ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

AN INVERSE PROBLEM STRATEGY BASED ON FORWARD MODEL EVALUATIONS: GRADIENT-BASED OPTIMIZATION WITHOUT ADJOINT SOLVES

Miguel A. Aguiló

Computational Solid Mechanics and Structural Dynamics
Sandia National Laboratories, PO Box 5800, MS 0380 Albuquerque, NM 87185-0380, USA
e-mail: maguilo@sandia.gov

Keywords: Inverse problem, Compliance error minimization, Gradient-based optimization, Newton's method, Nonlinear conjugate gradient, Heat transfer.

Abstract. *A new inverse problem formulation based on the compliance error functional is presented. The proposed formulation permits the derivation of an explicit expression for the Lagrange multipliers. Thus, the computation of the Lagrange multipliers does not require the solution of the computationally intensive adjoint system of equations. This enables the computation of the first order derivative information at the expense of just one model evaluation. Leading to significant speedups for large-scale, gradient-based inverse problems.*

Second, if second order optimization algorithms are available, Newton's method can be applied to the first order necessary optimality conditions. Newton's method relies on second order derivative information to compute descent directions during optimization. This paper presents two Hessian formulations based on the compliance error functional. The first formulation relies on the mathematical properties of the compliance error functional to compute the application of the trial step to the analytical Hessian at the expense of one model evaluation per Newton iteration. The second formulation is based on a Gauss-Newton approximation to the Hessian. The Gauss-Newton approximation further speedups analysis since the second order derivative information is computed without performing additional model evaluations during the Newton iterations.

Third, examples in heat transfer are presented to demonstrate the effectiveness of the compliance error functional. The compliance error minimization formulation is compared against the data misfit minimization formulation. Results will show that the compliance error minimization formulation outperforms the data misfit formulation.

1 INTRODUCTION

This paper investigates the problem of large-scale parameter estimation, i.e. inverse problem, in the context of a particular problem: determining the thermal conductivity properties of a region of interest given an 'observed' temperature field. Specifically, a new inverse problem formulation based on the compliance error functional is presented. A general inverse problem framework for the application of first and second order optimization algorithms is presented, The proposed framework enables the implementation of the compliance error minimization formulation to many applications of interests in science and engineering.

Multiple inverse problem techniques have been proposed in the literature for the solution of inverse problems. For instance, many researchers have applied data misfit error functionals for the solution of inverse problems in many physics settings [1, 2, 3, 4]. The objective of data misfit functional is to characterize the parameter of interest by minimizing the discrepancy between the simulation and the experimental data (target/observed data). An alternative to data misfit functionals is the modified error in the constitutive equation (MECE) functional. The objective of the MECE functional is to characterize the parameter of interests by minimizing the discrepancy in the constitutive equations [5, 6, 7, 8, 9, 10, 11, 12]. This formulation leads to a coupled system of equations that is solved for any new set of materiel parameters.

Allix *et al.* [13] performed several numerical studies that demonstrated that the MECE functional improved the convexity of the objective function. Furthermore, Gockenbach *et al.* [14, 15, 16] showed that the energy norm term of the MECE functional is convex for elliptic boundary value problems when full field measurements are available. However, in a subsequent study, Gockenbach [17] showed that inverse problem formulations based on energy norm minimization can lead to inaccurate estimates if the 'observed' data is corrupted. Moreover, he showed that the data misfit functional was less sensitive to data corruption than the energy norm functional. Lastly, the virtual field method (VFM) is a recent inverse problem technique developed for extracting constitutive parameters from full-field measurements [18]. The VFM combines the principle of virtual work with kinematically admissible virtual fields, which are computed *a-priori*, to characterize the constitutive parameters. The choice of the kinematically admissible virtual field is key in order to improve the performance of the VFM. Thus, most research is focus on improving the predictive reliability of the kinematically admissible virtual fields [19].

In this work, a new inverse problem formulation based on the compliance error functional is presented. The main advantage of the compliance error functional is that it enables the computation of the Lagrange multipliers, and thus the first order derivative information, at the expense of just one model evaluation. Thus, the calculation of the Lagrange multipliers does not require the solution of the computationally intensive adjoint system of equations. This leads to significant speedups for large-scale, gradient-based inverse problems since the gradient computation is only based on forward model evaluations. The proposed formulation also simplifies implementation in production software since only forward model evaluations are needed during optimization.

If second order optimization algorithms are available, Newton's method can be applied to the first order necessary optimality conditions. Newton's method relies on second order derivative information to compute descent directions during optimization. If the data misfit functional is used, two model evaluations are required per Newton iteration to compute the second order derivative information. This paper presents two Hessian formulations based on the compliance error functional. The first formulation relies on the mathematical properties of the compliance

error functional to reduce the number of model evaluations from two to one without sacrificing accuracy. Consequently, substantial speedups are gained. The second formulation relies on a Gauss-Newton Hessian approximation based on the compliance error functional to further speedup the Newton solver. By using a Gauss-Newton approximation, the second order derivative information is computed without performing additional model evaluations during the Newton iterations. Results will demonstrate that the compliance error functional leads to substantial speedups over the the data misfit functional.

This paper is organized as follows: Section 2 presents the general first and second order inverse problem formulations for the compliance error and data misfit functionals. Furthermore, the respective general inverse problem solution framework is presented for both formulations. Section 3 presents an inverse problem example in heat transfer to showcase the performance of the compliance error functional. Finally, Section 4 provides concluding remarks and future research directions.

2 FORMULATION

2.1 Inverse problem

Let $\Omega \subseteq \mathbb{R}^d$, $d \in \{1, 2, 3\}$ denote the computational domain with boundary $\partial\Omega$. Lets now define the Lebesgue space $\mathbb{H} = L^2(\Omega; \mathbb{R}^{\mathsf{n}})$ of measurable and square intregrable functions endowed with inner product $\langle \phi, \psi \rangle_{\mathbb{H}} = \int_\Omega \phi\, \psi$ for $\phi, \psi \in \mathbb{H}$ and norm $\|\phi\|_{\mathbb{H}} = \langle \phi, \phi \rangle_{\mathbb{H}}^{1/2}$. Lets also define finite dimensional spaces $\mathbb{U} = \{\text{span}\{\phi^a\}_{a=1}^A \,|\, \phi \in \mathbb{H}\} \subset U$, $\mathbb{Z} = \{\text{span}\{\psi^b\}_{b=1}^B \,|\, \psi \in \mathbb{H}\} \subset Z$, and $\mathbb{Y} = \{\text{span}\{\chi^c\}_{c=1}^C \,|\, \chi \in \mathbb{H}\} \subset Y$. This enables then the following finite dimensional approximations for the state, control, and Lagrange multipliers $\mathbf{u} = \sum_a^A \tilde{u}^a \phi^a \,|\, \tilde{u} \in \mathbb{R}$, $\mathbf{z} = \sum_b^B \tilde{z}^b \psi^b \,|\, \tilde{z} \in \mathbb{R}$, and $\mathbf{v} = \sum_c^C \tilde{v}^c \chi^c \,|\, \tilde{v} \in \mathbb{R}$, respectively.

Lets now define a general parameter estimation (inverse) problem as

$$\begin{aligned} \min_{(\mathbf{u},\mathbf{z}) \in \mathbb{U} \times \mathbb{Z}} \quad & \mathsf{J}(\mathbf{u}, \mathbf{z}) \\ \text{s.t.} \quad & \\ & \mathsf{g}(\mathbf{u}, \mathbf{z}) = 0, \end{aligned} \tag{1}$$

where $\mathbf{u}$ and $\mathbf{z}$ respectively denote the state and control variables, $\mathsf{J}(\mathbf{u}, \mathbf{z})\colon \mathbb{U} \times \mathbb{Z} \to \mathbb{R}$ denotes the objective function and $\mathsf{g}(\mathbf{u}, \mathbf{z})\colon \mathbb{U} \times \mathbb{Z} \to \mathbb{Y}$ denotes the equality constraint (physics model).

The implicit function theorem admits the definition of a solution operator $\mathbf{u}\colon \mathbb{Z} \to \mathbb{U}$ such that $\{(\mathbf{u}(\mathbf{z}), \mathbf{z}) \mid \mathbf{z} \in \mathbb{Z}\} = \{(\mathbf{u}, \mathbf{z}) \in \mathbb{U} \times \mathbb{Z} \mid \mathsf{J}(\mathbf{u}, \mathbf{z}) = 0\}$. This enables the redefinition of Equation 1 as

$$\min_{\mathbf{z} \in \mathbb{Z}} \quad \mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z}), \tag{2}$$

where the solution operator $\mathbf{u}(\mathbf{z})$ is obtained by solving $\mathsf{g}(\mathbf{u}(\mathbf{z}), \mathbf{z}) = 0$. This formulation is commonly known as the reduced-space formulation for partial differential equation (PDE) constrained optimization.

Finally, lets assume that the objective function and equality constraint are given by

$$\mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z}) = \frac{\beta}{2} \| \langle \mathbf{u}(\mathbf{z}), \mathbf{A}(\mathbf{z})\mathbf{u}(\mathbf{z}) \rangle_{\mathbb{H}} - \langle \hat{\mathbf{u}}, \mathbf{A}(\mathbf{z})\hat{\mathbf{u}} \rangle_{\mathbb{H}} \|_{\mathbb{H}}^2 + \mathsf{R}(\mathbf{z}), \tag{3}$$

$$\mathsf{g}(\mathbf{u}(\mathbf{z}), \mathbf{z}) = \mathbf{A}(\mathbf{z})\mathbf{u} - \mathbf{f} = 0, \tag{4}$$

where $\mathbf{A}(\mathbf{z})\colon \mathbb{Z} \to \mathbb{U} \times \mathbb{U}$ is a non-singular, self-adjoint linear operator, $\mathsf{R}(\mathbf{z})\colon \mathbb{Z} \to \mathbb{R}$ is a regularization functional, $\mathbf{f} \in \mathbb{Y}$ is an external force and $\hat{\mathbf{u}} \in \hat{\Omega} \subset \Omega$ denotes measured data. This formulation will be referred as the compliance error minimization (CEM) formulation.

2.2 First order formulation

2.2.1 Data misfit functional

Lets define the Lagrangian functional $\mathcal{L}\colon \mathbb{U} \times \mathbb{Z} \times \mathbb{Y} \to \mathbb{R}$ for Equation 2 as

$$\mathcal{L}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v}) = \mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z}) + \langle \mathbf{v}, \mathsf{g}(\mathbf{u}(\mathbf{z}), \mathbf{z}) \rangle_{\mathbb{Y}^*, \mathbb{Y}}, \tag{5}$$

where $\mathbf{v}$ denotes the Lagrange multipliers and $\mathbb{Y}^*$ is the dual space of $\mathbb{Y}$. The data misfit objective function is given by

$$\mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z}) = \frac{1}{2} \|\mathbf{u}(\mathbf{z}) - \hat{\mathbf{u}}\|^2_{\mathbb{H}}. \tag{6}$$

The equality constraint $\mathsf{g}(\mathbf{u}(\mathbf{z}), \mathbf{z})$ is given by Equation 4. If $\mathbf{z}^\star \in \mathbb{Z}$ is a local solution of Equation 2; then, there exists a set of Lagrange multipliers $\mathbf{v}^\star \in \mathbb{Y}$ such that the first order necessary optimality conditions are satisfied at $\mathbf{z}^\star$.

The first order necessary optimality conditions are given by

$$\mathcal{L}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v}) = \mathsf{J}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z}) + \mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^* \mathbf{v} = 0 \tag{7}$$

$$\mathcal{L}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v}) = \mathsf{J}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}), \mathbf{z}) + \mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^* \mathbf{v} = 0, \tag{8}$$

where the subscripts $\mathbf{u}$ and $\mathbf{z}$ respectively denote derivatives with respect to the state and control variables. The Lagrange multipliers are computed by solving

$$\mathbf{v} = -(\mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^*)^{-1} \mathsf{J}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z}). \tag{9}$$

Substituting Equation 9 into Equation 8 yields a reduced gradient operator of the form

$$\nabla \mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z}) = \mathsf{J}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}), \mathbf{z}) + \mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^* [-(\mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^*)^{-1} \mathsf{J}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z})]. \tag{10}$$

At each optimization iteration, the following sequence of steps are performed to compute the reduced gradient operator and minimize the objective function:

1. Solve equality $\mathsf{g}(\mathbf{u}(\mathbf{z}), \mathbf{z}) = 0$ for $\mathbf{u} \in \mathbb{U}$;
2. Solve $\mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^* \mathbf{v} = -\mathsf{J}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z})$ for $\mathbf{v} \in \mathbb{Y}$;
3. Compute the reduced gradient operator defined in Equation 10;
4. Compute descent direction $\mathbf{s} \in \mathbb{Z}$ and set $\mathbf{z}_{k+1} = \mathbf{z}_k + \gamma \mathbf{s}_k$, $\gamma \in \mathbb{R}$.

This sequence of steps are necessary to solve an inverse problem based on the data misfit functional [20].

2.2.2 Compliance error functional

Assume that the objective function and equality constraint for the parameter estimation problem defined in Equation 2 are given by Equations 3 and 4, respectively. Then, the first order derivative operators $\mathsf{g}_{\mathbf{u}}(\mathbf{u},\mathbf{z})^*$ and $\mathsf{J}_{\mathbf{u}}(\mathbf{u},\mathbf{z})$ are given by

$$\mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^* = \mathbf{A}(\mathbf{z})^* \tag{11}$$

and

$$\mathsf{J}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z}) = \alpha\mathbf{A}(\mathbf{z})\mathbf{u}, \tag{12}$$

where

$$\alpha = 2(\langle\mathbf{u}(\mathbf{z}),\mathbf{A}(\mathbf{z})\mathbf{u}(\mathbf{z})\rangle_{\mathbb{H}} - \langle\hat{\mathbf{u}},\mathbf{A}(\mathbf{z})\hat{\mathbf{u}}\rangle_{\mathbb{H}}). \tag{13}$$

Recall that $\mathbf{A}(\mathbf{z})$ is assumed to be a non-singular, self-adjoint linear operator. Thus, an explicit expression for the Lagrange multipliers can be derived by substituting Equations 11 and 12 into Equation 9. After substitution, the Lagrange multipliers are given by

$$\mathbf{v} = -\alpha\beta\mathbf{A}(\mathbf{z})^{-1}(\mathbf{A}(\mathbf{z})\mathbf{u}(\mathbf{z})) = -\alpha\beta\mathbf{I}\mathbf{u}(\mathbf{z}) = -\alpha\beta\mathbf{u}(\mathbf{z}), \tag{14}$$

where $\mathbf{I}$ denotes the identity linear operator. Substituting Equation 14 into Equation 10 gives a reduced gradient operator of the form

$$\nabla_{\mathbf{z}}\mathsf{J}(\mathbf{u}(\mathbf{z}),\mathbf{z}) = \mathsf{J}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z}) - \alpha\beta\langle\mathbf{u}(\mathbf{z}),\mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\rangle. \tag{15}$$

Notice that the adjoint problem defined in Equation 9 is not solved to compute the Lagrange multipliers, which should speedup the optimization problem.

At each optimization iteration, the following sequence of steps are performed to compute the reduced gradient operator and minimize the compliance error objective function:

1. Solve equality $\mathsf{g}(\mathbf{u}(\mathbf{z}),\mathbf{z}) = 0$ for $\mathbf{u} \in \mathbb{U}$;
2. Compute Lagrange multipliers $\mathbf{v} = -\alpha\beta\mathbf{u}(\mathbf{z})$;
3. Compute the reduced gradient operator given by Equation 15.
4. Compute descent direction $\mathbf{s} \in \mathbb{Z}$ and set $\mathbf{z}_{k+1} = \mathbf{z}_k + \gamma\mathbf{s}_k$, $\gamma \in \mathbb{R}$.

In addition to the speedups gained due to the omission of the adjoint model evaluations during optimization, the CEM formulation facilitates implementation since it only relies on forward model evaluations to compute the reduced gradient operator.

2.3 Second order formulation

2.3.1 Data misfit functional

If second order derivative optimization algorithms are available, Newton's method can be applied to the first order necessary optimality conditions. Let $\kappa \in \mathbb{R}^*_+$ and $\delta\mathbf{z} \in \mathbb{Z}$, if $\mathbf{z}^\star \in \mathbb{Z}$ satisfy the first order necessary optimality conditions and

$$\langle\delta\mathbf{z},\nabla^2\mathsf{J}(\mathbf{u}(\mathbf{z}^\star)\mathbf{z}^\star)\delta\mathbf{z}\rangle \geq \kappa\|\delta\mathbf{z}\|^2_{\mathbb{H}} \;\forall\; \delta\mathbf{z} \in \ \ker\mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}^\star),\mathbf{z}^\star),$$

the second-order sufficient condition is satisfied at $\mathbf{z}^\star$. Furthermore, $\mathbf{z}^\star$ is a strict local minimum of Equation 2.

The application of the trial step $\delta\mathbf{z}$ to the Hessian operator is given by

$$\nabla^2 \mathsf{J}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = \mathcal{L}_{\mathbf{zu}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{u} + \mathcal{L}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{z} + \mathcal{L}_{\mathbf{zv}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{v}, \tag{16}$$

for $\delta\mathbf{u} \in \mathbb{U}$ and $\delta\mathbf{v} \in \mathbb{Y}$. Notice that $\delta\mathbf{u}$ and $\delta\mathbf{v}$ are necessary to compute the application of the trial step to the Hessian operator. Thus, explicit expressions for $\delta\mathbf{u}$ and $\delta\mathbf{v}$ are necessary to calculate Equation 16.

Let $\mathsf{g}(\mathbf{u}(\mathbf{z}),\mathbf{z}) = 0 \; \forall \; \mathbf{z} \in \mathbb{Z}$. Then, $\mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = 0 \; \forall \; (\mathbf{z},\delta\mathbf{z}) \in \mathbb{Z} \times \mathbb{Z}$, where

$$\mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = \mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{u} + \mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = 0 \tag{17}$$

and $\delta\mathbf{u} \equiv \mathbf{u}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z}$. Solving Equation 17 for $\delta\mathbf{u}$ gives

$$\delta\mathbf{u} = -\mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^{-1}\mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z}. \tag{18}$$

Next, an explicit expression is derived for $\delta\mathbf{v}$. By definition, $\mathcal{L}_{\mathbf{u}}(\hat{\mathbf{u}}(\mathbf{z}),\mathbf{z},\mathbf{v}) = 0 \; \forall \; (\mathbf{u},\mathbf{z},\mathbf{v}) \in \mathbb{U} \times \mathbb{Z} \times \mathbb{Y}$; thus, the derivative of $\mathcal{L}_{\mathbf{u}}(\hat{\mathbf{u}}(\mathbf{z}),\mathbf{z},\mathbf{v})$ in the direction of $\delta\mathbf{z}$ gives

$$\mathcal{L}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{u} + \mathcal{L}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{z} + \mathcal{L}_{\mathbf{uv}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{v} = 0, \tag{19}$$

$\forall \quad (\mathbf{z},\delta\mathbf{u},\delta\mathbf{z},\delta\mathbf{v}) \in \mathbb{Z} \times \mathbb{U} \times \mathbb{Z} \times \mathbb{Y}$. Solving Equation 19 for $\delta\mathbf{v}$ gives

$$\delta\mathbf{v} = -\mathcal{L}_{\mathbf{uv}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})^{-1}[\mathcal{L}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{u} + \mathcal{L}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{z}], \tag{20}$$

where

$$\mathcal{L}_{\mathbf{uv}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v}) = \mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^* \tag{21}$$

$$\mathcal{L}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v}) = \mathsf{J}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z}) + \mathsf{g}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\mathbf{v} \tag{22}$$

$$\mathcal{L}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v}) = \mathsf{J}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z}) + \mathsf{g}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\mathbf{v}. \tag{23}$$

The following sequence of steps are performed to compute the application of the trial step $\delta\mathbf{z}$ to the analytical Hessian operator at each Newton iteration:

1. Solve $\mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{u} = -\mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z}$ for $\delta\mathbf{u} \in \mathbb{U}$
2. Solve $\mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\delta\mathbf{v} = -[\mathcal{L}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{u} + \mathcal{L}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{z}]$ for $\delta\mathbf{v} \in \mathbb{Y}$
3. Compute the application of the trial step to the reduced Hessian operator

$$\nabla^2 \mathsf{J}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = \mathcal{L}_{\mathbf{zu}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{u} + \mathcal{L}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{z} + \mathcal{L}_{\mathbf{zv}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{v}, \tag{24}$$

where $\mathcal{L}_{\mathbf{zv}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v}) = \mathsf{g}_{\mathbf{z}}(\hat{\mathbf{u}}(\mathbf{z}),\mathbf{z})^*$.

Notice that the data misfit functional requires two model evaluations per Newton iteration to compute the application of the trial step to the analytical Hessian operator.

2.3.2 Compliance error functional: analytical Hessian

Assume that the objective function and equality constraint are given by Equations 3 and 4, respectively. Then, the second order derivative operators $\mathcal{L}_{\mathbf{uv}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v})$, $\mathcal{L}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v})$, and $\mathcal{L}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v})$ are given by

$$\mathcal{L}_{\mathbf{uv}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v}) = \mathbf{A}(\mathbf{z})^* \tag{25}$$

$$\mathcal{L}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v}) = 2\beta[\alpha\mathbf{A}(\mathbf{z})\delta\mathbf{u} + 2\mathbf{A}(\mathbf{z})\mathbf{u}(\mathbf{z})\langle\mathbf{u}(\mathbf{z}), \mathbf{A}(\mathbf{z})\delta\mathbf{u}\rangle_{\mathbb{H}}] \tag{26}$$

$$\mathcal{L}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v}) = 2\beta[\alpha(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z})\mathbf{u}(\mathbf{z}) + \gamma\mathbf{A}(\mathbf{z})\mathbf{u}(\mathbf{z})] + (\mathbf{A}(\mathbf{z})^*\mathbf{v})\delta\mathbf{u}, \tag{27}$$

where

$$\gamma = \langle\mathbf{u}(\mathbf{z}), (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z})\mathbf{u}(\mathbf{z})\rangle_{\mathbb{H}} - \langle\hat{\mathbf{u}}, (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z})\hat{\mathbf{u}}\rangle_{\mathbb{H}} \tag{28}$$

Recall that $\mathbf{A}(\mathbf{z})$ is a non-singular, self-adjoint linear operator. This enables the derivation of an explicit expression for $\delta\mathbf{v}$ by substituting Equations 14, 25, 26 and 27 into Equation 20. After simplification, $\delta\mathbf{v}$ is given by

$$\delta\mathbf{v} = -2\beta[\alpha\delta\mathbf{u} + 2\mathbf{u}(\mathbf{z})\langle\mathbf{u}(\mathbf{z}), \mathbf{A}(\mathbf{z})\delta\mathbf{u})\rangle_{\mathbb{H}} + \gamma\mathbf{u}(\mathbf{z})], \tag{29}$$

where $\delta\mathbf{u}$ is given by Equation 18.

To compute the application of the trial step to the analytical Hessian operator, the second order derivative operators $\mathcal{L}_{\mathbf{zu}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v})$, $\mathcal{L}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v})$ and $\mathcal{L}_{\mathbf{zv}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v})$ are required. These second order derivative operators are explicitly given by

$$\mathcal{L}_{\mathbf{zu}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v}) = 2\beta[(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u}(\mathbf{z}))\mathbf{u}(\mathbf{z}) - (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\hat{\mathbf{u}})\hat{\mathbf{u}}] \tag{30}$$

$$\mathcal{L}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v}) = \beta\gamma[(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u}(\mathbf{z}))\mathbf{u}(\mathbf{z}) - (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\hat{\mathbf{u}})\hat{\mathbf{u}}] + \mathsf{R}_{\mathbf{zz}}(\mathbf{z}) \tag{31}$$

$$\begin{aligned}\mathcal{L}_{\mathbf{zv}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v}) = -2\beta[&\alpha(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u}(\mathbf{z}))\delta\mathbf{u} + 2(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u}(\mathbf{z}))\mathbf{u}(\mathbf{z})\langle\mathbf{u}(\mathbf{z}), \mathbf{A}(\mathbf{z})\delta\mathbf{u}\rangle_{\mathbb{H}} \\ &+ \gamma(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u}(\mathbf{z}))\mathbf{u}(\mathbf{z})],\end{aligned} \tag{32}$$

where Equations 14 and 29 were used to derived and simplified Equations 30 and 32.

An explicit expression for the application of the trial step to the analytical Hessian operator is obtained by substituting Equations 30-32 into Equation 24. After some simplifications, the application of the trial step to the analytical Hessian operator is given by

$$\begin{aligned}\nabla^2\mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z})\delta\mathbf{z} = &-2\beta[(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u}(\mathbf{z}))\mathbf{u}(\mathbf{z}) + (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\hat{\mathbf{u}})\hat{\mathbf{u}}]\langle\mathbf{u}(\mathbf{z}), \mathbf{A}(\mathbf{z})\delta\mathbf{u}\rangle \\ &- \beta\gamma[(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u}(\mathbf{z}))\mathbf{u}(\mathbf{z}) + (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\hat{\mathbf{u}})\hat{\mathbf{u}}] + \mathsf{R}_{\mathbf{zz}}(\mathbf{z})\delta\mathbf{z} \\ &- 2\beta\alpha(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u}(\mathbf{z}))\delta\mathbf{u}.\end{aligned} \tag{33}$$

Therefore, the following sequence of steps are necessary to compute the application of the trial step $\delta\mathbf{z}$ to the analytical Hessian operator per Newton iteration:

1. Solve $\mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z})\delta\mathbf{u} = -\mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}), \mathbf{z})\delta\mathbf{z}$ for $\delta\mathbf{u} \in \mathbb{U}$
2. Compute $\nabla^2\mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z})\delta\mathbf{z}$ as defined by Equation 33.

The proposed compliance error minimization formulation enables the calculation of the second order derivative information at the expense of one model evaluation per Newton iteration. Contrary, the data misfit formulation presented in Section 2.3.1 requires two model evaluations per Newton iteration. Thus, significant speedups are possible with the CEM formulation by omitting one model evaluation per Newton iteration.

2.3.3 Compliance error functional: Gauss-Newton Hessian

To circumvent the additional computational demands associated with computing Equation 33, a Gauss-Newton Hessian formulation based on the compliance error functional is considered. Lets define the application of the trial step to the Gauss-Newton Hessian as

$$\nabla^2 J(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} \equiv \mathcal{L}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{z} = [J_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z}) + g_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\mathbf{v}]\delta\mathbf{z}. \tag{34}$$

Notice that the derivative operators $\mathcal{L}_{\mathbf{zu}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{u}$ and $\mathcal{L}_{\mathbf{zv}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{v}$ are omitted in Equation 34 since these nonlinear terms vanish. Thus, the solution of Equation 18 is not necessary to compute the application of the trail step to the Gauss-Newton Hessian during the Newton iterations. Results will demonstrate that the Gauss-Newton Hessian approximation based on the compliance error functional can lead to significant speedups without hindering solution accuracy.

3 EXAMPLE IN HEAT TRANSFER

The Intrepid PDE discretization package from Trilinos [21] was used to build the finite element models. The direct solver routine from MATLAB [22] scientific package were used to solve the linear system of equations in this numerical study. The optimization algorithms in this work were implemented in C++ and used to generate the results presented herein [23]. Readers are encourage to explore other optimization library of their preference [22, 24, 25, 26, 27]. Finally, all calculations were performed on a Linux workstation with a 2.93 GHz Intel(R) Core Xeon(R) processor and 24 GB of RAM.

To synthesize the 'observed' temperature field, a finer grid with 80,000 triangles was used to generate the experimental temperature field. The experimental temperature field was then projected onto a computational grid 20,000. This was done to avoid using the same computational mesh used to generate the 'observed' temperature field during optimization. Different levels of random Gaussian noise were also considered, $\Delta \sim N(0,\widehat{\sigma})$ in order to test the tolerance of the proposed formulation to corrupt data. A Gaussian distributed set of random numbers has 65%, 95%, and 99.7% certainty of respectively being within one, two, and three standard deviations from the mean. Lets thus assume that the Gaussian random noise generated to test the compliance error minimization formulation is 95% certain of being within $\theta\%$ of the actual data. Then, the perturbation applied to the 'observed' temperature field is scaled from interval $(-2\widehat{\sigma}, 2\widehat{\sigma})$ to $(-\theta\%, \theta\%)$. This produces a perturbation parameter of the form $\varepsilon^\theta = \frac{1}{2}(\frac{\theta}{100})$ [28]. Therefore, the corrupt data was generated as follows

$$\hat{\mathbf{u}}_i^\theta = \mathbf{u}_i(1+\varepsilon_i^\theta), \text{ for } i = 1,\ldots,n_{\mathbf{u}} \tag{35}$$

where $\theta \in \Theta = \{1\%, 3\%, 5\%\}$ and $n_{\mathbf{u}}$ denotes the number of states.

Finally, the compliance error minimization formulation was compared against data misfit formulation. The objective was to highlight the effectiveness of the CEM formulation versus a common inverse problem formulation strategy. For completeness, the corresponding first and second order derivative operators for the data misfit functional are defined herein

$$\mathsf{J}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z}) = \beta(\mathbf{u}(\mathbf{z}) - \hat{\mathbf{u}}) \tag{36}$$

$$\mathsf{J}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{u} = \beta\delta\mathbf{u} \tag{37}$$

$$\mathsf{J}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z}) = \mathsf{J}_{\mathbf{zu}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{u} = \mathsf{J}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = \mathsf{J}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = \mathbf{0} \tag{38}$$

3.1 Problem formulation

Lets consider the following parameter estimation (inverse) problem in heat transfer

$$\begin{aligned} \min_{\mathbf{z}\in\hat{\mathbb{Z}}} \quad & \frac{\beta}{2}\|\langle \mathbf{u}, \mathbf{A}(\mathbf{z})\mathbf{u}\rangle_{\mathbb{H}} - \langle \hat{\mathbf{u}}, \mathbf{A}(\mathbf{z})\hat{\mathbf{u}}\rangle_{\mathbb{H}}\|^2_{\mathbb{H}} + \mathsf{R}(\mathbf{z}) \\ \text{s.t.} \quad & \\ & \mathbf{A}(\mathbf{z})\mathbf{u} = \mathbf{f}(\mathbf{x}) \quad \text{in} \quad \Omega \\ & \mathbf{u} = 0 \qquad \text{on} \quad \partial\Omega, \end{aligned} \tag{39}$$

where $\hat{\mathbb{Z}} = \{\mathbf{z} \in \mathbb{Z} : \mathscr{L} \leq \mathbf{z} \leq \mathscr{U}\}$. Here, $\mathscr{L}$ denotes the control lower bounds and $\mathscr{U}$ denotes the control upper bounds. For a steady-state heat equation, $\mathbf{z}$ is the coefficient of thermal conductivity, $\mathbf{u}$ is the temperature field, $\hat{\mathbf{u}} \in \Omega^m \subseteq \Omega$ are the temperature measurements. $\mathbf{f}(\mathbf{x})$ is a heat source given by

$$\mathbf{f}(\mathbf{x}) = A * \sin(\omega\mathbf{x})\cos(\omega\mathbf{x}), \tag{40}$$

where $A \in \mathbb{R}$ is a given amplitude, $\omega \in \mathbb{R}$ denotes angular frequency and $\mathbf{x} \in \Omega$ denotes a position is space. $\mathbf{A}(\mathbf{z})\colon \mathbb{Z} \to \mathbb{U} \times \mathbb{U}$ is a linear operator that depends on the coefficient of thermal conductivity and $\beta \in \mathbb{R}_+$ denotes a penalty parameter. The finite dimensional approximations for the state, control, and Lagrange multipliers were previously defined in Section 2.1.

3.1.1 Regularization

The regularization functional $\mathsf{R}(\mathbf{z})\colon \mathbb{Z} \to \mathbb{R}$ in Equation 39 is given by

$$\mathsf{R}(\mathbf{z}) = \frac{\zeta}{2\theta}(\langle \nabla\mathbf{z}, \nabla\mathbf{z}\rangle_{\mathbb{H}} + \nu)^{\theta}, \tag{41}$$

where $\theta = -1/2$, $0 < \nu \leq 1$ and $0 < \zeta \leq 1$. In this work, the regularization functional in Equation 41 was preferred over Tikhonov regularization due to its ability to capture sharp discontinuities in inverse problems settings. The interested reader is referred to [29] and explore other regularization methodologies that can be applied for inverse problems.

3.1.2 Helmholtz filter

The aim of this section is to present the Helmholtz filter as an alternative to regularization methods, e.g. Equation 41. Regularization functionals are often (if not always) explicitly incorporated into the objective function to solve ill-posed inverse problems. The purpose of regularization functionals is to penalize the objective function and enhance the smoothness of the control field. Thus, bounding the objective function and preventing undesired data overfitting.

Instead of just applying a regularization functional, e.g. Equation 41, to solve the inverse problem in Equation 39, a Helmholtz PDE filter is also employed to filter the optimal control computed using the CEM formulation. Helmholtz PDE filters have been successfully used in topology optimization to avoid numerical artifacts and enhance the design's smoothness and boundary description [30]. The filtered control is computed by applying a convolution operator to the optimal control. However, instead of explicitly defining the convolution integral, the filtered control can be defined implicitly as the solution of the following Helmholtz PDE

$$-r^2\nabla^2\tilde{\mathbf{z}} + \tilde{\mathbf{z}} = \mathbf{z}, \tag{42}$$

with Neumann boundary conditions

$$\frac{\partial \tilde{\mathbf{z}}}{\partial \mathbf{n}} = 0. \tag{43}$$

The parameter r denotes a given filter length scale and $\tilde{\mathbf{z}}$ is the filtered control, which was set to $\frac{1}{3\times 10^3}$ for this study. As $\mathbf{x}/r \to \infty$, the filtering effect on the optimal control is reduced. Thus, larger values of r minimize the filtering effect on the optimal control.

In this work, the discretized Helmholtz PDE is solved after optimization. Contrary, we could have eliminated the regularization functional from Equation 39 and solved the discretized Helmholtz PDE every time a new set of trial controls was computed by the optimization algorithm. However, this would have increased the computational demands of the problem. Furthermore, preliminary results performed as part of this study did not justify this approach. The quality of the solution obtained by applying the filter during optimization did not improve greatly compared to the quality obtained by applying the filter after optimization.

3.2 Optimality conditions

The Lagrangian functional $\mathcal{L}\colon \mathbb{U} \times \hat{\mathbb{Z}} \times \mathbb{Y} \to \mathbb{R}$ for the inverse problem defined in Equation 39 is given by

$$\mathcal{L}(\mathbf{u}, \mathbf{z}, \mathbf{v}) = \frac{\beta}{2} \| \langle \mathbf{u}, \mathbf{A}(\mathbf{z})\mathbf{u} \rangle_{\mathbb{H}} - \langle \hat{\mathbf{u}}, \mathbf{A}(\mathbf{z})\hat{\mathbf{u}} \rangle_{\mathbb{H}} \|^2_{\mathbb{H}} + \mathsf{R}(\mathbf{z}) + \langle \mathbf{v}, \mathbf{A}(\mathbf{z})\mathbf{u} - \mathbf{f} \rangle_{\mathbb{Y}^*, \mathbb{Y}}. \tag{44}$$

The first order necessary optimality conditions for Equation 44 are given by

$$\mathcal{L}_{\mathbf{u}}(\mathbf{u}, \mathbf{z}, \mathbf{v}) = \alpha\beta \mathbf{A}(\mathbf{z})\mathbf{u} + \mathbf{A}(\mathbf{z})\mathbf{v} = 0 \tag{45}$$

$$\mathcal{L}_{\mathbf{z}}(\mathbf{u}, \mathbf{z}, \mathbf{v}) = \alpha\beta[(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u})\mathbf{u} + (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\hat{\mathbf{u}})\hat{\mathbf{u}}] + \mathsf{R}_{\mathbf{z}}(\mathbf{z}) + (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u})\mathbf{v} = 0, \tag{46}$$

where α is given by Equation 13. The first order derivative operator $R_{\mathbf{z}}(\mathbf{z})\colon \hat{\mathbb{Z}} \to \mathbb{Z}$ is given by

$$R_{\mathbf{z}}(\mathbf{z}) = \frac{\zeta}{2}(\theta - 1)(\langle \nabla \mathbf{z}, \nabla \mathbf{z} \rangle_{\mathbb{H}} + \nu^2)^{\theta - 1} \mathbf{B}\, \mathbf{z}, \tag{47}$$

where

$$\mathbf{B} = \int_{\Omega} \nabla\psi \nabla\psi \, d\Omega. \tag{48}$$

If $\mathbf{A}(\mathbf{z})$ is a non-singular, self-adjoint linear operator, the Lagrange multipliers for an inverse problem in heat transfer are given by Equation 14. Substituting Equation 14 into Equation 46 yields the following gradient operator

$$\nabla \mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z}) = -\alpha\beta[(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u})\mathbf{u} + (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\hat{\mathbf{u}})\hat{\mathbf{u}}] + \mathsf{R}_{\mathbf{z}}(\mathbf{z}). \tag{49}$$

Notice, as previously demonstrated in Section 2, that the computation of Equation 49 does not require the solution of the adjoint system of equations.

If second order optimization algorithms are available, Newton's method can be applied to the first order necessary optimality conditions. The following derivative operators are then required to compute the application of the trial step to the analytical Hessian operator

$$\mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}), \mathbf{z})\delta\mathbf{z} = (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z})\mathbf{u} \tag{50}$$

$$\mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{u} = \mathbf{A}(\mathbf{z})\delta\mathbf{u} \tag{51}$$

$$\mathsf{g}_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\delta\mathbf{z} = (\mathbf{A}_{\mathbf{z}}(\mathbf{z})^*\mathbf{v})\mathbf{u} \tag{52}$$

$$\mathsf{g}_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\delta\mathbf{u} = \mathbf{A}(\mathbf{z})^*\mathbf{v} \tag{53}$$

$$\mathsf{g}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\delta\mathbf{u} = \mathbf{0} \tag{54}$$

$$\mathsf{g}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\delta\mathbf{z} = (\mathbf{A}_{\mathbf{z}}(\mathbf{z})^*\delta\mathbf{z})\mathbf{v} \tag{55}$$

$$\mathsf{g}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\delta\mathbf{u} = \mathbf{0} \tag{56}$$

$$\mathsf{g}_{\mathbf{zu}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\delta\mathbf{u} = (\mathbf{A}_{\mathbf{z}}(\mathbf{z})^*\mathbf{v})\delta\mathbf{u} \tag{57}$$

$$\mathsf{J}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{u} = 2\beta[\alpha\mathbf{A}(\mathbf{z})\delta\mathbf{u} + 2\langle\mathbf{u},\mathbf{A}(\mathbf{z})\delta\mathbf{u}\rangle_{\mathbb{H}}\mathbf{A}(\mathbf{z})\mathbf{u}] \tag{58}$$

$$\mathsf{J}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = 2\beta[\alpha(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z})\mathbf{u} + \gamma\mathbf{A}(\mathbf{z})\mathbf{u}] \tag{59}$$

$$\mathsf{J}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = \gamma\beta[(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u})\mathbf{u} - (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\hat{\mathbf{u}})\hat{\mathbf{u}}] + \mathsf{R}_{\mathbf{zz}}(\mathbf{z})\delta\mathbf{z} \tag{60}$$

$$\mathsf{J}_{\mathbf{zu}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{u} = 2\beta[\alpha(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u})\delta\mathbf{u} + ((\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u})\mathbf{u} - (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\hat{\mathbf{u}})\hat{\mathbf{u}})\langle\mathbf{u},\mathbf{A}(\mathbf{z})\delta\mathbf{u}\rangle_{\mathbb{H}}], \tag{61}$$

where γ is given by Equation 28 and the Lagrange multipliers $\mathbf{v}$ are given by Equation 14. Finally, the second order derivative operator $\mathsf{R}_{\mathbf{zz}}(\mathbf{z})$ is given by

$$\mathsf{R}_{\mathbf{zz}}(\mathbf{z}) = \frac{\zeta}{2}((\theta-1)(\langle\nabla\mathbf{z},\nabla\mathbf{z}\rangle_{\mathbb{H}} + \nu^2)^{\theta-2})\langle\nabla\mathbf{z},\nabla\mathbf{z}\rangle_{\mathbb{H}}\mathbf{B} + (\langle\nabla\mathbf{z},\nabla\mathbf{z}\rangle_{\mathbb{H}} + \nu^2)^{\theta-1}\mathbf{B}. \tag{62}$$

Substituting Equations 50-61 into Equation 24 yields the application of the trial step to the analytical Hessian, as defined in Equation 33. However, if the Gauss-Newton Hessian is applied, the application of the trial step to the Gauss-Newton Hessian operator is then given by

$$\nabla^2\mathsf{J}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = \gamma\beta[(\mathbf{A}_{\mathbf{z}}(\mathbf{z})\mathbf{u})\mathbf{u} - (\mathbf{A}_{\mathbf{z}}(\mathbf{z})\hat{\mathbf{u}})\hat{\mathbf{u}}] + \mathsf{R}_{\mathbf{zz}}(\mathbf{z})\delta\mathbf{z}, \tag{63}$$

where $\mathsf{R}_{\mathbf{zz}}(\mathbf{z})$ is given by Equation 62.

Finally, the discretized Helmholtz equation is given by

$$r\mathbf{B}\hat{\mathbf{z}} + \mathbf{M}\hat{\mathbf{z}} = \mathbf{z}, \tag{64}$$

where

$$\mathbf{M} = \int_{\Omega} \psi\psi \, d\Omega. \tag{65}$$

and $\mathbf{B}$ is defined by Equation 48.

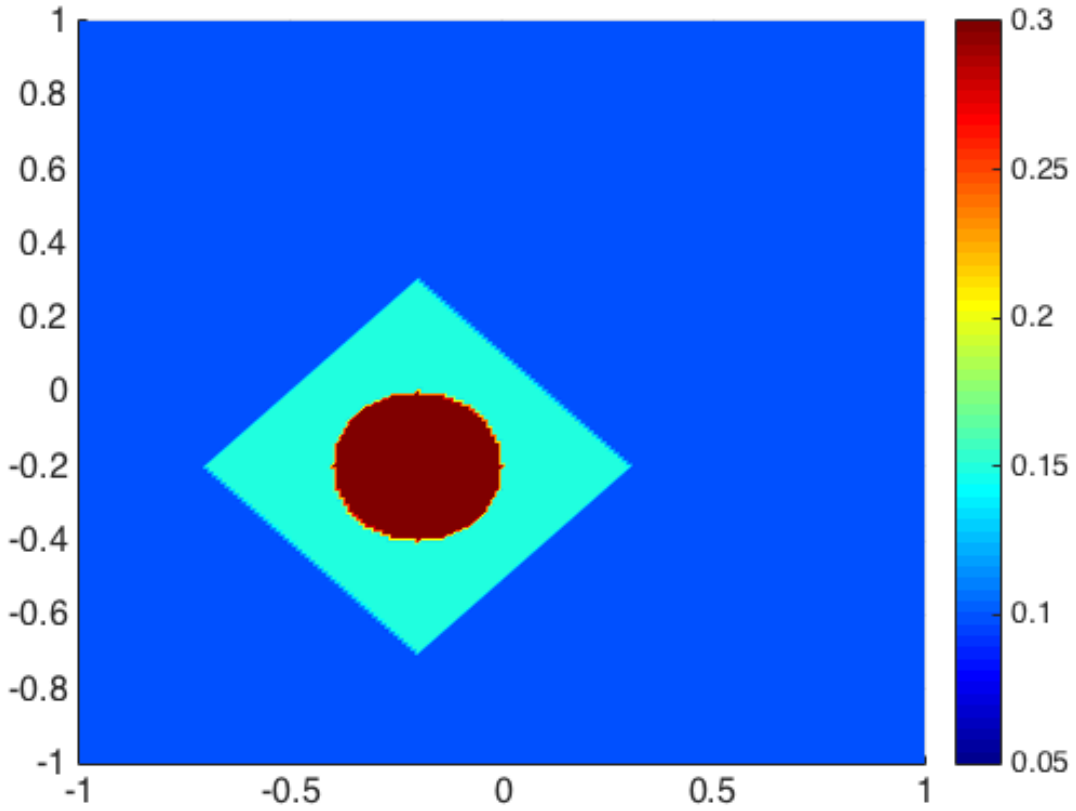


Figure 1: Target thermal conductivity field.

3.3 Results: full-field temperature field

The compliance error minimization and the data misfit formulations were applied to the inverse problem defined in Equation 39. The amplitude for the heat source was set to 1×10^2 and the angular frequency was set to 4π and 16π for the numerical studies performed with the data misfit and compliance error functionals, respectively. The target thermal conductivity field is shown in Figure 1. The thermal conductivity coefficients lower and upper bounds were respectively set to 0.01 and 1.0 during optimization. The regularization parameter ζ was set to 1.0 for all the numerical studies performed herein. To quantify the computational efficacy of the compliance error functional, the corresponding speedups are computed, where

$$S = \frac{CPU_{DMF}}{CPU_{CEM}}. \tag{66}$$

In Equation 66, CPU denotes central processing unit time, DMF denotes data misfit functional and CEM denotes compliance error minimization.

3.3.1 First order formulation

The Perry-Shanno nonlinear conjugate gradient was used to solve the inverse problem in heat transfer [31, 32]. The optimization algorithm stopped when $\mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z})_k$, $\|\nabla_{\mathbf{z}}\mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z})_k\|$ or $\|\mathbf{s}_k\|$ was below a predefined tolerance of 1×10^{-4}. A backtracking line search with cubic

Noise	DMF	CEM	S
0%	132.7	1.08	122.87
1%	132.25	1.15	119.35
3%	103.18	1.07	96.43
5%	63.2	1.19	53.11

Table 1: CPU time (seconds) and corresponding speedups obtained using the Perry-Shanno nonlinear conjugate gradient algorithm.

step interpolation was applied to enhance the global convergence capability of the nonlinear conjugate gradient algorithm. The line search contraction parameter was set to 0.5 and the step lower bound was set to 1×10^{-5}. Furthermore, the maximum number of line search iterations was set to 5.

The regularization parameters used for the numerical studies based on the data misfit functional were $(\zeta, \nu) = (5 \times 10^{-4}, 1 \times 10^{-4})$ for $\theta = \{0\%, 1\%\}$ and $(\zeta, \nu) = (5 \times 10^{-3}, 1 \times 10^{-4})$ for $\theta = \{3\%, 5\%\}$. Contrary, the regularization parameters used for all the numerical studies based on the CEM formulation were $(\zeta, \nu) = (1 \times 10^{-8}, 1 \times 10^{-8})$.

Table 1 shows the central processing unit times obtained for the numerical studies that are based on first order derivative information. Regardless of the noise level, the compliance error formulation produced noticeable speedups over the data misfit formulation. Clearly, the CEM formulation outperforms the data misfit formulation. Figure 2 shows that the CEM strategy required less than 10 optimization iterations to converged to an optimal and feasible solution in all the numerical studies. Contrary, the data misfit formulation strategy needed over 1000 iterations to meet the convergence criteria. Figure 2 also shows the objective function values produced by Perry-Shanno nonlinear conjugate gradient algorithm. The reader can clearly appreciate that the compliance error functional produced faster convergence rates than the data misfit functional, regardless of the level of corruption in the data.

Figure 3 shows the optimal thermal conductivity field computed using the data misfit functional and the Perry-Shanno nonlinear conjugate gradient algorithm. The thermal conductivity field was accurately approximated for all noise levels. However, the solution obtained with the data misfit functional was sensitive to higher noise levels. Figure 4 shows the optimal thermal conductivity field computed using the compliance error functional and the Perry-Shanno nonlinear conjugate gradient algorithm. The compliance error minimization formulation yielded accurate thermal conductivity fields regardless of the noise levels. However, the CEM strategy produced non-optimal thermal conductivity fields were the Dirichlet and Neumann boundary conditions were applied. More research is needed to understand why these inaccuracies are obtained around areas were Dirichlet or Neumann boundary conditions are applied.

3.3.2 Second order formulation

A dogleg trust region inexact Newton algorithm [33] was used to solve the inverse problem defined in Equation 39 when second order derivative information was available. The optimization algorithm once more stopped when one of the following stopping criterion was satisfied: $\mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z})_k < 1 \times 10^{-4}$, $\|\nabla_{\mathbf{z}} \mathsf{J}(\mathbf{u}(\mathbf{z}), \mathbf{z})_k\| < 1 \times 10^{-4}$ or $\|\mathbf{s}_k\| < 1 \times 10^{-4}$. The trust region contraction and expansion parameters were respectively set to 0.5 and 2. The maximum number of trust region sub-problem iterations was set to 5 and the minimum ratio between the actual

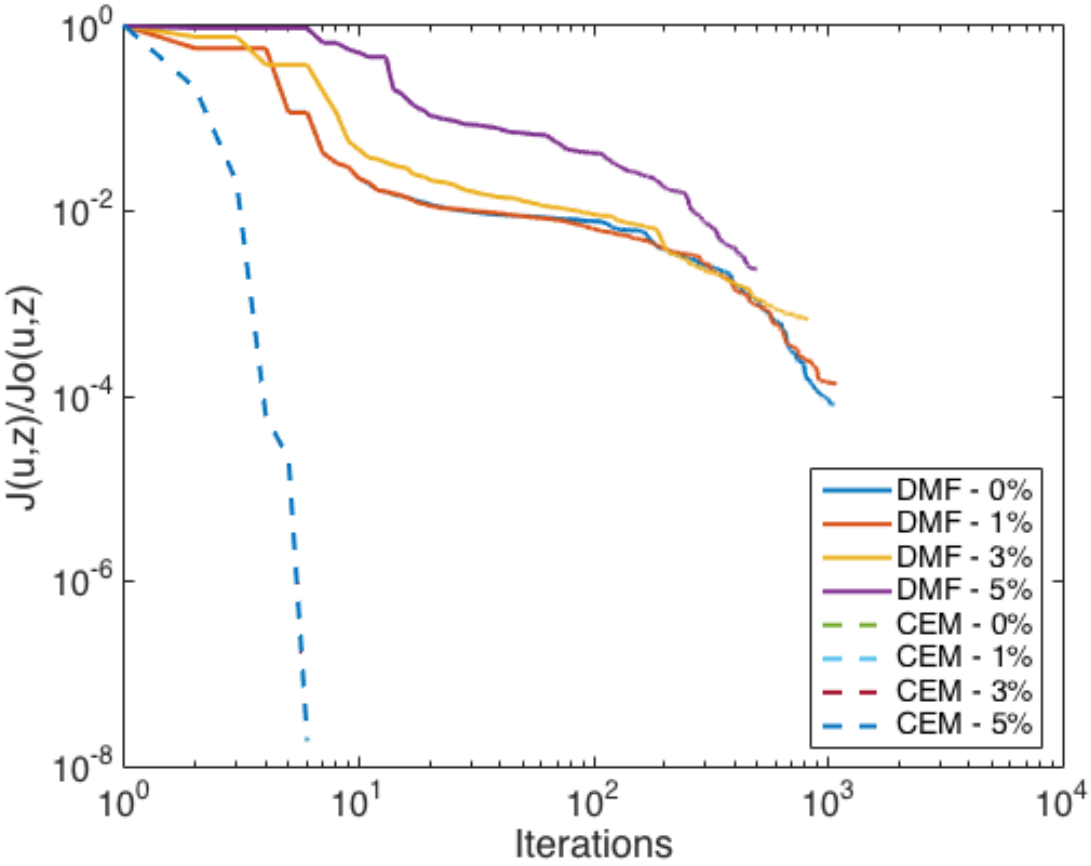
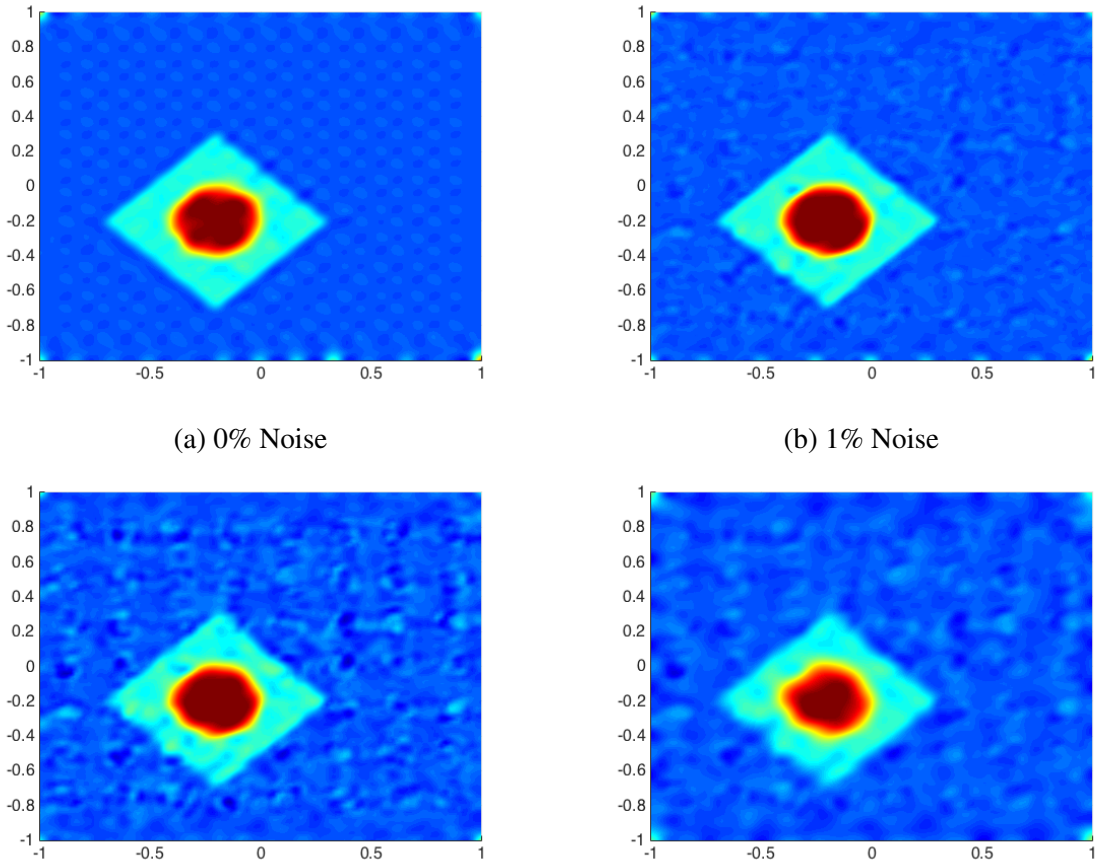


Figure 2: Objective function values computed by the Perry-Shanno nonlinear conjugate gradient algorithm.



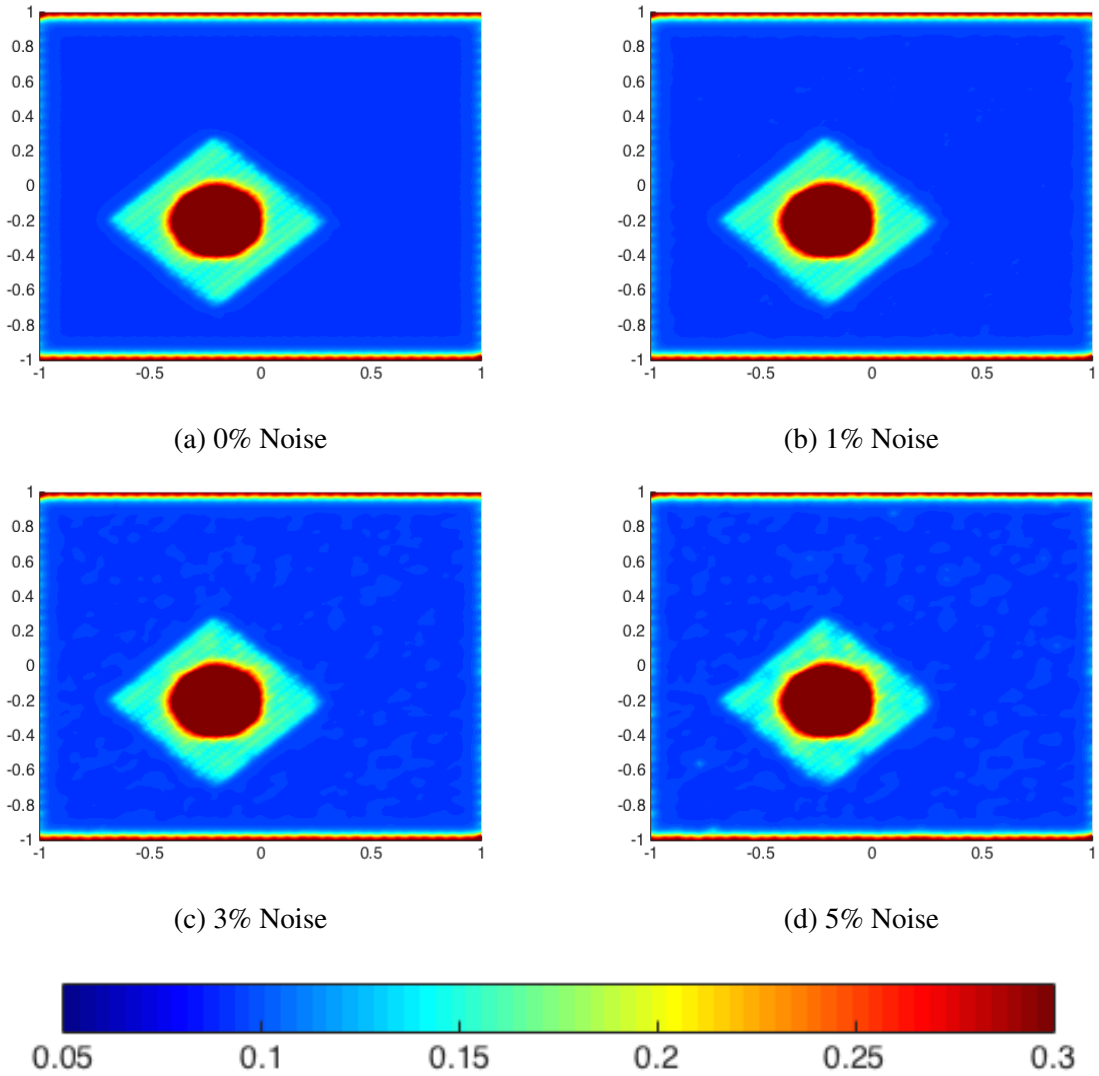
(a) 0% Noise

(b) 1% Noise

(c) 3% Noise

(d) 5% Noise

Figure 3: Optimal thermal conductivity field computed using the first order data misfit formulation strategy and the Perry-Shanno nonlinear conjugate gradient algorithm.



(a) 0% Noise

(b) 1% Noise

(c) 3% Noise

(d) 5% Noise

Figure 4: Optimal thermal conductivity field computed using the first order compliance error minimization formulation and the Perry-Shanno nonlinear conjugate gradient algorithm.

Noise	DMF	CEM-AH	CEM-GN	S-AH	S-GN
0%	123.64	8.18	5.37	15.11	23.02
1%	192.42	7.98	4.61	24.11	41.74
3%	211.29	8.06	6.07	26.21	34.81
5%	91.76	8.08	5.06	11.36	18.13

Table 2: CPU time (seconds) and corresponding speedups obtained with the dogleg trust region inexact Newton algorithm. Here, AH and GN respectively denote analytical and Gauss-Newton Hessians.

and predicted reduction was set to 0.2.

The regularization parameters used for the numerical studies based on the data misfit functional were $(\zeta, \nu) = (1\times10^{-3}, 1\times10^{-3})$ for $\theta = \{0\%, 1\%, 3\%\}$ and $(\zeta, \nu) = (5\times10^{-3}, 1\times10^{-3})$ for $\theta = 5\%$. Once more, the regularization parameters used for all the numerical studies based on the compliance error functional were $(\zeta, \nu) = (1 \times 10^{-8}, 1 \times 10^{-8})$.

Table 2 shows the CPU times gathered for the numerical studies done with the dogleg trust region inexact Newton algorithm. The results on Table 2 once more show that the compliance error minimization formulation produced significant speedups over the data misfit formulation. However, the CPU times obtained with the CEM formulation and the second order optimization algorithm were higher than those obtained with the first order optimization algorithm. Therefore, the second order derivative information did not enable additional speedups. However, these results are specific to the parameter estimation problem in heat transfer and more studies are necessary to further understand the possible benefits of the second order compliance error formulation. Lets recall that an effective preconditioning strategy could be applied to improve the performance of the Newton algorithm and reduce the number of Newton iterations. However, this was outside the scope of this study.

Figure 5 displays the objective function values computed using the dogleg trust region inexact Newton algorithm. Once more, the compliance error minimization formulations required less than 10 optimization iterations to converged to an optimal solution. Contrary, the data misfit formulation required over 50 optimization iterations (in some cases over 100 iterations) to meet one of the convergence criterion. However, why the CPU times produced by the second order optimization algorithm using the data misfit functional are greater than those produced by the first order optimization algorithm? These results can be counterintuitive to the reader since the numerical studies based on the first order data misfit formulation needed more iterations to converge to an optimal solution. The reader should recall that second order optimization algorithms require at least 4 model evaluations (2 for the gradient and 2 for the Hessian calculation) per optimization iteration. Furthermore, every Newton iteration requires 2 model evaluations to compute trial descent direction. Hence, regardless of the fact that the trust region Newton algorithm needed less optimization iterations to converge, the number of model evaluations will always dominate CPU time. This further motivates future research to enable an effective preconditioning strategy for the compliance error minimization formulation.

The same argument can be applied to the second order CEM formulation based on the full Hessian since at least 2 model evaluations (1 for the gradient and 1 for the Hessian calculation) are performed per iteration to compute a new control field. Hence, the higher CPU times seen on Table 2. Next, lets consider the CEM formulation based on the Gauss-Newton Hessian. This formulation only needs one model evaluation per optimization iteration. Thus, why the CPU

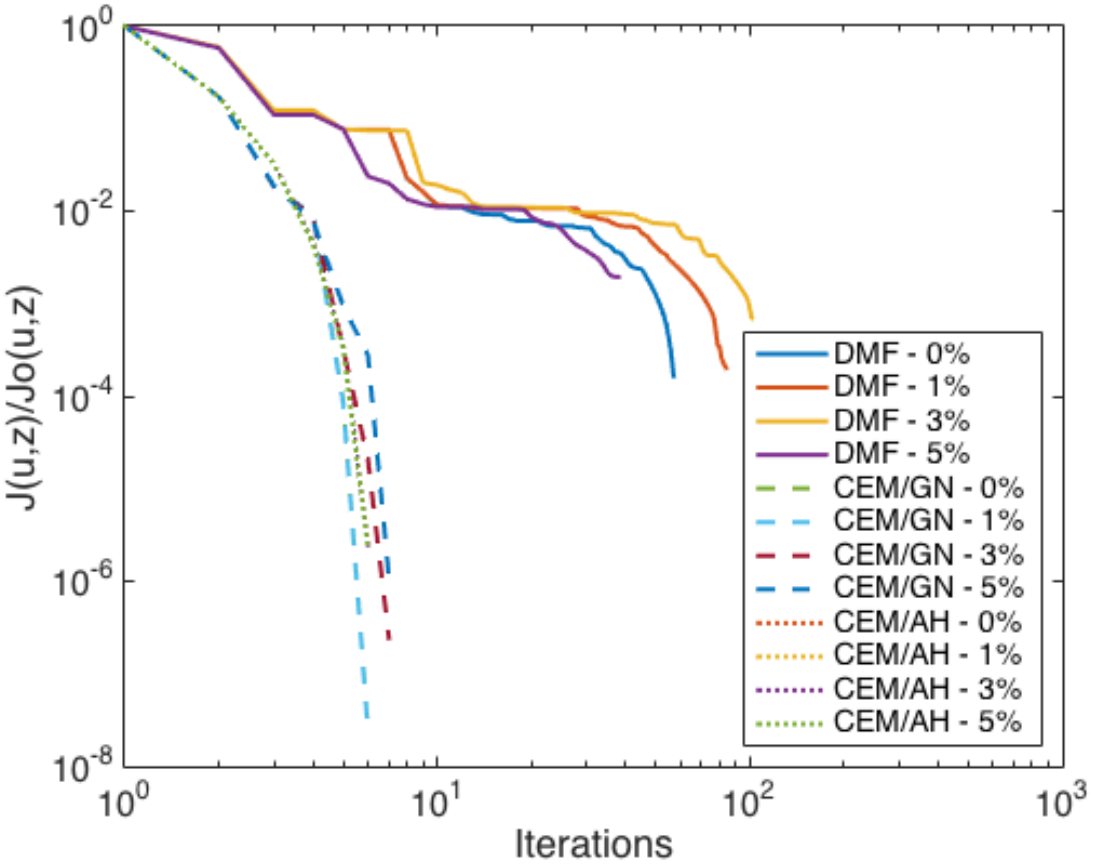
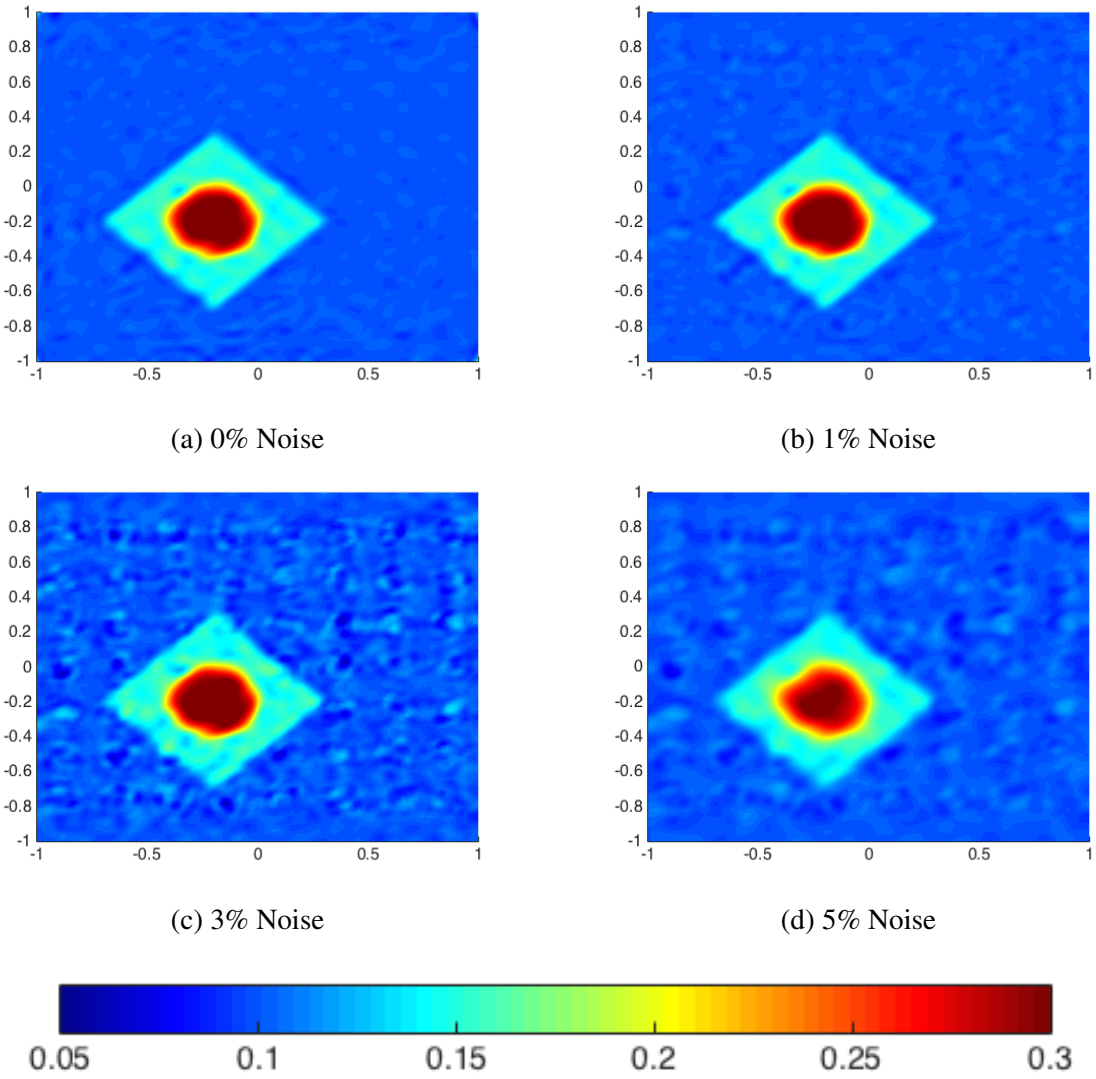


Figure 5: Objective function values computed using the dogleg trust region inexact Newton algorithm.

times are not closer/similar to the CPU times observed for the first order CEM formulation? In all our numerical studies, the optimization algorithm required several trust region sub-problem iterations to compute a trial control that met the predefined ratio between the actual and the predicted reduction. Contrary, the first order CEM formulation did not need additional line search iterations to compute a descent direction that produced an optimal/feasible trial control. Therefore, less model evaluations were performed during optimization; thus, reducing computational demands.

Figure 5 also shows the convergence rates for the second order formulations. Notice that the CEM formulations displayed faster convergence rates than the second order data misfit formulation, similar to the results shown in Section 3.3.1. However, the convergence rates for the second order data misfit formulation displayed faster convergence rates near the optimal/feasible point. This was expected since Newton algorithms are expected to converge quadratically near the optimal/feasible point.

Finally, Figure 6 shows the optimal thermal conductivity field computed using the second order data misfit formulation for all noise levels. Results demonstrate that the second order data misfit formulation produced optimal thermal conductivity fields. However, once more the results show that the optimal solutions were sensitive to corruption in the data. Figures 7 and 8 show the optimal thermal conductivity field computed using the second order CEM formulations based on the analytical and Gauss-Newton Hessians, respectively. Both approaches



(a) 0% Noise

(b) 1% Noise

(c) 3% Noise

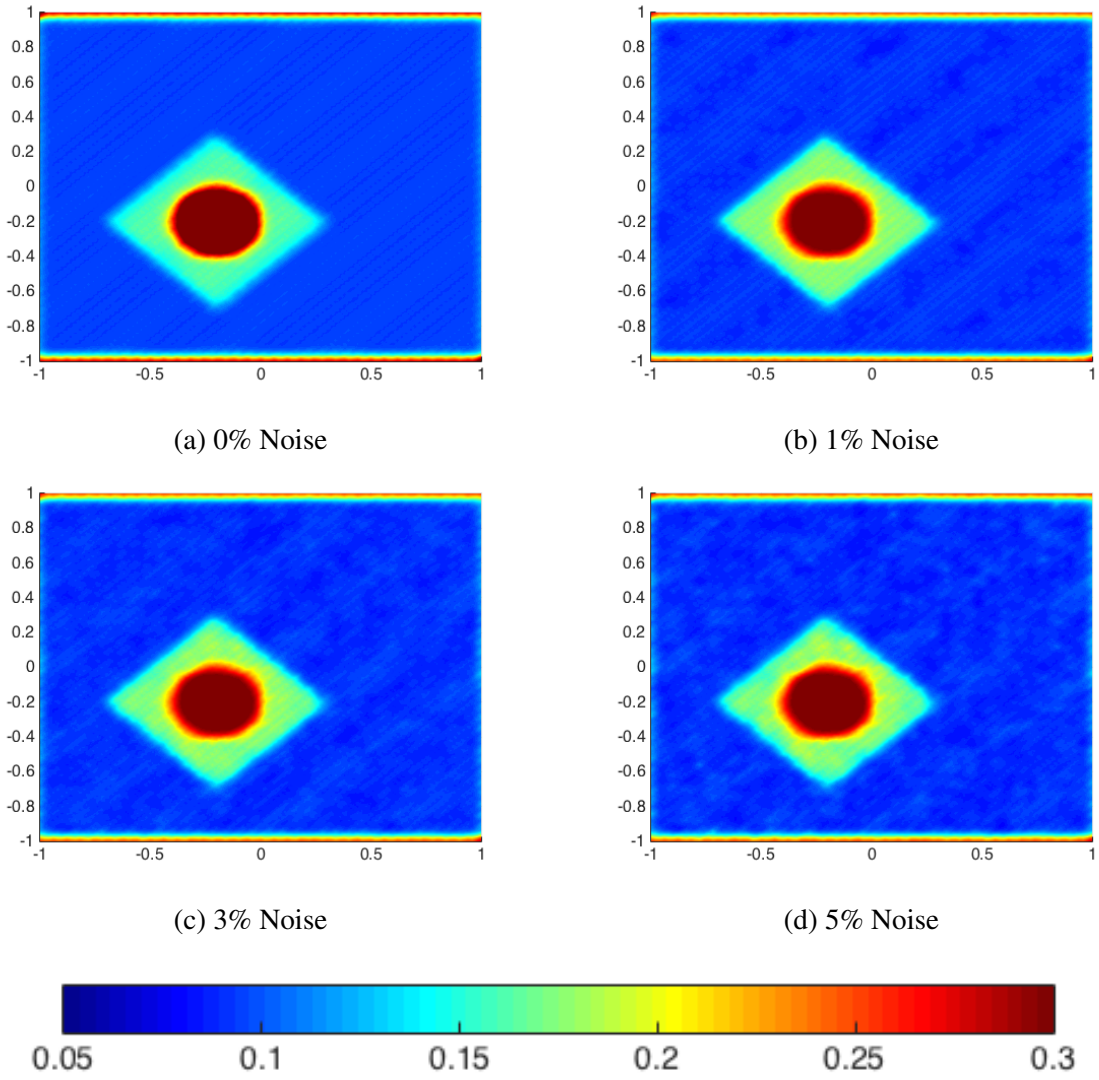
(d) 5% Noise

Figure 6: Optimal thermal conductivity field computed using the second order data misfit formulation and the inexact dogleg trust region Newton algorithm.

accurately characterized the location and the shape of the thermal conductivity field of interest. However, the magnitude of the thermal conductivity field inside the heterogeneous conductivity field was overestimated. Furthermore, the compliance error minimization formulation once more produced non-optimal thermal conductivity values around the regions were Dirichlet and Neumann boundary conditions were applied. These results further highlight the need for more research to eliminate the numerical artifacts computed around these regions. However, the significant speedups observed with the compliance error minimization formulations motivate future research to continue improving the proposed inverse problem formulation.

4 CONCLUSIONS

This paper presented a new formulation for inverse problems based on the compliance error functional. The compliance error functional enabled the computation of the Lagrange multipliers at the expense of just one model evaluation. Thus, the calculation of the Lagrange multipliers does not require the solution of the computationally intensive adjoint problem. This, leads to



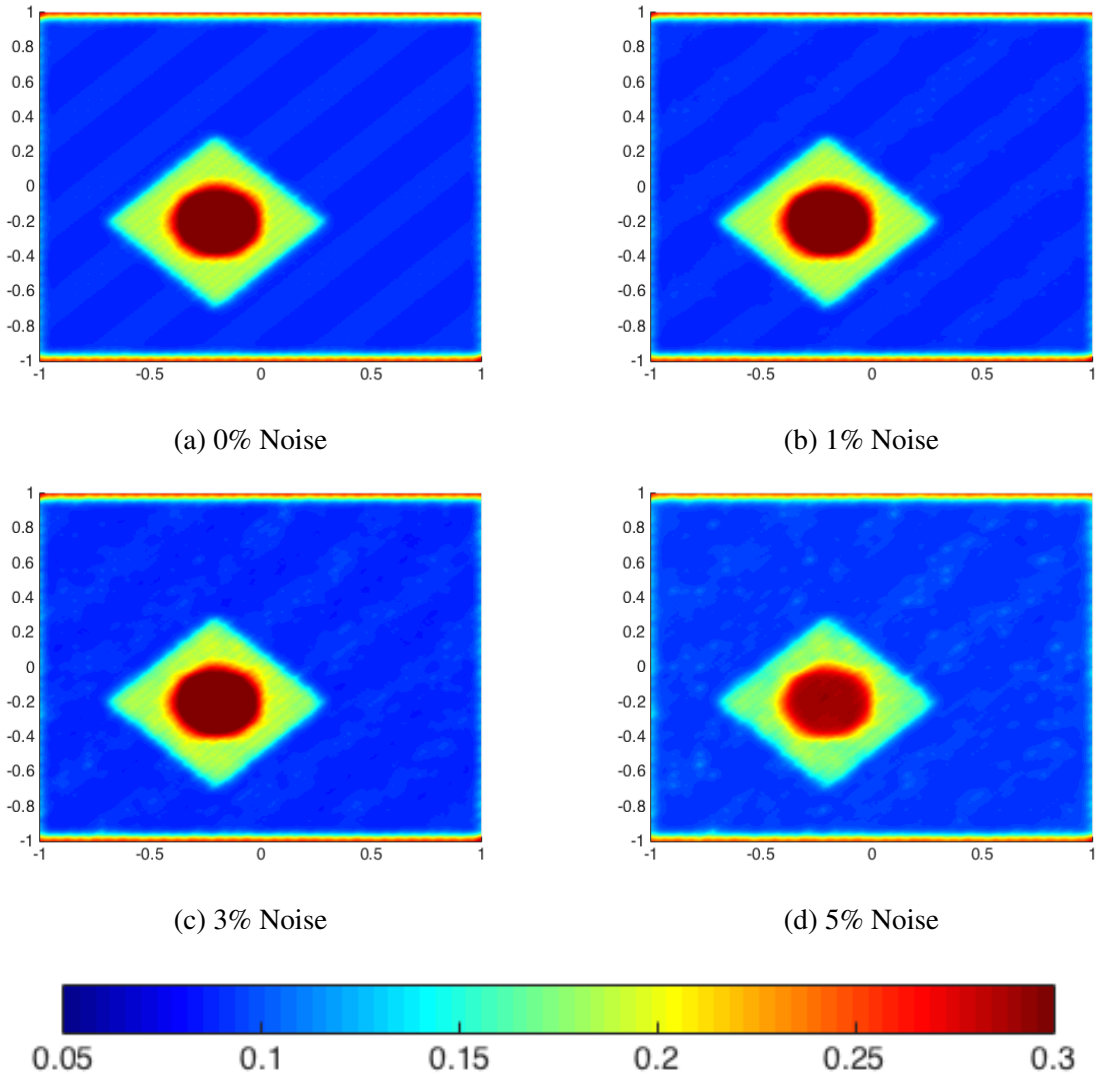
(a) 0% Noise

(b) 1% Noise

(c) 3% Noise

(d) 5% Noise

Figure 7: Optimal thermal conductivity field computed using the second order compliance error minimization formulation based on the analytical Hessian and the inexact dogleg trust region Newton algorithm.



(a) 0% Noise

(b) 1% Noise

(c) 3% Noise

(d) 5% Noise

Figure 8: Optimal thermal conductivity field computed using the second order compliance error minimization formulation based on the Gauss-Newton Hessian and the inexact dogleg trust region Newton algorithm.

significant speedups since the gradient operator can be computed at the expense of one model evaluation per optimization iteration. Furthermore, the implementation of the CEM formulation in production finite element software is greatly simplified since the propose formulation only requires forward model evaluations to compute the gradient. Therefore, the implementation of the adjoint system of equations is not necessary to solve the inverse problem. Likewise, computing the application of the trial step to the analytical Hessian operator is greatly simplified since only one additional model evaluation is requires. This also leads to significant speedups when second order optimization algorithms and the CEM formulation are applied to solve the inverse problem.

This paper also presented a Gauss-Newton Hessian formulation based on the compliance error functional that effectively approximates the analytical Hessian during optimization. By applying the Gauss-Newton Hessian, optimal results were recovered, while reducing the computational cost associated with evaluating the analytical Hessian. Furthermore, the Gauss-Newton Hessian approximation further reduced the computational demands of the inverse problem. However, results suggested that the first order CEM formulation was computationally more effective than the second order CEM formulation. Therefore, the second order CEM formulations did not provide additional computational benefits over the first order CEM formulation. However, more research is needed since the second order CEM formulations can be combined with an effective preconditioning strategy to improve accuracy and computational efficacy.

Results also showed that the data misfit formulation produced better results around the regions were Dirichlet and Neumann boundary conditions were applied. The compliance error functional produce perceivable numerical artifacts near these regions. However, the interior thermal conductivity fields computed with the first and second order CEM formulations were accurate. It was also observed that the second order CEM formulations were inclined to overestimate the thermal conductivity field around the heterogeneous thermal conductivity region. Future research will focus on exploring alternate inverse problem formulations to improve the CEM strategy, without losing its computational advantages. Finally, it was observed in this study that compliance error minimization formulation was less sensitive to corruption in the experimental data.

Overall, the results obtained with the compliance error minimization formulation are encouraging due to the significant speedups produced during optimization. Furthermore, the first and second order CEM formulation yield accurate thermal conductivity fields. Although their is room for improvements, the gains in speed are substantial and thus motivate future research to continue improving the efficacy of the proposed inverse problem formulation.

REFERENCES

[1] M. Aguiló, W. Aquino, J. Brigham, and M. Fatemi, "An inverse problem approach for elasticity imaging through vibroacoustics," *Medical Imaging, IEEE Transactions on*, vol. 29, no. 4, pp. 1012–1021, 2010.

[2] M. Gockenbach, B. Jadamba, and A. Khan, "Numerical estimation of discontinuous coefficients by the method of equation error," *Mathematics and Mechanics of Solids*, vol. 1, pp. 343–359, 2006.

[3] H. L. Liew and P. M. Pinsky, "Recovery of shear modulus in elastography using an adjoint method with b-spline representation," *Finite Elements in Analysis and Design*, vol. 41, no. 7-8, pp. 778–799, 2005.

[4] R. Sampath and N. Zabaras, "An object-oriented framework for the implementation of adjoint techniques in the design and control of complex continuum systems," *International Journal for Numerical Methods in Engineering*, vol. 48, no. 2, pp. 239–266, 2000.

[5] A. Deraemaeker, P. Ladevèze, and P. Leconte, "Reduced bases for model updating in structural dynamics based on the constitutive relation error," *Computer Method in Applied Mechanics and Engineering*, vol. 191, no. 21-22, pp. 2427–2444, 2002.

[6] P. Feissel and O. Allix, "Modified constitutive relation error identification strategy for transient dynamics with corrupted data: The elastic case," *Computer Method in Applied Mechanics and Engineering*, vol. 196, no. 13-16, pp. 1968–1983, 2007.

[7] R. V. Kohn and B. D. Lowe, "A variational method for parameter identification," *Mathematical Modeling and Numerical Analysis*, vol. 22, no. 1, pp. 119–158, 1998.

[8] R. Kohn and A. McKenney, "Numerical implementation of a variational method for electrical impedance tomography," *Inverse Problems*, vol. 6, no. 3, pp. 389–414, 1990.

[9] R. Kohn and M. Vogelius, "Determining conductivity by boundary measurements," *Communications on Pure and Applied Mathematics*, vol. 37, no. 3, pp. 289–298, 1984.

[10] R. Kohn and M. Vogelius, "Determining conductivity by boundary measurements II. interior results," *Communications on Pure and Applied Mathematics*, vol. 38, no. 5, pp. 643–667, 1985.

[11] P. Ladevèze and C. A., "Updating structural dynamic models with emphasis on the damping properties," *American Institute of Aeronautics and Astronautics Journal*, vol. 36, no. 6, pp. 1094–1099, 1998.

[12] P. Ladevèze and D. Leguillon, "Error estimates procedures in the finite element method and applications," *SIAM Journal on Numerical Analysis*, vol. 20, no. 3, pp. 485–509, 1983.

[13] O. Allix, P. Feissel, and H. Nguyen, "Identification strategy in the presence of corrupted measurements," *Engineering computations*, vol. 22, no. 5-6, pp. 487–504, 2005.

[14] M. S. Gockenbach and A. A. Khan, *Mathematical Models and Methods for Real World Systems*, ch. A convex objective functional for elliptic inverse problems. Chapman & Hall CRC Taylor & Francis Group, 2006.

[15] M. S. Gockenbach and A. A. Khan, "An abstract framework for elliptic inverse problems: Part 1. An output least-squares approach," *Mathematics and Mechanics of Solids*, vol. 12, no. 3, pp. 259–276, 2007.

[16] M. S. Gockenbach and A. A. Khan, "An abstract framework for elliptic inverse problems: Part 2. An augmented Lagrangian approach," *Mathematics and Mechanics of Solids*, vol. 14, no. 6, pp. 517–539, 2009.

[17] M. Gockenbach, "Numerical analysis of elliptic inverse problems with interior data," *Journal of Physics: Conference Series*, vol. 124, no. 1, pp. 1–12, 2008.

[18] F. Pierron and M. Grédiac, *The Virtual Fields Method: Extracting Constitutive Mechanical Parameters from Full-field Deformation Measurements*. Springer New York, 2012.

[19] M. Grdiac, F. Pierron, S. Avril, and E. Toussaint, "The virtual fields method for extracting constitutive parameters from full-field measurements: a review," *Strain*, vol. 42, no. 4, pp. 233–253, 2006.

[20] M. Aguiló, L. Swiler, and A. Urbina, "First-order formulations for large-scale stochastic parameter estimation within the frameworks of steady state dynamics: the elastic and viscoelastic case," *Inverse Problems*, vol. 28, no. 7, p. 075003, 2012.

[21] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, and A. Williams, "An Overview of Trilinos," SAND2003-2927, Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185-1110, 2003.

[22] "MATLAB 8.0 and Statistics Toolbox 8.1, The MathWorks, Inc., Natick, Massachusetts, United States."

[23] M. A. Aguiló, "Design Optimization Toolkit (DOTk): Users' Manual," SAND2014-16272, Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185-0845, 2014.

[24] B. Adams, W. Bohnhoff, K. Dalbey, J. Eddy, M. Eldred, D. Gay, K. Haskell, P. Hough, and L. Swiler, "Dakota, a multilevel parallel object-oriented framework for design optimization, parameter estimation, uncertainty quantification, and sensitivity analysis: version 5.0 users manual," *Sandia National Laboratories, Tech. Rep. SAND2010-2183*, 2009.

[25] D. Kouri, D. Ridzal, B. van Bloemen Waanders, and G. van Winckel, "Rapid optimization library (ROL)." `https://trilinos.org/packages/rol/`, 2014.

[26] P. Gill, W. Murray, and M. Saunders, "Snopt: An sqp algorithm for large-scale constrained optimization," *SIAM journal on optimization*, vol. 12, no. 4, pp. 979–1006, 2002.

[27] A. Wächter and L. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, no. 1, pp. 25–57, 2006.

[28] H. Banks, M. Joyner, B. Wincheski, and W. Winfree, "Nondestructive evaluation using reduced-order computational methodology," *Inverse Problems*, vol. 16, no. 4, pp. 929–945, 2000.

[29] C. Vogel, *Computational methods for inverse problems*. Society for Industrial and Applied Mathematics, 2002.

[30] B. Lazarov and O. Sigmund, "Filters in topology optimization based on helmholtz-type differential equations," *International Journal for Numerical Methods in Engineering*, vol. 86, no. 6, pp. 765–781, 2011.

[31] J. Perry, "A class of conjugate gradient algorithms with a two-step variable-metric memory," *Discussion Papers*, vol. 269, 1977.

[32] D. Shanno, "On the convergence of a new conjugate gradient algorithm," *SIAM Journal on Numerical Analysis*, vol. 15, no. 6, pp. 1247–1257, 1978.

[33] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Trust region methods*, vol. 1. SIAM, 2000.