentially nearly parallel with the gradient. Other works have also discussed the challenges of applying ssN to this problem, [6] chose the parameter lower bound sufficiently far away from 0 but as the authors note this does not guarantee that all ssN parameter iterates will generate a well-posed state equation.

After numerically observing that the ssN direction provided by $\hat{\boldsymbol{m}}, \hat{\boldsymbol{z}}_\ell, \hat{\boldsymbol{z}}_u$ was not a descent direction for the discretized objective functional J, we considered a common merit function [10, 8, 5] but designed to be consistent with the infinite dimensional problem, that is

$$\Theta = \frac{1}{2}\int_\Omega \left(\left(\frac{\delta \mathcal{L}}{\delta m} \right)^2 + \phi\left(m - m_\ell, z_\ell\right)^2 + \phi\left(m_u - m, z_u\right)^2 \right) \mathrm{d}V, \tag{45}$$

where $\frac{\delta \mathcal{L}}{\delta m}$ is the variational derivative [4] of the Lagrangian functional with respect to m

$$\frac{\delta \mathcal{L}}{\delta m} = \gamma\, m - \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} p - (z_\ell - z_u)\,. \tag{46}$$

It was observed numerically that the ssN direction was not always a descent direction for the merit functional Θ. Having not obtained a merit functional where each projected ssN direction is a descent direction, we were then unable to globalize the method in order to estimate KKT points for this problem.

6 Conclusions

One limitation of this framework is the approximation $\boldsymbol{P}_{\boldsymbol{\mathcal{I}}} \boldsymbol{M} \boldsymbol{P}_{\boldsymbol{\mathcal{A}}}{}^\top = \boldsymbol{0}$ we have enforced. The error of this decoupling approximation increases as the order of the finite element discretization of m and the number of nonzeros per row of the mass matrix $\boldsymbol{M}$ increases. In this work, we have used the nodal values of the finite element approximations of the fields u, m, p, z_ℓ, z_u at each ssN iteration to compute the respective active and inactive sets $\boldsymbol{\mathcal{A}}, \boldsymbol{\mathcal{I}}$ and

then employed the decoupling approximation. Alternatively, one could use the finite element approximants to compute the boundary between the active and inactive sets $\Omega_{\mathcal{A}}, \Omega_{\mathcal{I}}$, which will, in general, not conform to the finite element discretization. After determining the active, inactive set boundary the domain Ω, could be rediscretized in order that the finite element discretization utilized conforms to the active, inactive set boundary. To the best of our knowledge, `MFEM` does not currently offer API to implement such a task. We are uncertain if this change would make the resulting algorithm mesh dependent. The decoupling approximation allowed for the ssN updates to be obtained by solving a linear system by the conjugate gradient algorithm, we however did not incorporate a preconditioner such as multigrid which has been developed for similar problems [3, 2].

We found `MFEM` to be a user-friendly productive software environment. `MFEM` allowed for the utilization of many numerical subroutines needed for an implementation of this computational framework. Examples of such software features are the infrastructure to create a custom `Operator`, defined only by its action on vectors, and to pass the resulting object to a Krylov subspace `IterativeSolver`. This allows for the determination of linear system solutions, where the `Operator` object implicitly defines the system matrix. Our numerical results lead us to conclude that more work is needed to globalize the ssN method. Globalization is necessary in order that the ssN method can be effectively used to estimate solutions of topology optimization problems, in which, the state equation is not well defined for densities outside of the feasible region.

Acknowledgements

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] R. Anderson, J. Andrej, A. Barker, J. Bramwell, J.-S. Camier, J. Cerveny, V. Dobrev, Y. Dudouit, A. Fisher,

T. Kolev, et al., *MFEM: a modular finite element methods library*, Computers & Mathematics with Applications, (2020).

[2] A. T. Barker and A. Drăgănescu, *Algebraic multigrid preconditioning of the hessian in optimization constrained by a partial differential equation*, Numerical Linear Algebra with Applications, 28 (2021), p. e2333.

[3] A. Drăgănescu, *Multigrid preconditioning of linear systems for semismooth Newton methods applied to optimization problems constrained by smoothing operators*, Optimization Methods and Software, 29 (2014), pp. 786–818.

[4] I. Gelfand and S. Fomin, *Variational calculus*, Moscow, Fizmatgiz, (1961).

[5] M. Gerdts, S. Horn, and S.-J. Kimmerle, *Line search globalization of a semismooth Newton method for operator equations in Hilbert spaces with applications in optimal control*, Journal of Industrial & Management Optimization, 13 (2017), p. 47.

[6] E. Hans, *Globally convergent B-semismooth Newton methods for ℓ_1-Tikhonov regularization*, PhD thesis, Johannes Gutenberg-Universität Mainz, 2017.

[7] M. Hintermüller, K. Ito, and K. Kunisch, *The primal-dual active set strategy as a semismooth Newton method*, SIAM Journal on Optimization, 13 (2002), pp. 865–888.

[8] K. Ito and K. Kunisch, *On a semi-smooth Newton method and its globalization*, Mathematical Programming, 118 (2009), pp. 347–370.

[9] B. S. Lazarov and O. Sigmund, *Filters in topology optimization based on Helmholtz-type differential equations*, International Journal for Numerical Methods in Engineering, 86 (2011), pp. 765–781.

[10] J. Nocedal and S. Wright, *Numerical optimization*, Springer Science & Business Media, 2006.

[11] N. Petra and G. Stadler, *Model variational inverse problems governed by partial differential equations*, tech. rep., The Institute For Computational Engineering And Sciences, The University of Texas at Austin, 2011.

[12] G. Stadler, *Elliptic optimal control problems with L^1-control cost and applications for the placement of control devices*, Computational Optimization and Applications, 44 (2009), p. 159.

[13] D. Sun, R. S. Womersley, and H. Qi, *A feasible semismooth asymptotically Newton method for mixed complementarity problems*, Mathematical Programming, 94 (2002), pp. 167–187.