SAND2015-9581J

A trust region algorithm for PDE-constrained optimization with bound constraints using reduced order modeling

Miguel A. Aguiló[a,*]

[a] *Computational Solid Mechanics and Structural Dynamics Department, Sandia National Laboratories, PO Box 5800, MS 0380 Albuquerque, NM 87185-0380, USA*

Abstract

This paper presents a novel trust region algorithm that relies on proper orthogonal decomposition techniques to construct accurate reduced order models during optimization. The algorithm samples high fidelity snapshots to compute the POD functions that are used to generate reduced order models. The reduced order models are used to replace the computationally intensive high fidelity finite element evaluations during optimization. The proposed algorithm employs a trust region framework to detect loss in predictive accuracy in the reduced order model and automatically update the POD functions during optimization. The trust region framework allows the algorithm to use sound mathematical metrics to effectively improve the accuracy and robustness of the reduced order models during optimization. The algorithm also employs a projected gradient algorithm to model bound constraints and compute optimal and feasible controls.

This paper also presents an accurate Hessian formulation for topology optimization problems. The proposed trust region framework relies on a quadratic model to update the control. This quadratic model needs reliable second order derivative information to predict the behavior of the objective function within a suitable trust region. If a nonlinear Hessian formulation is used, the computational effort increases due to additional finite element evaluations. The proposed linear Hessian formulation reduces the computational effort and enables the calculation of the second order derivative information without additional finite element model evaluations. Examples in topology optimization are presented to demonstrate the applicability of the proposed algorithm and linear Hessian formulation for large-scale PDE-constrained optimization.

Keywords: trust region, Newton algorithm, reduced order model, bound constraints, projected gradient, topology optimization

*Corresponding author

Email address: maguilo@sandia.gov (Miguel A. Aguiló)

1. Introduction

High fidelity models have become an integral tool across a range of engineering applications. Furthermore, large-scale partial differential equation (PDE) constrained optimization problems are essential in many engineering design applications. However, the iterative nature and computational requirements of optimization problems governed by PDE constraints limits the amount of optimization iteration that can be preformed. Thus, the development of faster, accurate optimization algorithms for large-scale optimization is crucial to meet the pressing needs of high-consequence parts designers. This paper presents a novel trust region algorithm that relies on proper orthogonal decomposition techniques to generate accurate reduced order models during optimization. The proposed algorithmic framework enables the reduction of computational time by replacing high fidelity finite element model (FEM) evaluations with accurate, low fidelity FEM evaluations.

Proper orthogonal decomposition (POD) is one popular model reduction technique that is used to construct low fidelity models of high fidelity FEMs [2, 8, 21]. POD is based on projecting the system response onto subspaces of basis elements that contain characteristics of the expected state solution [37]. Several approaches have been proposed in the literature for surrogate-based optimization [28]. Galbally [15] employed masked projection techniques to compute approximations of nonlinear terms using a subset of interpolations terms to construct a hyper-reduced order model. The reduced model was then applied to solve a statistical inverse problems to characterize two control parameters. In this work, the reduced order model was not updated during optimization. Kahlbacher [19] applied POD surrogate models to solve an optimal control problem governed by a bilinear elliptic equation. A sequential quadratic programming (SQP) algorithm was apply to solve the optimal control problem. The resulting derivative operators were discretized by a Galerkin-POD approach and a Gauss-Newton approach was applied to approximate the second order derivative information. The inexactness induced by the reduced solution was controlled by an *a-posteriori* error estimate introduced by Tröltzsch [40]. In this work, the reduced order model was first generated using 441 snapshots computed on an equidistant grid before solving the optimization problem. The state, control, and lagrange multipliers were reduced during optimization and the total number of control variables was two. Sachs [32] used POD *a-posteriori* error estimates based on a perturbation method to control the inexactness induced by the reduced solution. A SQP algorithm was applied to solve the optimal control problem. The number of control variables was four in this work. Bound constraints were ignored in [19, 32].

Fahl [4, 14] presented a trust region POD (TRPOD) framework that enabled adaptive updates of the POD functions during optimization. Instead of using a quadratic model to quantify the ratio between the actual and predicted reduction in the objective function, Fahl used a nonlinear function for the objective function based on the reduced order model. This approach required the evaluation of the high fidelity model in every trust region subproblem iteration. Kragel [20] extended the work done in [4, 14] by applying a multi-level

strategy that relied on coarser (low fidelity) FEM evaluations during the trust region subproblem iterations. The POD approach was also applied to the coarser FEM evaluations to further expedite analysis. The TRPOD framework was only applied to small-scale unconstrained optimization. Fewer than 10 control variables were used the work by Kragel.

Model reduction techniques have not been extensively used in topology optimization settings. Jensen [18] used Padé approximations to construct reduced order models for topology optimization of dynamic problems. The proposed approach is specific to dynamic problems and thus is not applicable to general optimization problems outside the scoop of topology optimization of dynamic problems. Furthermore, no mathematical metric was used for the adaptive Padé functions updates during optimization. Yoon [43] studied the application of mode superposition, Ritz vector, and quasi-static Ritz vector methods in the context of topology optimization of dynamic problems. In [43], the orthogonal functions are updated at every optimization iteration. Thus, no adaptive updating scheme is proposed for the orthogonal functions. Additional computational savings can be achieved in [43] through the implementation of an adaptive updating scheme.

POD methods have not been explored in the context of topology optimization problems. However, these techniques have been successfully applied to solve multiple calibration and shape optimization problems in the literature [3, 44, 45]. Although, the number of control variables was small in all of these case studies, the POD strategy was successful reducing computational cost. There are recent efforts towards applying model reduction techniques for large-scale optimization settings [6]. Lieberman et al. recently combined POD techniques and a greedy procedure to construct a reduced models for both the state (pressure) and control (spatially varying conductivity) variables to solve a statistical inverse problem [22]. The reduced order model was generated by randomly sampling 1000 control fields from the Gaussian prior before solving the optimization problem. These samples were used to generate the reduced order models for the state and control variables. However, the state and control reduced order models remained constant during optimization.

This work expands on previous seminal work on surrogate-based optimization and trust region POD through the development of a trust region algorithm for large-scale PDE-constrained optimization with bound constraints using reduced order modeling. The proposed algorithm applies the Galerkin-POD approach to generate a reduced order model that substitutes the high fidelity FEM during optimization. The adaptive updating strategy relies on a trust region framework and a quadratic model to detect loss of predictive accuracy in the reduced order model during optimization. Based on the ratio between the actual and predicted reduction in the objective function, the POD functions are automatically updated during optimization. This allows the algorithm to control the inexactness induced by the reduced order model and adaptively improve its predictive accuracy during optimization. Furthermore, the algorithm has dual purpose since it serves as both an optimization engine and a sampling algorithm. Thus, offline samples are not necessary to generate the reduced order model. The samples created by the algorithm are used to generate the reduced order model during optimization. The algorithm also employs a

projected gradient algorithm to model bound constraints [5]. This feature is crucial for engineering design as well as other applications in optimization. Is important to emphasize that the proposed algorithm is not specific to Galerkin-POD methods. Alternative reduced order modeling techniques can be successfully applied without any changes to the main algorithmic framework.

This paper also presents an accurate Hessian formulation for topology optimization problems. The proposed trust region algorithm relies on a quadratic model to predict the behavior of the objective function during the trust region sub-problem iterations. The quadratic model relies on accurate second order derivative information to predict the behavior of the objective function. If a nonlinear programming Hessian formulation is used, the computational effort will increase due to additional FEM evaluations during optimization. These additional FEM evaluation are necessary to compute the application of the trial step to the nonlinear Hessian operator. A linear Hessian formulation is proposed herein to minimize computational effort and thus expedite optimization. This linear Hessian formulation allows the algorithm to compute accurate second order derivative information without performing additional FEM evaluations.

This paper is organized as follows. A background on reduced space optimization formulation, topology optimization for compliance minimization problems, and POD methods is provided in Section 2. Section 3 describes the trust region algorithm for PDE-constrained optimization with bound constraints using reduced order modeling. Here, the trust region framework for bound constraint optimization and the adaptive POD functions adaptive updating scheme are described. Examples in topology optimization are presented in Section 4 and concluding remarks are provided in Section 5. Finally, the second order derivative operators used to compute the nonlinear and linear Hessian operator for compliance minimization topology optimization problems are provided in the Appendix.

2. Background

Let Z, U, and Y be Banach spaces, where both Z and U are reflexive, *i.e.* $z \sim z \; \forall \; z \in Z$ and $u \sim u \; \forall \; u \in U$. Furthermore, let $J \colon U \times Z \to \mathbb{R}$ and $g \colon U \times Z \to Y$. Lets now consider an optimization problem

$$\underset{(u,z) \; \in \; U \times Z}{\text{minimize}} \quad J(u,z) \quad \text{s.t.} \quad g(u,z) = 0, \tag{1}$$

where $u \in U_{ad} \subset U$ and $z \in Z_{ad} \subset Z$. U_{ad} and Z_{ad} denote admissible subsets of the state and control spaces, respectively. If the following conditions are met:

1. $Z_{ad} \subset Z$ is convex, bounded and closed;
2. $U_{ad} \subset U$ is convex, closed, and contains a feasible point; i.e. $g(u,z) = 0$ has a bounded solution operator, $\hat{u}(z) \colon Z \to U$;
3. the mapping $(u,z) \mapsto g(u,z)$ is continuous under weak convergence; and

4. J is sequentially lower semicontinuous;

there exists a solution to the optimization problem defined in Equation 1 [5, 27]. The above result ensures the existence of an optimal solution to the optimization problem defined in Equation 1. However, the uniqueness of the solution is problem dependent.

First order necessary optimality conditions and second order sufficient conditions are required to find the optimal solution of Equation 1 through Newton's method. These conditions involve the gradient of the objective function being zero at the optimal solution and the Hessian operator being positive semidefinite at the optimal solution. These conditions can be derived from Lagrangian multiplier theory [30].

2.1. Reduced Space Formulation

Lets define a Frèchet differentiable function $J\colon U \times Z \to \mathbb{R}$ and constraints $g(u,z)\colon U \times Z \to Y$ with Lipschitz continuous second derivatives. Then, the implicit function theorem admits the definition of a solution operator $\hat{u}\colon Z \to U$ such that $\{(\hat{u}(z), z) \mid z \in Z\} = \{(u,z) \in U \times Z \mid g(u,z) = 0\}$ [10]. This allows the redefinition of the optimization problem defined in Equation 1 as an unconstrained optimization problem of the form

$$\underset{z \in Z}{\text{minimize}} \quad J(\hat{u}(z), z), \tag{2}$$

where $\hat{u}(z)$ is obtained by solving $g(u(z), z) = 0$. In practice, this formulation is known as the reduced space formulation for PDE-constrained optimization.

Applying lagrange multiplier theory enables the definition of a Lagrangian functional $\mathcal{L}\colon U \times Z \times Y \to \mathbb{R}$ of the form

$$\mathcal{L}(\hat{u}(z), z, \lambda) = J(\hat{u}(z), z) + \langle \lambda, g(\hat{u}(z), z) \rangle_{Y^*, Y}, \tag{3}$$

where Y^* is the dual space of Y. The first order necessary optimality conditions are derived from Equation 3 to ensured optimality and are given by

$$\langle \mathcal{L}_u(\hat{u}(z), z, \lambda), \delta u \rangle = \langle J_u(\hat{u}(z), z) + \langle g_u(\hat{u}(z), z)^* \lambda, \delta u \rangle = 0 \tag{4}$$

$$\langle \mathcal{L}_z(\hat{u}(z), z, \lambda), \delta z \rangle = \langle J_z(\hat{u}(z), z) + \langle g_z(\hat{u}(z), z)^* \lambda, \delta z \rangle = 0, \tag{5}$$

where $\delta u \equiv u_z(z)\delta z$.

The Lagrange multipliers are computed by solving the adjoint system of equations obtained from Equation 4 as follows

$$\lambda = -(g_u(\hat{u}(z), z)^*)^{-1} J_u(\hat{u}(z), z). \tag{6}$$

Substituting Equation 6 into Equation 5 leads to the following reduced gradient operator

$$\nabla J(\hat{u}(z), z) = J_z(\hat{u}(z), z) - \langle g_z(\hat{u}(z), z)^*, (g_u(\hat{u}(z), z)^*)^{-1} J_u(\hat{u}(z), z) \rangle. \tag{7}$$

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

$$\langle \delta z, \nabla^2 J(\hat{u}(z^\star), z^\star)\delta z\rangle \geq \kappa \|\delta z\|^2 \ \forall \ \delta z \in \ \ker \ g_z(\hat{u}(z^\star), z^\star);$$

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

The application of the trial step δz to the Hessian operator is defined as

$$\nabla^2 J(\hat{u}(z), z)\delta z = \mathcal{L}_{zu}(\hat{u}(z), z, \lambda)\delta u + \mathcal{L}_{zz}(\hat{u}(z), z, \lambda)\delta z + \mathcal{L}_{z\lambda}(\hat{u}(z), z, \lambda)\delta\lambda, \tag{8}$$

where $\delta\lambda \in Y$. Notice that δu and $\delta\lambda$ are required to compute the application of the trial step to the Hessian operator. These quantities can be obtained by solving two independent system of equations that are derived for δu and $\delta\lambda$.

Let $g(\hat{u}(z), z) = 0 \ \forall \ z \in Z$. Then, $g_z(\hat{u}(z), z)\delta z = 0 \ \forall \ (z, \delta z) \in Z \times Z$, where

$$g_z(\hat{u}(z), z)\delta z = g_u(\hat{u}(z), z)\delta u + g_z(\hat{u}(z), z)\delta z = 0. \tag{9}$$

Solving Equation 9 gives

$$\delta u = -g_u(\hat{u}(z), z)^{-1} g_z(\hat{u}(z), z)\delta z. \tag{10}$$

Next, an explicit expression is derived for $\delta\lambda$ from Equation 4. By definition, $\mathcal{L}_u(\hat{u}(z), z, \lambda) = 0 \ \forall \ (u, z, \lambda) \in U \times Z \times Y$; thus, the derivative of $\mathcal{L}_u(\hat{u}(z), z, \lambda)$ in the direction of δz gives

$$\mathcal{L}_{uu}(\hat{u}(z), z, \lambda)\delta u + \mathcal{L}_{uz}(\hat{u}(z), z, \lambda)\delta z + \mathcal{L}_{u\lambda}(\hat{u}(z), z, \lambda)\delta\lambda = 0, \tag{11}$$

$\forall \quad (z, \delta u, \delta z, \delta\lambda) \in Z \times U \times Z \times Y$. Solving Equation 11 for $\delta\lambda$ yields

$$\delta\lambda = -\mathcal{L}_{u\lambda}(\hat{u}(z), z, \lambda)^{-1}[\mathcal{L}_{uu}(\hat{u}(z), z, \lambda)\delta u + \mathcal{L}_{uz}(\hat{u}(z), z, \lambda)\delta z], \tag{12}$$

where

$$\mathcal{L}_{u\lambda}(\hat{u}(z), z, \lambda) = g_u(\hat{u}(z), z)^* \tag{13}$$

$$\mathcal{L}_{uu}(\hat{u}(z), z, \lambda) = J_{uu}(\hat{u}(z), z) + g_{uu}(\hat{u}(z), z)^*\lambda \tag{14}$$

$$\mathcal{L}_{uz}(\hat{u}(z), z, \lambda) = J_{uz}(\hat{u}(z), z) + g_{uz}(\hat{u}(z), z)^*\lambda. \tag{15}$$

For reduced space optimization, the following sequence of steps are required at each optimization iteration to compute the application of the trial step to the Hessian operator

1. Solve equality $g(\hat{u}(z), z) = 0$ for $\hat{u}(z) \in U$
2. Solve $g_u(\hat{u}(z), z)^*\lambda = -J_u(\hat{u}(z), z)$ for $\lambda \in Y$, where $\{(\lambda, z) \mid z \in Z\} = \{(\lambda, z) \in Y \times Z \mid \mathcal{L}_u(u, z, \lambda) = 0\}$

3. Compute the reduced gradient operator

$$\nabla J(\hat{u}(z), z) = J_z(\hat{u}(z), z)^* + g_z(\hat{u}(z), z)^*(-g_u(\hat{u}(z), z))^{-*} J_z(\hat{u}(z), z)^*$$

4. Solve $g_u(\hat{u}(z), z)\delta u = g_z(\hat{u}(z), z)\delta z$ for $\delta u \in U$
5. Solve $g_u(\hat{u}(z), z)^*\delta\lambda = -[\mathcal{L}_{uu}(\hat{u}(z), z, \lambda)\delta u + \mathcal{L}_{uz}(\hat{u}(z), z, \lambda)\delta z]$ for $\delta\lambda \in Y$
6. Compute the application of the trial step δz to the reduced Hessian operator

$$\nabla^2 J(\hat{u}(z), z)\delta z = \mathcal{L}_{zu}(\hat{u}(z), z, \lambda)\delta u + \mathcal{L}_{zz}(\hat{u}(z), z, \lambda)\delta z + \mathcal{L}_{z\lambda}(\hat{u}(z), z, \lambda)\delta\lambda.$$

If first order optimization algorithms are applied to solve the unconstrained optimization problem defined in Equation 2, only steps 1 through 3 are necessary. Contrary, if second order optimization algorithms are applied, steps 1 through 6 are necessary.

2.2. Topology Optimization

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 $\mathbb{H} = L^2(\Omega; \mathbb{R}^{\mathfrak{n}})$ 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 $\mathbb{U} = \{\text{span}\{\phi_i^a\}_{a=1}^A \,|\, \phi_i \in \mathbb{H}\} \subset U$ for $i \in \{1, 2, 3\}$, $\mathbb{Z} = \{\text{span}\{\psi^b\}_{b=1}^B \,|\, \psi \in \mathbb{H}\} \subset Z$, and $\mathbb{Y} = \{\text{span}\{\chi_i^c\}_{c=1}^C \,|\, \chi_i \in \mathbb{H}\} \subset Y$. This enables the following finite dimensional approximations for the state, control, and lagrange multipliers $\mathbf{u} = \sum_a^A \vartheta_i^a \phi_i^a \,|\, \vartheta_i \in \mathbb{R}$, $\mathbf{z} = \sum_b^B \varrho^b \psi^b \,|\, \varrho \in \mathbb{R}$, and $\mathbf{v} = \sum_c^C \varsigma_i^c \chi_i^c \,|\, \varsigma_i \in \mathbb{R}$, respectively.

A topology optimization problem for compliance minimization can be defined as

$$\min_{\mathbf{z} \in \hat{\mathbb{Z}}} \quad J(\mathbf{u}(\mathbf{z}), \mathbf{z}) \tag{16}$$

for $\hat{\mathbb{Z}} = \{\mathbf{z} \in \mathbb{Z} : lb \leq \mathbf{z} \leq ub\}$ and objective function

$$J(\mathbf{u}(\mathbf{z}), \mathbf{z}) = \frac{\varepsilon_{\mathbf{K}}}{2}\langle\mathbf{u}, \mathbf{K}(\mathbf{z})\mathbf{u}\rangle_{\mathbb{H}} + \frac{\varepsilon_V}{2}\|V(\mathbf{z}) - V_0\|_{\mathbb{H}}^2 + R(\mathbf{z}). \tag{17}$$

The state $\mathbf{u} \in \mathbb{U}$ is obtained by solving the linear elastostatic equations

$$\mathbf{K}(\mathbf{z})\mathbf{u} = \mathbf{f} \quad \text{in} \quad \Omega \mid \mathbf{u} = 0 \text{ on } \partial\Omega_{\mathbf{u}}. \tag{18}$$

Here, $\partial\Omega_{\mathbf{u}} \subset \partial\Omega$ is the boundary where Dirichlet conditions are applied, $\mathbf{f}$ denotes an external excitation (force), and $\mathbf{K}$ denotes the stiffness matrix. The parameters lb and ub respectively denote the lower and upper bounds for the control $\mathbf{z}$. Recall from Section 2.1 that this formulation is known as the reduced space formulation for PDE-constrained optimization.

The modified solid isotropic material with penalization (SIMP) density-based method [35] was used to penalized the stiffness of the material. The modified SIMP formulation is given by

$$E(\mathbf{z}) = E_{\min} + \mathbf{z}^p(E_0 - E_{\min}) \quad | \quad E(\mathbf{z}) \colon \hat{\mathbb{Z}} \to \mathbb{R}, \tag{19}$$

where E_0 is the stiffness of the material, $0 < E_{\min} < 1$ is a small stiffness assigned to void regions to prevent the stiffness matrix from becoming singular, and p is a penalty parameter introduced to ensure "black-and-white solutions". The modified SIMP approach differs from the original SIMP formulation through the introduction of $E_{\min}$. In the classical SIMP approach, elements with zero stiffness are avoided only through the enforcement of the bound constraints during optimization. The reader is referred to [35] for a discussion on the advantages obtained with the modified SIMP formulation in topology optimization problems.

The functional $V(\mathbf{z})\colon \hat{\mathbb{Z}} \to \mathbb{R}$ in Equation 17 quantifies the volume of the structure and $0 < V_0 < 1$ is the target volume. The parameters $\varepsilon_{\mathbf{K}} \in \mathbb{R}_+^*$ and $\varepsilon_V \in \mathbb{R}_+^*$ are weights for the compliance and volume misfit objective terms, respectively. The following definition from [1, 13] was used for the regularization functional

$$R(\mathbf{z}) = \frac{\zeta}{\tau} \int_\Omega (\langle \nabla \mathbf{z}, \nabla \mathbf{z} \rangle_{\mathbb{H}} + \nu^2)^\tau. \tag{20}$$

This expression gives the flexibility to employ total variation regularization when ($\tau = 1$, $\nu = 0$) and a modified form of total variation regularization when ($\tau = 1/2, 0 < \nu \leq 1$). The parameter $0 < \zeta \leq 1$ is a penalty coefficient. In this work, total variation regularization was preferred over Tikhonov regularization due to its ability to capture sharp discontinuities in inverse problems settings [41].

The Lagrangian functional $\mathcal{L}\colon \mathbb{U} \times \hat{\mathbb{Z}} \times \mathbb{Y} \to \mathbb{R}$ for the compliance minimization topology optimization problem, Equation 16, is given by

$$\mathcal{L}(\mathbf{u}, \mathbf{z}, \mathbf{v}) = J(\mathbf{u}(\mathbf{z}), \mathbf{z}) + \langle \mathbf{v}, \mathbf{K}\mathbf{u} - \mathbf{f} \rangle_{\mathbb{Y}^*, \mathbb{Y}}. \tag{21}$$

The first order necessary optimality conditions for Equation 21 are given by

$$\mathcal{L}_{\mathbf{u}}(\mathbf{u}, \mathbf{z}, \mathbf{v}) = \varepsilon_{\mathbf{K}} \mathbf{K}(\mathbf{z})\mathbf{u} + \mathbf{K}(\mathbf{z})\mathbf{v} = 0 \tag{22}$$

$$\mathcal{L}_{\mathbf{z}}(\mathbf{u}, \mathbf{z}, \mathbf{v}) = \frac{\varepsilon_{\mathbf{K}}}{2} \mathbf{u}\, \mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u} + \mathbf{v}\, \mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u} + \varepsilon_V (V(\mathbf{z}) - V_0) V_{\mathbf{z}}(\mathbf{z}) + R_{\mathbf{z}}(\mathbf{z}) = 0, \tag{23}$$

where the subscripts $\mathbf{u}$ and $\mathbf{z}$ respectively denote derivatives with respect to the state and control. The derivative operator $R_{\mathbf{z}}(\mathbf{z})\colon \hat{\mathbb{Z}} \to \mathbb{Z}$ is given by Equation 58, see Appendix.

For compliance minimization problems, the lagrange multipliers are given by

$$\mathbf{v} = -\varepsilon_{\mathbf{K}} \mathbf{u}. \tag{24}$$

This explicit expression is derived from Equation 22 and it is due to the self-adjoint property of $\mathbf{K}$. Substituting Equation 24 into Equation 23 yields a reduced gradient operator of the form

$$\nabla J(\mathbf{u}(\mathbf{z}), \mathbf{z}) = -\frac{\varepsilon_{\mathbf{K}}}{2} \mathbf{u}\, \mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u} + \varepsilon_V (V(\mathbf{z}) - V_0) V_{\mathbf{z}}(\mathbf{z}) + R_{\mathbf{z}}(\mathbf{z}). \tag{25}$$

Equation 24 enables the computation of the reduced gradient operator without solving the computationally intensive adjoint problem needed in PDE-constrained optimization problems.

The quadratic model

$$m(\mathbf{w}_k) = \langle \nabla J(\mathbf{u}(\mathbf{z}_k), \mathbf{z}_k), \mathbf{w}_k \rangle_{\mathbb{H}} + \frac{1}{2} \langle \mathbf{w}_k, \nabla^2 J(\mathbf{u}(\mathbf{z}_k), \mathbf{z}_k), \mathbf{w}_k \rangle_{\mathbb{H}}. \tag{26}$$

is evaluated at every *k-th* trust region sub-problem iteration to decide if the trial control $\mathbf{z}$ is accepted. Notice that the Hessian operator, or an approximation to the Hessian operator, is required to evaluate Equation 26. To accurately compute the application of the trial step $\mathbf{w}$ to the nonlinear Hessian operator $\nabla^2 J(\mathbf{u}(\mathbf{z}_k), \mathbf{z}_k)$, Equations 10 and 12 are solved during optimization. This will increase the computational effort because the application of the trial step to the Hessian operator is carry out several times during a trust region sub-problem iteration.

To minimize the computational expense associated with the solution of Equations 10 and 12, the Hessian operator can be approximated using quasi-Newton methods [5, 11, 27]. Quasi-Newton methods have been successfully used in many applications. However, in this work, the second order derivative information is approximated by a linear programming Hessian formulation of the form

$$\nabla^2 J(\mathbf{u}(\mathbf{z}), \mathbf{z}) \equiv \mathcal{L}_{\mathbf{zz}}(\mathbf{u}, \mathbf{z}, \mathbf{v}) = J_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}), \mathbf{z}) + g_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}), \mathbf{z})\mathbf{v}, \tag{27}$$

where Equation 27 is given by

$$\nabla^2 J(\mathbf{u}(\mathbf{z}), \mathbf{z}) = -\frac{\varepsilon_{\mathbf{K}}}{2} \mathbf{u}\, \mathbf{K}_{\mathbf{zz}}(\mathbf{z})\mathbf{u} + \varepsilon_V V_{\mathbf{z}}(\mathbf{z}) V_{\mathbf{z}}(\mathbf{z}) + \varepsilon_V (V(\mathbf{z}) - V_0) V_{\mathbf{zz}}(\mathbf{z}) + R_{\mathbf{zz}}(\mathbf{z}). \tag{28}$$

This linear approximation does not require the solution of Equations 10 and 12 and thus reduces the computational cost associated with this calculation. Furthermore, this approach avoids the storage of additional snapshots ensembles as well as symmetric eigenvalue solves, which further expedites optimization. Results will show that significant speedups are attained by using the linear Hessian approximation for the solution of Equation 16 over its nonlinear counterpart.

2.3. Proper Orthogonal Decomposition

Let $\widehat{\mathbf{u}} = \{\mathbf{u}_p\}_{p=1}^{N_s}$ denote the snapshot ensemble that contains characteristics of the expected state solution, where N_s denotes the total number of snapshots. This enables the definition of the finite dimensional linear space $\hat{\mathbb{U}} = \text{span}\{\mathbf{u}_p\}_{p=1}^{N_s} \subset \mathbb{H}$, where $\dim(\hat{\mathbb{U}}) \leq N_s$. Let $\Phi = \{(\varphi_p)_q\}_{q=1}^{\ell} \mid \varphi_p \in \mathbb{H}$ denote the eigenvector ensemble associated to the state ensemble $\widehat{\mathbf{u}}$. The POD basis can then be computed by solving the following equality constrained optimization problem

$$\underset{\Phi_q \in \mathbb{H}}{\text{minimize}} \quad \sum_{p=1}^{N_s} \|\widehat{\mathbf{u}}_p - \sum_{q=1}^{\ell} \langle \widehat{\mathbf{u}}_p, \Phi_q \rangle_{\mathbb{H}} \Phi_q\|_{\mathbb{H}}^2 \quad \text{s.t.} \quad \langle \Phi_q, \Phi_r \rangle_{\mathbb{H}} = \delta_{qr}, \tag{29}$$

where $\ell \leq \dim(\hat{\mathbb{U}})$ and $1 \leq r \leq \ell$. The solution to Equation 29 yields the POD basis that is used to generate the reduced order model during optimization.

The optimization problem defined in Equation 29 is equivalent to solving a symmetric eigenvalue problem or a singular value problem [17, 38]. During optimization, based on the ratio between the actual and predicted reduction, the POD basis is constructed by solving a symmetric eigenvalue problem of the form

$$\Psi_{pq}\Phi_q = \sum_{p=1}^{N_s} \langle \widehat{\mathbf{u}}_p, \Phi_q \rangle \widehat{\mathbf{u}}_p = \Upsilon_q \Phi_q, \quad 1 \leq q \leq \ell, \tag{30}$$

where $\Upsilon_q \in \mathbb{R}$ denotes the real-value eigenvalue associated with eigenvector $\Phi_q \in \mathbb{H}$, $\Psi = \widehat{\mathbf{u}}\,\widehat{\mathbf{u}}^{\intercal} \colon \mathbb{H} \to \hat{\mathbb{U}}$ is a bounded linear, compact, self-adjoint and positive operator. The operator Ψ is commonly known as the covariance operator.

Lets define a finite dimensional linear space $\widehat{\mathbb{W}} = \text{span}\{(\widehat{\varphi}_i)_q\}_{q=1}^{N_\varphi} \subset \mathbb{H}$ for $1 \leq N_\varphi \leq \ell$ and a set of real-value POD coefficients $\Theta = \{(\theta_i)_q\}_{q=1}^{N_\varphi} \mid \theta_i \in \mathbb{R}$. Then, the POD method seeks to develop an approximation to the state solution by finding a finite and low dimensional representation $\bar{\mathbf{u}} \in \widehat{\mathbb{W}}$ of the form

$$\bar{\mathbf{u}} = \sum_{q=1}^{N_\varphi} (\theta_i)_q (\widehat{\varphi}_i)_q, \tag{31}$$

where the POD function $(\widehat{\varphi}_i)_q \in \widehat{\mathbb{W}}$ is given by

$$(\widehat{\varphi}_i)_q = \Upsilon_q^{-1/2} \sum_{p=1}^{N_s} (\mathbf{u}_i)_p (\varphi_p)_q \quad \text{for} \quad 1 \leq q \leq N_\varphi. \tag{32}$$

This POD function ensemble is used to effectively build accurate reduced order models during optimization. This reduces the number of high fidelity FEM evaluations during optimization and thus accelerates optimization.

3. Algorithm

The trust region algorithm for PDE-constrained optimization with bound constraints using reduced order modeling is presented in this section. The trust region Newton's version of the proposed algorithm was presented by Lin [23]. This works adapts the trust region algorithm presented by Lin *et.al.* to PDE-constrained optimization problems where the PDE is modeled using reduced order modeling techniques. Specifically, the POD-Galerkin method is used in this work to construct a reduced order model for the high fidelity FEM during optimization. However, the proposed algorithm is not specific to POD-Galerkin methods. The proposed trust region algorithm can be effectively implemented to enable

the used of other reduced order modeling techniques. The main algorithmic framework is described in Algorithm **A.1**.

Initialization. Given $\mathbf{z}_{k=0}$ compute $J(\mathbf{u}(\mathbf{z}_{k=0}), \mathbf{z}_{k=0})$ and $\nabla J(\mathbf{u}(\mathbf{z}_{k=0}), \mathbf{z}_{k=0})$ and store initial state solution $\widehat{\mathbf{u}}_{p=1} = \mathbf{u}(\mathbf{z}_{k=0})$
while *not converged* **do**
  Solve trust region sub-problem as described in A.2.
  Update orthogonal basis as described in A.3.
  if $M \leq N$
    Compute $\nabla J(\mathbf{u}(\mathbf{z}_{k+1}), \mathbf{z}_{k+1})$
  else If $M > N$ **and** $\rho_k \leq \upsilon$
    Compute $\nabla J(\mathbf{u}(\mathbf{z}_{k+1}), \mathbf{z}_{k+1})$
  else
    Compute $\nabla J(\bar{\mathbf{u}}(\mathbf{z}_{k+1}), \mathbf{z}_{k+1})$
  Check convergence.
  $k = k + 1$
end

Algorithm 1: Main framework

3.1. Trust region sub-problem

At each *n-th* iteration of a trust region algorithm there is an approximation to the control $\mathbf{z}_n \in \hat{\mathbb{Z}}$, a trust region radius Δ_n, and a model $m(\mathbf{s}_n)\colon \mathbb{Z} \to \mathbb{R}$ of the possible reduction in the objective function defined by $J(\mathbf{u}(\mathbf{z}_{n+1}), \mathbf{z}_{n+1}) - J(\mathbf{u}(\mathbf{z}_n), \mathbf{z}_n)$. Here, $\mathbf{z}_{n+1} = \mathbf{z}_n + \mathbf{s}_n$, such that $\|\mathbf{s}_n\| \leq \Delta_n$, $\mathbf{s}_n \in \mathbb{Z}$. The proposed algorithm applies a quadratic model of the form

$$m(\mathbf{s}_n) = \langle \nabla J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0}), \mathbf{s}_n \rangle_{\mathbb{H}} + \frac{1}{2} \langle \mathbf{s}_n, \nabla^2 J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0}), \mathbf{s}_n \rangle_{\mathbb{H}},$$

which was first introduced in Equation 26.

The control $\mathbf{z}_n$ and trust region radius Δ_n are updated through standard rules in trust region methods for unconstrained optimization [9]. Thus, given a trial step $\mathbf{w}_n$ and a trial control $\mathbf{z}_{n+1} = \mathbf{z}_{n=0} + \mathbf{s}_n$, the ratio between the actual reduction in the objective function and the predicted reduction in the quadratic model

$$\rho_n = \frac{J(\mathbf{u}(\mathbf{z}_{n+1}), \mathbf{z}_{n+1}) - J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})}{m(\mathbf{s}_n)} \tag{33}$$

is evaluated. A trial step $\mathbf{s}_n$ with $\rho_n > 0$ yields a reduction in the objective function. The trial control $\mathbf{z}_{n+1} = \mathbf{z}_{n=0} + \mathbf{s}_n$ is accepted depending on how well the quadratic model $m(\mathbf{s}_n)$ approximates the actual reduction in the objective function within a suitable neighborhood

or trust region. Given acceptance criterion $\eta_1 \in \mathbb{R}_+^*$, $\mathbf{z}_{n+1}$ is accepted if $\rho_n > \eta_1$.If $\rho_n \leq \eta_1$, $\mathbf{z}_{n+1}$ is rejected, the trust region radius is updated, and a new trial step is computed. The rules used to update the trust region radius Δ_n are given by

$$\begin{aligned} &\Delta_{n+1} \in [\sigma_1 \min\{\|\mathbf{s}_n\|, \Delta_n\}, \sigma_2 \Delta_n] && \text{if} \quad \rho_n \leq \eta_1, \\ &\Delta_{n+1} \in [\sigma_1 \Delta_n, \sigma_3 \Delta_n] && \text{if} \quad \rho_n \in (\eta_1, \eta_2), \\ &\Delta_{n+1} \in [\Delta_n, \sigma_3 \Delta_n] && \text{if} \quad \rho_n \geq \eta_2 \end{aligned} \tag{34}$$

where $\eta_1, \eta_2 \in \mathbb{R}_+^*$ and $0 < \sigma_1 < \sigma_2 < 1 < \sigma_3$.

A trial step $\mathbf{s}_n$ is selected to give as much reduction in the quadratic model as the Cauchy step $\mathbf{w}_n^C(\alpha)\colon \mathbb{R} \to \mathbb{Z}$ generated by a projected gradient method

$$\mathbf{w}_n^C(\alpha) = \mathbf{P}_{\hat{\mathbb{Z}}}[\mathbf{z}_n - \alpha \nabla J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})] - \mathbf{z}_{n=0}, \tag{35}$$

where $\mathbf{P}_{\hat{\mathbb{Z}}}\colon \mathbb{Z} \to \hat{\mathbb{Z}}$ is the projection to the feasible set. The projection operator is given by

$$\mathbf{P}_{\hat{\mathbb{Z}}}(\mathbf{z}_\imath) = \begin{cases} lb_\imath & \text{if} \quad \mathbf{z}_\imath \leq lb_\imath \\ \mathbf{z}_\imath & \text{if} \quad lb_\imath < \mathbf{z}_\imath < ub_\imath \\ ub_\imath & \text{if} \quad \mathbf{z}_\imath \geq ub_\imath \end{cases} \tag{36}$$

for $1 \leq \imath \leq \mathsf{n}_{\mathbf{z}}$. The scalar $\alpha \in \mathbb{R}_+^*$ is selected such that

$$m(\mathbf{w}_n^C(\alpha)) \leq \mu_0 \langle \nabla J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0}), \mathbf{w}_n^C(\alpha) \rangle_{\mathbb{H}}, \ \|\mathbf{w}_n^C(\alpha)\| \leq \mu_1 \Delta_n, \tag{37}$$

where $\mu_0 < \frac{1}{2}$ and $\mu_1 \in \mathbb{R}_+^*$. The criteria in Equation 37 can be satisfied in a finite number of iterations [7, 23, 25].

Given the Cauchy step $\mathbf{w}_n^C \equiv \mathbf{w}_n^C(\alpha)$, a descent direction $\mathbf{w}_n$ is selected by solving the trust region sub-problem

$$\underset{\mathbf{w}_n \in \mathbb{Z}}{\text{minimize}} \ m(\mathbf{w}_n) : \mathbf{w}_\imath = 0, \ \imath \in \mathcal{A}(\mathbf{z}), \ \|\mathbf{S}\mathbf{w}_n\| \leq \Delta_n, \tag{38}$$

where $\mathcal{A}(\mathbf{z})$ is the set of active constraints at $\mathbf{z} \in \hat{\mathbb{Z}}$ defined by

$$\mathcal{A}(\mathbf{z}) = \{\imath \in \mathcal{I} : \mathbf{z}_\imath \in \{lb_\imath, ub_\imath\}\} \tag{39}$$

and $\mathbf{S}$ is defined as a diagonal matrix whose diagonal entry is set to one if the constraint is inactive. The trust region sub-problem is solved using the preconditioned conjugate gradient algorithm proposed by Steihaug [39]. However, a proper implementation of the trust region framework described in Algorithm **A.2** enables the used of other Krylov solvers [31]. Furthermore, different trust region methods (dogleg [29] or double dogleg [12]) can be easily applied to solve the trust region sub-problem in Equation 38.

Data: **If** $M \leq N$ **then** low fidelity model is inactive **else** low fidelity model is active
Result: Updated control $\mathbf{z}_{k+1}$
Step 0: Set current control $\mathbf{z}_{n=0} = \mathbf{z}_k$ and initial trust region radius $\Delta_{n=0} = \Delta_k$
while $\rho \leq \eta_1$ **do**
Step 1: Compute Cauchy step.

$$\mathbf{w}_n^C(\alpha) = P[\mathbf{z}_n - \alpha \nabla J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})]$$

Step 2: Compute descent direction.

$$\underset{\substack{\mathbf{w_n} \in \mathbb{Z} \\ \|\mathbf{S}\mathbf{w}_n\| \leq \Delta_n}}{\text{minimize}}\ m(\mathbf{w_n}) \equiv \langle \nabla J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0}), \mathbf{S}\mathbf{w}_n \rangle + \tfrac{1}{2} \langle \mathbf{w}_n \mathbf{S}, \nabla^2 J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0}) \mathbf{S}\mathbf{w}_n \rangle$$

Step 3: Compute trial step.

$$\mathbf{s}_n \equiv \mathbf{w}_n(\beta) = P[\mathbf{z}_n + \beta \mathbf{w}_n] - \mathbf{z}_{n=0}$$

such that $m(\mathbf{s}_n) \leq \mu_0\, m(\mathbf{w}_n^C)$, $\|\mathbf{s}_n\| \leq \Delta_n$, and $\mathbf{z}_{n+1} = \mathbf{z}_{n=0} + \mathbf{s}_n \in \hat{\mathbb{Z}}$
Step 4: Acceptance of the trial control.

$$\rho_n = \frac{J(\mathbf{u}(\mathbf{z}_{n+1}), \mathbf{z}_{n+1}) - J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})}{m(\mathbf{s}_n)},$$

if $\rho_n > \eta_1$ **then** $\mathbf{z}_{k+1} = \mathbf{z}_{n+1}$ **else** $\mathbf{z}_{n+1} = \mathbf{z}_{n=0}$
Step 5: Trust region radius update.

$$\Delta_{n+1} = \begin{cases} \Delta_{n+1} \in [\sigma_1 \min\{\|\mathbf{w}_n\|, \Delta_n\}, \sigma_2 \Delta_n] & \text{if} \quad \rho_n \leq \eta_1 \\ \Delta_{n+1} \in [\sigma_1 \Delta_n, \sigma_3 \Delta_n] & \text{if} \quad \rho_n \in (\eta_1, \eta_2) \\ \Delta_{n+1} \in [\Delta_n, \sigma_3 \Delta_n] & \text{if} \quad \rho_n \geq \eta_2 \end{cases}$$

$n = n + 1$
end

Algorithm 2: Trust region sub-problem

Given the descent direction $\mathbf{w}_n$, a projected gradient line search is done to compute a trial step $\mathbf{s}_n \equiv \mathbf{w}_n(\beta) = \mathbf{z}_+ - \mathbf{z}_{n=0}$ that guarantees that

$$\mathbf{z}_+ = \mathbf{P}[\mathbf{z}_n + \beta \mathbf{w}_n] \in \hat{\mathbb{Z}}, \quad \|\mathbf{s}_n\| \leq \mu_1 \Delta_n, \quad \mathbf{z}_{n+1} \in \hat{\mathbb{Z}} \tag{40}$$

and

$$m(\mathbf{s}_n) \leq m(\mathbf{z}_+^C - \mathbf{z}_{n=0}) + \mu_0 \min\{\nabla \langle m(\mathbf{z}_+^C - \mathbf{z}_{n=0}), \mathbf{s}_n \rangle_{\mathbb{H}}, 0\} \tag{41}$$

are satisfied. In Equation 41, $\mathbf{z}_+^C = \mathbf{P}[\mathbf{z}_n + \mathbf{w}_n^C(\alpha)]$. The reader is referred to [26] for a detailed discussion on the projected searches. Given trial step $\mathbf{s}_n$, the ratio between the actual and predicted reduction is computed to determine if the trial control $\mathbf{z}_{n+1}$ is accepted, if $\rho_n > \eta_1$ then $\mathbf{z}_{k+1} = \mathbf{z}_{n+1}$ else $\mathbf{z}_{n+1} = \mathbf{z}_{n=0}$. The trust region radius is updated according to the rules presented in Equation 34. Algorithm **A.2** describes the trust region

sub-problem framework.

Data: $\widehat{\mathbf{u}}$, $\widehat{\Lambda}$ ρ, ϵ_u, ϵ_λ
Result: $(\theta_i)_q$, $(\widehat{\varphi}_i)_q$, $(\theta_i^\lambda)_q$, $(\widehat{\varphi}_i^\lambda)_q$
if $M = N$ **then**
  Construct POD functions for state and lagrange multipliers.
    1. Solve $\Psi_{pq}\Phi_q = \Upsilon_q\Phi_q$ and $\Psi_{pq}^\lambda\Phi_q^\lambda = \Upsilon_q^\lambda\Phi_q^\lambda$
    2. Truncate orthogonal basis based on Equation 42
    3. Construct POD functions $\bar{\mathbf{u}} = \sum_{q=1}^{N_\varphi}(\theta_i)_q(\widehat{\varphi}_i)_q$ and $\mathbf{\Lambda} = \sum_{q=1}^{N_\varphi^\lambda}(\theta_i^\lambda)_q(\widehat{\varphi}_i^\lambda)_q$
else
  if $\rho \leq \upsilon$ **then**
    Solve high fidelity state and adjoint system of equations.
      1. Solve $g(\mathbf{u}(\mathbf{z}_{k+1}), \mathbf{z}_{k+1}) = 0$ for $\mathbf{u}(\mathbf{z}_{k+1}) \in \mathbb{U}$
      2. Solve $g_{\mathbf{u}}(\mathbf{u}(\mathbf{z}_{k+1}), \mathbf{z}_{k+1})^*\mathbf{v}_{k+1} = -J_{\mathbf{u}}(\mathbf{u}(\mathbf{z}_{k+1}), \mathbf{z}_{k+1})$ for $\mathbf{v}_{k+1} \in \mathbb{Y}$
    Update state and lagrange multipliers snapshot ensembles.
      Set $\widehat{\mathbf{u}}_{p=N_s+1} = \mathbf{u}(\mathbf{z}_{k+1})$ and $\widehat{\mathbf{\Lambda}}_{p=N_s+1} = \mathbf{v}_{k+1}$; update $N_s = p$
    Construct POD functions for state and lagrange multipliers.
      1. Solve $\Psi_{pq}\Phi_q = \Upsilon_q\Phi_q$ and $\Psi_{pq}^\lambda\Phi_q^\lambda = \Upsilon_q^\lambda\Phi_q^\lambda$
      2. Truncate orthogonal basis based on Equation 42
      3. Construct POD functions $\bar{\mathbf{u}} = \sum_{q=1}^{N_\varphi}(\theta_i)_q(\widehat{\varphi}_i)_q$ and $\mathbf{\Lambda} = \sum_{q=1}^{N_\varphi^\lambda}(\theta_i^\lambda)_q(\widehat{\varphi}_i^\lambda)_q$
  end
end

Algorithm 3: POD functions update

3.2. Adaptive POD functions update

The proposed trust region algorithm for PDE-constrained optimization with bound constraints described in Section 3.1 applies model reduction techniques to construct a faster, low fidelity FEM during optimization. The goal is to apply model reduction techniques to accelerate the solution of large-scale PDE-constrained optimization problems and reduce computational time. The algorithm stores state and lagrange multipliers snapshots from previous high fidelity FEM evaluations and applies a POD-Galerkin projection method to generate reduced order models for the state and lagrange multipliers. The proposed algorithm eliminates ineffective sampling of state and lagrange multipliers snapshots before the solution of the optimization problem (**offline**). Contrary, the algorithm relies on the trust region framework to effectively sample state and lagrange multipliers snapshots during op-

timization (**online**) and construct/update the reduced order models. The conjecture is that the optimization algorithm will generate optimal data sets during optimization due to the explicit enforcement of desired requirements through the problem formulation. Thus, the snapshots generated by the optimization algorithm will produce optimal reduced order models.

At each successful *k-th* iteration of the trust region algorithm, the ratio between the actual and predicted reduction in the objective function, ρ_k, is computed. This ratio is used to inform the algorithm how well the reduced order model is performing during optimization. Base on the ratio between the actual and predicted reduction in the objective function, the algorithm applies an adaptive POD function updating scheme to enhance the predictive accuracy of the reduced order model during optimization.

Given an initial number of snapshots N_s, the proposed trust region algorithm relies on high fidelity FEM evaluations to compute state and lagrange multipliers snapshots while the algorithm advances towards the optimal control. These high fidelity state and lagrange multipliers snapshots are stored **online** until the maximum number of snapshots is reached. Once the initial state and lagrange multipliers ensembles have been computed, the algorithm generates reduced order models for the state and lagrange multipliers. The advantage of the proposed approach is that high fidelity FEM evaluations are not wasted **offline**. Thus, high fidelity state and lagrange multipliers snapshots are computed **online** as the algorithm advances towards the optimal control. The proposed trust region algorithm for PDE-constrained optimization using reduced order modeling completely eliminates time consuming state and lagrange multipliers snapshot sampling **offline**.

The proposed algorithm relies on ρ_k to determine if the reduced order models need to be updated. If $\rho_k \leq \upsilon$, the reduced order models are not accurate for optimization and a new set of high fidelity state and lagrange multipliers snapshots are sampled by solving

$$g(\mathbf{u}(\mathbf{z}_{k+1}), \mathbf{z}_{k+1}) = 0 \text{ for } \mathbf{u}(\mathbf{z}_{k+1}) \in \mathbb{U}$$

and

$$g_{\mathbf{u}}(\mathbf{u}(\mathbf{z}_{k+1}), \mathbf{z}_{k+1})^* \mathbf{v}_{k+1} = -J_{\mathbf{u}}(\mathbf{u}(\mathbf{z}_{k+1}), \mathbf{z}_{k+1}) \text{ for } \mathbf{v}_{k+1} \in \mathbb{Y}.$$

These new high fidelity state $\mathbf{u}(\mathbf{z}_{k+1})$ and lagrange multipliers $\mathbf{v}_{k+1}$ snapshots are added to their respective ensembles $\widehat{\mathbf{u}}_{p=N_s+1} = \mathbf{u}(\mathbf{z}_{k+1})$ and $\widehat{\mathbf{\Lambda}}_{p=N_s+1} = \mathbf{v}_{k+1}$. The new ensembles $\widehat{\mathbf{u}}_{p=N_s+1}$ and $\widehat{\mathbf{\Lambda}}_{p=N_s+1} = \mathbf{v}_{k+1}$ are used to compute a new set of eigenvalues and eigenvectors by solving the symmetric eigenvalue problems

$$\Psi_{pq}\Phi_q = \Upsilon_q \Phi_q$$

and

$$\Psi^{\lambda}_{pq}\Phi^{\lambda}_q = \Upsilon^{\lambda}_q \Phi^{\lambda}_q.$$

Here, $\upsilon \in \mathbb{R}$ is a pre-defined threshold used to determine if the reduced order models are accurate for optimization. The parameter $\Upsilon^{\lambda}_q \in \mathbb{R}$ denotes the real-value eigenvalue associated with eigenvector $\Phi^{\lambda}_q \in \mathbb{H}$. $\Psi^{\lambda} = \widehat{\mathbf{\Lambda}}\,\widehat{\mathbf{\Lambda}}^{\intercal} \colon \mathbb{H} \to \hat{\mathbb{Y}}$ is the covariance operator for the

lagrange multipliers, where $\hat{\mathbb{Y}} = \text{span}\{\mathbf{v}_p\}_{p=1}^{N_s} \subset \mathbb{H}$. This approach allows the algorithm to adaptively update the state and lagrange multipliers eigenvalues and eigenvectors during optimization.

Step 4: Acceptance of the trial control.

1. Compute $J(\mathbf{u}(\mathbf{z}_n), \mathbf{z}_n)$ and

$$\rho_n^1 = \frac{J(\mathbf{u}(\mathbf{z}_n), \mathbf{z}_n) - J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})}{m(\mathbf{s}_n) - \sigma_r}$$

$$\rho_n^2 = \frac{J(\mathbf{u}(\mathbf{z}_n), \mathbf{z}_n) - J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})}{m(\mathbf{s}_n)},$$

   and set $\rho_n = \max[\rho_n^1, \rho_n^2]$
   if $\rho_n \leq \eta_1$
      Set $\mathbf{z}_{n+1} = \mathbf{z}_{n=0}$ and go to **Step 5**
   else
      Set $\mathbf{z}_{k+1} = \mathbf{z}_n$ and update $\sigma_r = m(\mathbf{s}_n) - \sigma_r$ and $\sigma_c = m(\mathbf{s}_n) - \sigma_c$
   end
2. Update best objective function value
   if $J(\mathbf{u}(\mathbf{z}_n), \mathbf{z}_n) < J_{\min}$
      Set $J_c = J_{\min}$, $\sigma_c = 0$, and $\hat{\jmath} = 0$ and go to **Step 5**
   else
      $\hat{\jmath} = \hat{\jmath} + 1$
   end
3. Update J_c for the next reference objective function value
   if $J(\mathbf{u}(\mathbf{z}_n), \mathbf{z}_n) > J_c$
      Set $J_c = J(\mathbf{u}(\mathbf{z}_n), \mathbf{z}_n)$ and $\sigma_c = 0$
   end
4. Reset reference objective function value if necessary
   if $\hat{\jmath} = M$
      Set $J_r = J_c$ and $\sigma_r = \sigma_c$
   end

Algorithm 4: Acceptance criteria for a non-monotone trust region algorithm

The number of state eigenvectors (N_φ) and lagrange multipliers eigenvectors (N_φ^λ) conserved and used to generate the POD-Galerkin basis is based on the amount of energy captured by each basis function. This energy measures are computed from the eigenvalue

ensembles $\{\Upsilon_q\}_{q=1}^{\ell}$ and $\{\Upsilon_q^{\lambda}\}_{q=1}^{\ell_\lambda}$ These energy measures are given by

$$\epsilon_u = \frac{\sum_{q=1}^{N_\varphi} \Upsilon_q}{\sum_{q=1}^{\ell} \Upsilon_q}, \qquad \epsilon_\lambda = \frac{\sum_{q=1}^{N_\varphi^\lambda} \Upsilon_q^\lambda}{\sum_{q=1}^{\ell_\lambda} \Upsilon_q^\lambda}, \tag{42}$$

where ϵ_u and ϵ_λ are the respective energy measures for the state and lagrange multipliers ensembles. It is often observed that a limited number of eigenvalues and eigenvectors are necessary to compute the Galerkin-POD functions use to generate the corresponding reduced order models.

Given finite dimensional linear spaces $\widehat{\mathbb{W}} = \text{span}\{(\widehat{\varphi}_i)_q\}_{q=1}^{N_\varphi} \subset \mathbb{H}$ and $\widehat{\mathbb{W}}_\lambda = \text{span}$ $\{(\widehat{\varphi}_i^\lambda)_q\}_{q=1}^{N_\varphi^\lambda} \subset \mathbb{H}$, the Galerkin-POD function ensembles $\bar{\mathbf{u}}$ and $\mathbf{\Lambda}$ are computed by applying Equations 31 and 32. Is important to emphasize again that the proposed algorithm is not specific to POD-Galerkin projection methods. Other projection methods can be effectively used to generate the reduced order models **online**. The key is to design a general algorithmic framework that enables the used of different projection methods for model reduction. Algorithm **A.3** describes the adaptive Galerkin-POD updating scheme used by the algorithm to control the inexactness induced by the reduced order models during optimization.

3.3. Algorithmic Extensions

The trust region sub-problem algorithm proposed in Section 3.1 can be modified to accept non-monotone steps. The advantage of this strategy is that the algorithm can avoid getting stranded in local minima when solving highly nonlinear programming problems. The idea of non-monotone trust region algorithms is to modify the update strategy in order to accept trial controls even if descent in the objective function is not attained, *i.e.* $\rho_k < 0$.

Let $J_{\min}$ denote the current best value of the objective function at iteration n, that is

$$J_{\min} = \min_{\hat{\jmath} \in \{0,n\}} J_{\min}(\mathbf{u}(\mathbf{z}_{\hat{\jmath}}), \mathbf{z}_{\hat{\jmath}}) \tag{43}$$

Lets define $\hat{\jmath}$ successful iterations since $J_{\min}$ was first computed. The reference objective function value J_r is updated if $\hat{\jmath}$ exceeds a predefined constant integer M. The reference objective function value J_r is reset to the largest value observed over all previous successful iterations, J_c, since the last best objective function value $J_{\min}$ was last found. Before introducing the modified trust region algorithm, lets define parameters σ_c and σ_r. Here, σ_c denote the sum of predicted model decrease on all previous successful iterations; while, σ_r denotes the sum of predicted model decrease on all previous successful iteration since the reference iteration.

The non-monotone trust region algorithm is a special case of Algorithm **A.2**. To allow a non-monotone trust region iteration, two modifications to Algorithm **A.2** are necessary. First, **Step 0** is modified to allow the initialization of $J_{\min} = J_r = J_c = J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})$

Parameter	Value
E_0	1 Pa.
$E_{\min}$	1×10^{-6} Pa.
p	3
ν	1×10^{-8}
τ	0.5
ζ	0.5*max(L_e)
v_f	0.3
$\mathbf{z}_0$	$\text{diag}(v_f) \cdot \mathbf{1}$
ε_V	$1/V(\mathbf{z}_0)$
$\varepsilon_{\mathbf{K}}$	$1/(\mathbf{u}(\mathbf{z}_0)\mathbf{K}(\mathbf{z}_0)\mathbf{u}(\mathbf{z}_0))$

Table 1: Corresponding values for the topology optimization parameters.

and $\sigma_r = \sigma_c = 0$ in addition to $\mathbf{z}_{n=0} = \mathbf{z}_k$ and $\Delta_{n=0} = \Delta_k$. Finally, the acceptance criteria for the trial control given in **Step 4** for a non-monotone trust region algorithm is modified and described in Algorithm **A.4**.

In this work, the non-monotone trust region step algorithm was more effective than the monotone counterpart solving Equation 16. The non-monotone strategy required less iterations and objective function evaluations than the monotone strategy. This reduce the number of high fidelity FEM evaluations and thus expedite optimization. Finally, notice that the proposed trust region algorithm for PDE-constrained optimization with bound constraints using reduced order modeling, if properly implemented, can easily apply other trust region sub-problem algorithms without major changes to the main framework. Once more, the key is the proper implementation of the algorithm.

Algorithm **A.2** can be modified to be used as a non-monotone trust region algorithm by limiting the number of trust region sub-problem iterations. The computed trial control is then accepted without enforcing descent in the objective function. This simple modification enables the use of Algorithm **A.2** as a non-monotone trust region step. The interested reader is referred to [9] for detailed discussions on the implementation of non-monotone trust region steps as well as the related convergence theory.

4. Results

The proposed trust region algorithm for PDE-constrained optimization with bound constraints using reduced order modeling was applied to three compliance minimization problems. These case studies are used often to test new formulations and algorithms for topology optimization applications. Figure 1 shows the design domain, boundary conditions, and external load for each of the topology optimization test problems considered

Parameter	Value
η_1	0.2
η_2	0.8
σ_1	1
σ_2	0.5
σ_3	2
μ_0	1×10^{-2}
μ_1	1
υ	-0.1
$\hat{\epsilon}_u$	0.999

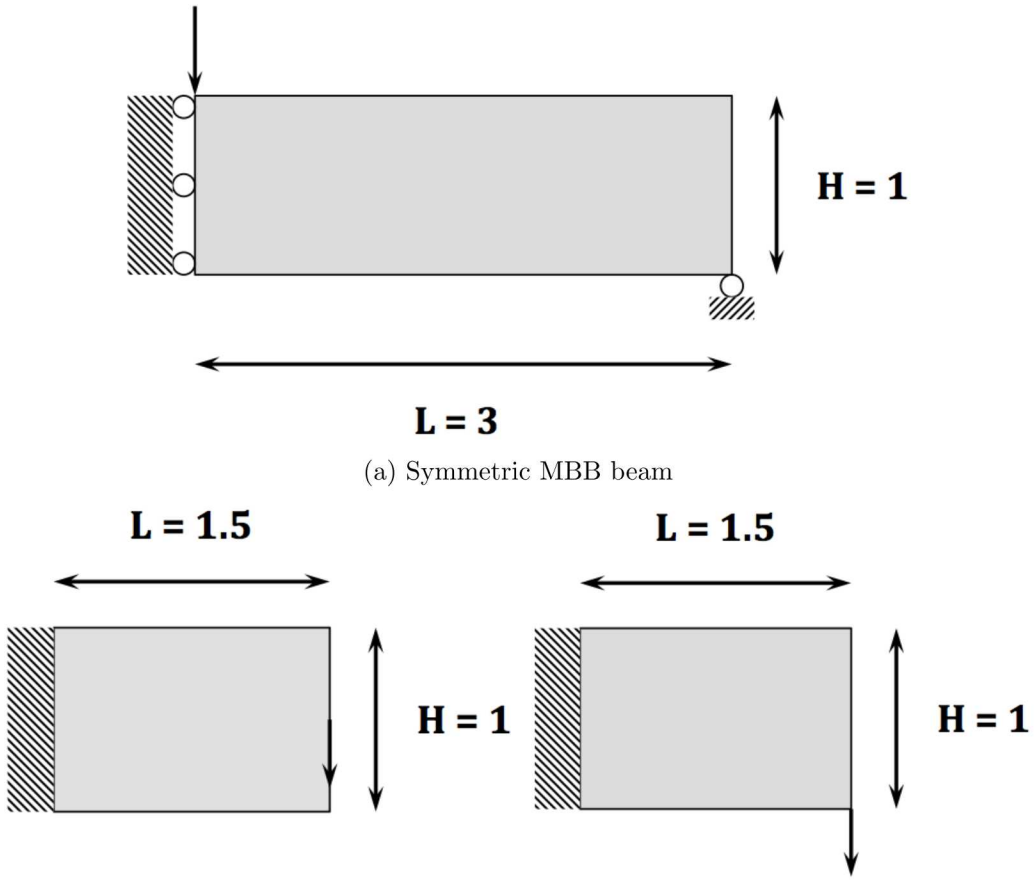
Table 2: Corresponding values for the trust region algorithm parameters.

herein. Table 1 shows the corresponding values used for the topology optimization parameters. These values will remain constant throughout, unless explicitly specified. In Table 1, $\max(L_e)$ is the maximum element length and $\mathbf{z}_0$ is the initial control. The parameter $\mathbf{1}$ denotes a vector of all ones and v_f is the volume fraction. The parameter $\text{diag}(v_f)$ denotes a diagonal matrix with values v_f along the main diagonal. The magnitude for each vertical force in Figure 1 was set to one in all the test problems. Finally, the upper and lower bounds were set to $ub = 1$ and $lb = 1 \times 10^{-2}$, repsectively.

The proposed algorithm was implemented in MATLAB scientific package [24]. Table 2 shows the corresponding values used for the trust region sub-problem parameters. The $\hat{\epsilon}_u$ parameter is the energy threshold on the POD functions. Recall that $\hat{\epsilon}_u$ is used to determine the number of eigenvectors conserved and used to generate the reduced order model for the state equation. The computational domain was discretized using piecewise linear finite elements on a regular grid. The Intrepid PDE discretization package from Trilinos [16] was used for the implementation of the linear elastostatics equality constraint. This was enable through MATLAB's MEX Library, which allows the call to external functions from the MATLAB command line. All calculations were performed on a Linux workstation with a 2.93 GHz Intel(R) Core Xeon(R) processor and 24 GB of RAM.

An advantage of compliance minimization topology optimization problems is that an explicit expression is derived for the lagrange multipliers. Thus, the reduced gradient operator can be computed without solving the adjoint system of equations. This reduces the computational cost associated with the assembly of reduced gradient operator during optimization. Furthermore, the storage requirements are reduced since only the state snapshot ensembles are needed to create the state reduced order model.

Several numerical studies were performed to test the feasibility of the proposed trust region algorithm for PDE-constrained optimization using reduced order modeling. First, the linear Hessian formulation was applied to a small-scale compliance minimization topology optimization problem. Results were compared against the results gathered by applying



(a) Symmetric MBB beam

(b) Mitchell beam

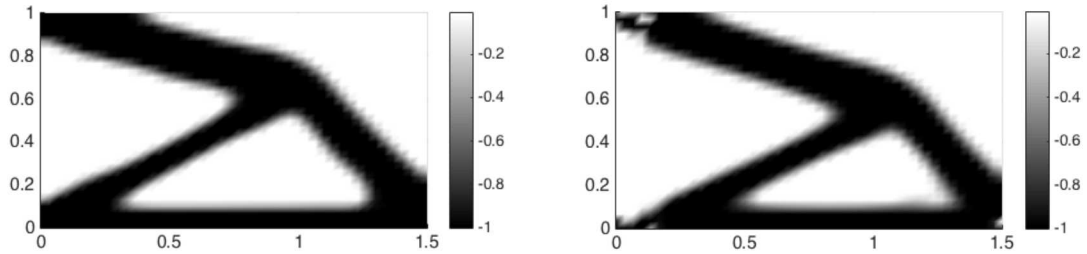
(c) Cantilever beam

Figure 1: Design domain, boundary conditions, and external load for the topology optimization case studies.

the nonlinear Hessian formulation to the same problem. The benefit of the proposed linear programming formulation is that Equations 10 and 12 are omitted during optimization. This reduces computational time since only the linear elastostaitcs FEM evaluations are needed during optimization. Second, the algorithm is applied to three topology optimization problems to investigate the robustness and feasibility of accelerating large-scale PDE constrained optimization through reduced order modeling. Finally, sensitivity of the algorithm to the initial sampling size is studied.

4.1. Linear Programming versus Nonlinear Programming Hessian Formulation

The feasibility of the proposed linear Hessian formulation for the solution of compliance minimization topology optimization problems is investigated in this section. A simple ex-



(a) Linear programming formulation

(b) Nonlinear programming formulation

Figure 2: Results for the cantilever beam with the a) linear and b) nonlinear programming Hessian formulations.

ample is presented to show that the proposed linear Hessian formulation produces optimal topologies. Thus, significant speedups are attained by avoiding the computational effort associated with the nonlinear Hessian formulation, see Section 2.1.

PDE-constrained optimization problems require nonlinear programming techniques to compute the derivative information needed to minimize the objective function. If second order derivative information is available, second order optimization algorithms can be applied to solve the problem. However, the sequence of steps presented in Section 2.1 are required to compute the application of the trial step to the nonlinear Hessian operator. Every time the application of the trial step to the Hessian operator is required, Equations 10 and 12 are solved. This fact increases computational effort since this calculation is needed multiple times during the trust region sub-problem iterations.

To circumvent having to solve Equations 10 and 12 during the trust region sub-problem iterations, a linear Hessian formulation is proposed. The advantage of this formulation is that the solutions to Equations 10 and 12 are not necessary to compute the application of the trial step to the Hessian operator. However, it is imperative to preserve the robustness and accuracy that is associated to the nonlinear Hessian formulation. Thus, the linear Hessian formulation is applied to a simple compliance minimization problem to investigate the robustness and performance of this formulation.

Figures 2a and 2b show the optimal topology obtained with the linear and nonlinear Hessian formulations. For this example, $\gamma = 1 \times 10^{-2}$ and $E_{\min} = 1 \times 10^{-2}$. Figures 2a and 2b show that the linear Hessian formulation yield better results than the nonlinear Hessian formulation. Indeed, the linear Hessian formulation required 36 iterations to converge to an optimal topology, while the nonlinear Hessian formulation did not converge to an optimal topology after 36 iterations. The convergence criterion for this example was $\|\mathbf{s}_k\| < 1 \times 10^{-3}$. However, what is more remarkable is the speedups obtained with the linear Hessian formulation. The algorithm took 32 seconds to perform 36 optimization iterations with the linear Hessian formulation. Contrary, it took the algorithm 648 seconds to perform

36 optimization iterations with the nonlinear Hessian formulation. Thus, the linear Hessian formulation enabled a 20x speedup ($\mathcal{S}$) over the nonlinear Hessian formulation.

The performance of the nonlinear formulation strategy can be further improved. In this study, the linear system of equations solve were prone to ill-conditioned due to small positive control values. This affects the performance of the algorithm since the inaccuracies induced by ill-conditioned system of equations will produce inaccurate solutions to Equations 10 and 12. This promotes inaccurate calculations of the trial step to the Hessian operator and thus impacts performance of the algorithm. Applying an effective preconditioner can control the inaccuracies induced by ill-conditioned system of equations [42]. This will produce accurate solutions to Equations 10 and 12 and thus improve the performance of the algorithm when the nonlinear Hessian formulation is apply. However, the use of a preconditioner for will also benefit the linear Hessian formulation strategy. Thus, the linear Hessian formulation will potentially outperformed the nonlinear Hessian formulation. Effective preconditioning strategies will be explored in the future to further understand the impact on both linear and nonlinear Hessian formulations.

4.2. Optimization Using Reduced Order Modeling

The proposed trust region algorithm for PDE-constrained optimization with bound constraints using reduced order modeling is demonstrated in this section. The algorithm is applied to three test problems in topology optimization; the symmetric MBB beam, Mitchell beam, and cantilever beam problems. The linear Hessian formulation is used in all the test problems. The objective is to investigate if further speedups ($\mathcal{S}$) can be achieved by applying reduced order modeling techniques for the solution of large-scale PDE-constrained optimization problems. Furthermore, the sensitivity of the algorithm to the initial sampling size is investigated.

The algorithm's stopping criteria ($\mathcal{C}$) for this numerical study are given by

1. $N_{itr} > N_{itr}^{\max}$
2. $\delta J < 1 \times 10^{-6}$
3. $\|\mathbf{s}_k\| < 1 \times 10^{-3}$
4. $\|\mathbf{w}_k^C\| < 1 \times 10^{-3}$
5. $\max(\delta \mathbf{z}_i) < 1 \times 10^{-3}$.

Here, N_{itr} is the total number of optimization iterations, $N_{itr}^{\max}$ is the maximum number of optimization iterations allow, and δJ is the change in the objective function between two subsequent iterations. The parameter $\max(\delta \mathbf{z}_i)$ is the maximum component-wise change in the control between two subsequent iterations.

Finally, it is important to emphasize that no filtering operator was used to generate the results presented in this section. Filtering operators are often applied to topology optimization problems to avoid numerical artifacts. These numerical artifacts include checkboard effect and mesh-dependent topologies [33, 34, 36]. Future work will consider the used of filtering operators and their effect on the performance of the algorithm.

Parameter	HFM	$N_{s=13}$	$N_{s=14}$	$N_{s=15}$	$N_{s=16}$
N_{itr}	100	42	100	84	43
N_{HFM}	-	23	38	41	34
N_{UPD}	-	11	24	26	18
Time (seconds)	1,577	424	1,064	848	450
$\mathcal{S}$	-	3.72	1.48	1.86	3.5
$\mathcal{C}$	$N_{itr}^{\max}$	δJ	$N_{itr}^{\max}$	$\max(\delta \mathbf{z}_i)$	$\max(\delta \mathbf{z}_i)$

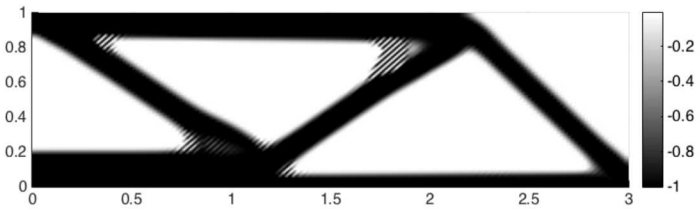
Table 3: Results for the symmetric MBB beam test problem with respect to different initial snapshot sampling size.

4.2.1. Symmetric MBB Beam

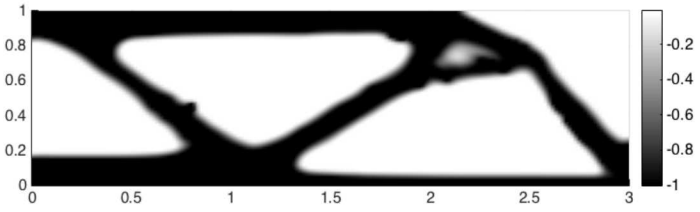
Figure 1a shows the design domain, boundary conditions, and external load used for the symmetric MBB beam test problem. The problem was discretized using a regular grid with 25,600 ($160 \times 80 \times 2$) triangles. This discretization lead to 22,082 and 11,041 state and control variables, respectively. Table 3 shows the results obtained for the symmetric MBB beam test problem. Notice that the baseline case (the high fidelity model (HFM) is used throughout the optimization process) took 1,577 seconds over 100 optimization iterations. The optimal topology for the baseline case is shown in Figure 3a. The baseline case was compared against the cases where the trust region algorithm for PDE-constrained optimization using reduced order modeling techniques was applied. Notice that for these cases the algorithm needed less computational time to produce an optimal topology. Furthermore, in most cases, the algorithm also needed less optimization iterations to produce the optimal topology. For instance, for cases $N_s = 13$, $N_s = 15$, and $N_s = 16$, significant speedups were attained, as seen from Table 3. Even for $N_s = 14$, which reached the maximum number optimization iterations allow, the computational time was lesser than the baseline case.

Figure 3 shows the optimal topologies for the symmetric MBB beam test problem. The optimal topologies obtained using the reduced order modeling techniques match the expected topology. There is a clear dependence on the initial snapshot sampling size. The performance of the algorithm as well as the quality of the optimal topology is affected by the initial sampling size. Notice that different topologies are attained as the initial snapshot sampling size is modified. However, a robust filtering operator could aid improve the quality of the solution and minimize numerical artifacts and also minimize the sensitivity to the initial sampling size. The filtering operator could also improve the predictive accuracy of the reduced order model during optimization by eliminating these potential numerical artifacts. This will enable the algorithm to reduce the number of POD function updates (N_{UPD}) and HFM evaluations (N_{HFM}) done during optimization. Reducing computational time and improving the robustness and accuracy of the algorithm during optimization.

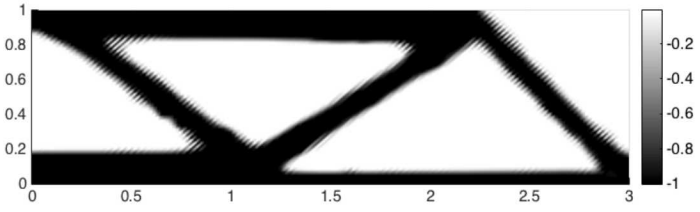
Regardless of the discernible dependence on the initial sampling size, the proposed



(a) High fidelity model

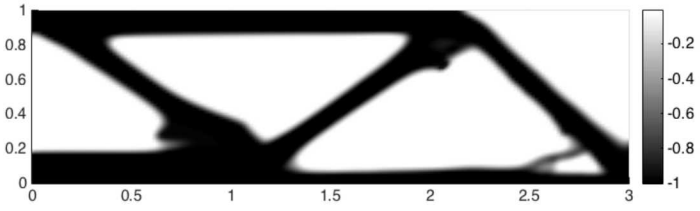


(b) $N_s = 13$



(c) $N_s = 14$

(d) $N_s = 15$



(e) $N_s = 16$

Figure 3: Results for the symmetric MBB beam test problem using reduced order modeling.

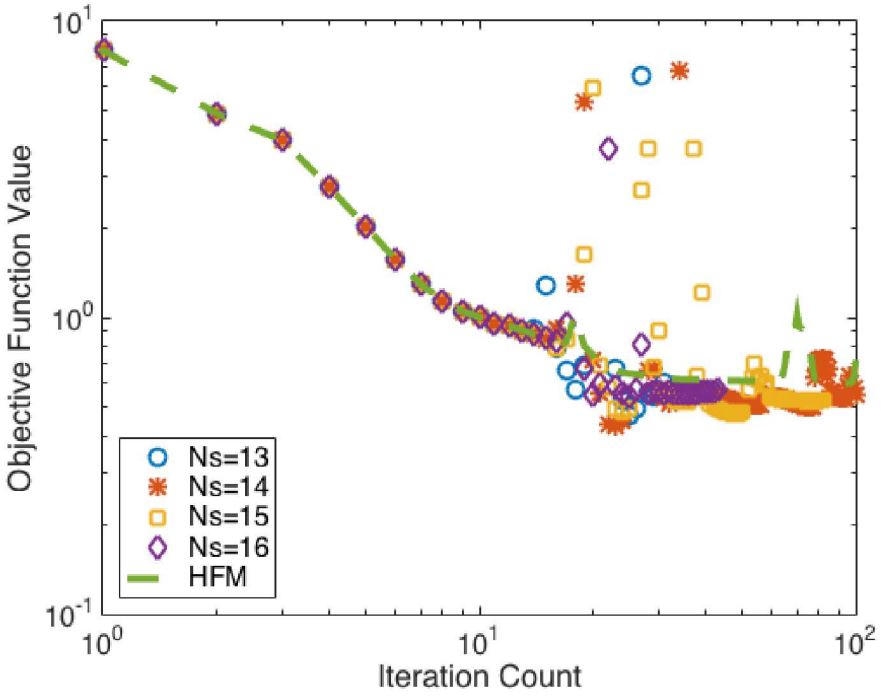


Figure 4: Objective function values for the symmetric MBB beam test problem.

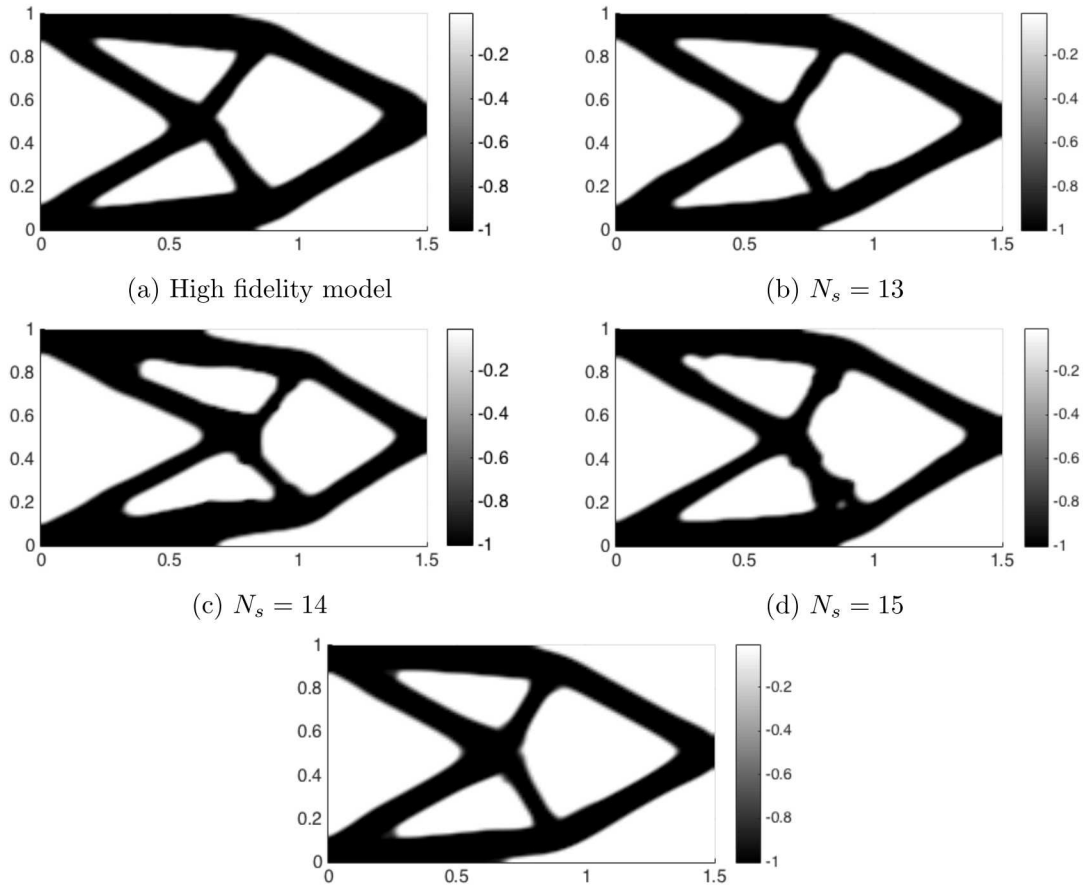
algorithm produce topologies that match the expected topology. Furthermore, significant speedups were attained by combining the linear Hessian formulation and reduced order modeling techniques to solve the compliance minimization problem. For completeness, the objective function histories for the symmetric MBB beam test problems are provided in Figure 4.

4.2.2. Mitchell Beam

Figure 1b shows the design domain, boundary conditions, and external load used for the Mitchell beam. The problem is discretized using a regular grid with 22,188 ($129 \times 86 \times 2$) triangles. This discretization lead to 22,620 and 11,310 state and control variables, respectively. Table 4 shows the results obtained for the Mitchell beam test problem. Notice that the baseline case took 2,542 seconds over 150 optimization iterations. The baseline optimal topology is shown in Figure 5a. The baseline results were compared against the results generated through PDE-constrained optimization using reduced order modeling techniques. The surrogate-based approach produce optimal results while reducing computational time. Furthermore, contrary to the symmetric MBB beam case study, the algorithm converged in less optimization iterations than the baseline case for the Mitchell beam example.

Parameter	HFM	$N_{s=13}$	$N_{s=14}$	$N_{s=15}$	$N_{s=16}$
N_{itr}	150	131	98	79	46
N_{HFM}	-	53	35	42	25
N_{UPD}	-	40	21	27	9
Time (seconds)	2,542	1,493	1,202	830	485
$\mathcal{S}$	-	1.7	2.11	3.06	5.24
$\mathcal{C}$	$N_{itr}^{\max}$	$\|\mathbf{s}_k\|$	$\max(\delta \mathbf{z}_i)$	$\max(\delta \mathbf{z}_i)$	$\|\mathbf{w}_k^C\|$

Table 4: Results for the Mitchell beam test problem with respect to different initial snapshot sampling size.



(a) High fidelity model

(b) $N_s = 13$

(c) $N_s = 14$

(d) $N_s = 15$

(e) $N_s = 16$

Figure 5: Results for the Mitchell beam test problem using reduced order modeling.

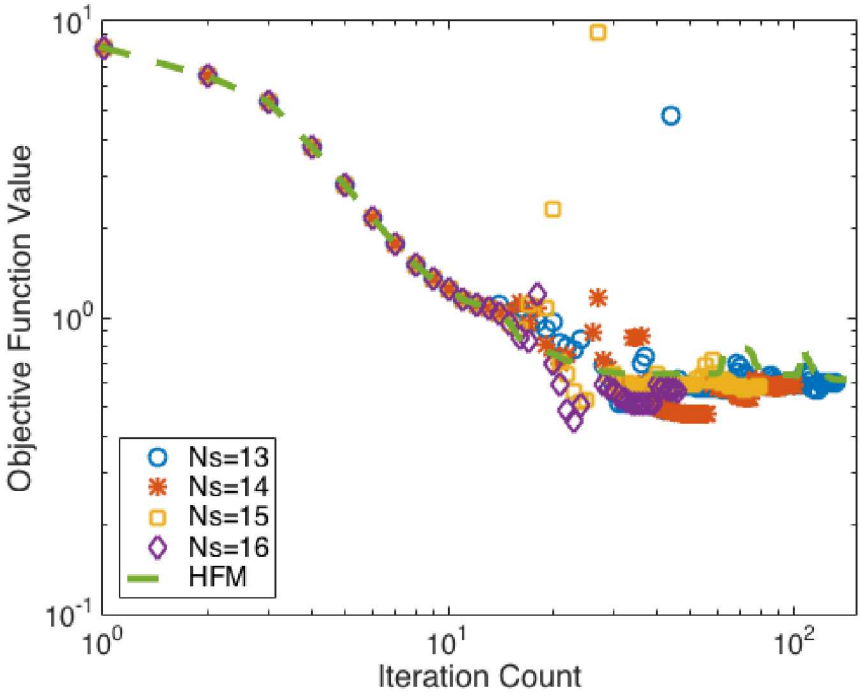


Figure 6: Objective function values for the Mitchell beam test problem.

Significant speedups, $1.7 \leq \mathcal{S} \leq 5.24$, were also attained for the Mitchell beam test problem. Indeed, the performance of the algorithm improved as the initial sampling size increased. Once again, the performance of the algorithm was dependent on the initial sampling size. However, Figure 5 shows that the optimal topologies obtained using the proposed algorithm match the expected optimal topology. Thus, the results suggest that the surrogate-based trust region algorithm can be effectively applied for the solution of large-scale compliance minimization topology optimization problems. For completeness, the objective function histories for the Mitchell beam test problems are provided in Figure 6.

4.2.3. Cantilever Beam

Figure 1c shows the design domain, boundary conditions, and external load used for the cantilever beam. The problem is discretized using a regular grid with 22,188 ($129 \times 86 \times 2$) triangles. This discretization lead to 22,620 and 11,310 state and control variables, respectively. Table 5 shows the results obtained for the cantilever beam test problem. The baseline case took 1,081 seconds over 90 optimization iterations. The optimal topology is shown in Figure 7a. Contrary to the optimal topologies shown in Figure 2, the topology

Parameter	HFM	$N_{s=13}$	$N_{s=14}$	$N_{s=15}$	$N_{s=16}$
N_{itr}	90	46	58	43	42
N_{HFM}	-	32	34	31	33
N_{UPD}	-	19	20	16	17
Time (seconds)	1,081	468	632	423	412
$\mathcal{S}$	-	2.31	1.71	2.56	2.62
$\mathcal{C}$	$\max(\delta \mathbf{z}_i)$	$\max(\delta \mathbf{z}_i)$	$\max(\delta \mathbf{z}_i)$	$\max(\delta \mathbf{z}_i)$	$\|\mathbf{w}_k^C\|$

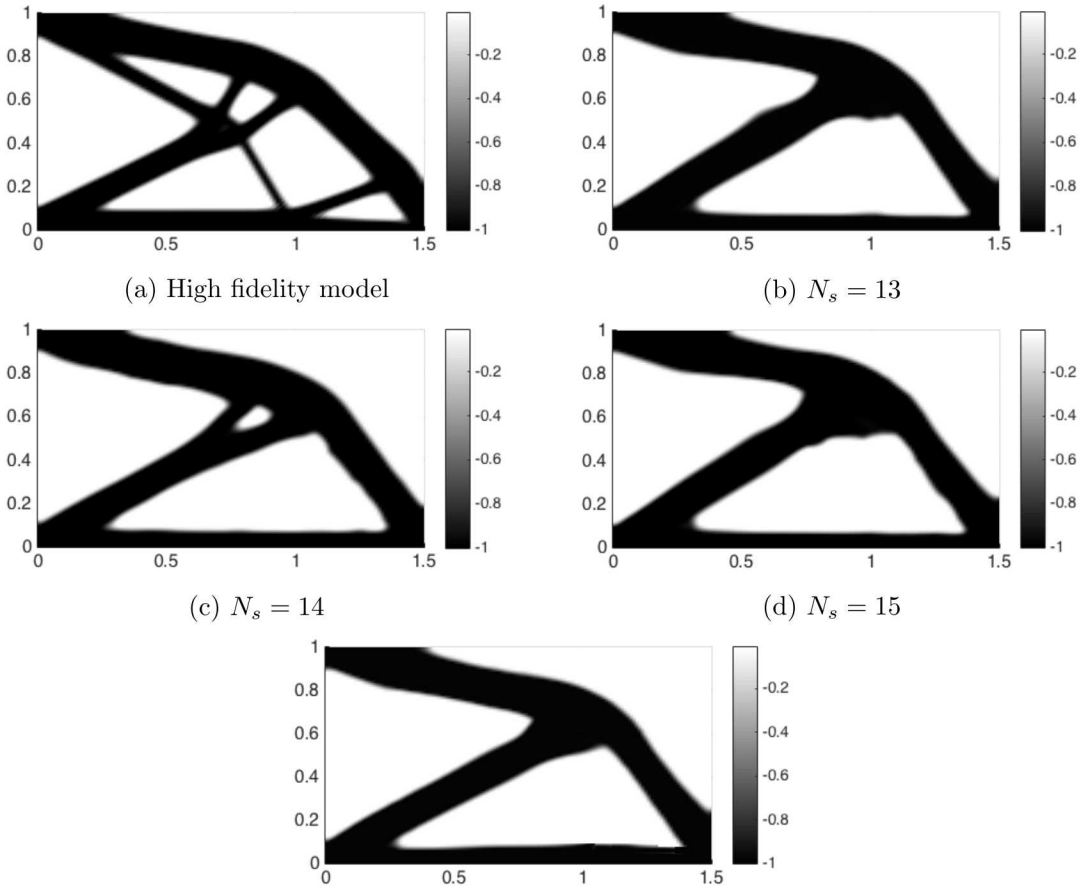
Table 5: Results for the cantilever beam test problem with respect to different initial snapshot sampling size.

in Figure 7a shows the emergence of mesh-dependent features as the element size is decreased. However, the topologies obtained with the proposed surrogate-based optimization algorithm converged to the expected optimal topology. These results suggest that the reduced order model aid filter mesh-dependent features, improving the quality of the optimal topologies produced by the algorithm.

The baseline results were compared against the results obtained with the surrogate-based trust region algorithm for PDE-constrained optimization. Once more, computational time was reduced due to fewer HFM evaluations during optimization. The proposed algorithm can be further improve by decreasing the number of POD function updates performed during optimization. Future work will focus on developing a mathematically sound strategy that enables the reduction of adaptive POD function updates during optimization. However, significant speedups, $1.71 \leq \mathcal{S} \leq 2.62$, were still attained for the cantilever beam test problem. The performance of the algorithm once more depended on the initial sampling size. Future work will also focus on improving the predictive accuracy of the orthogonal functions used to generate the reduced order model. This should reduce the sensitivity of the algorithm to the initial sampling size. Furthermore, the performance of the algorithm should improve due to the enhance accuracy of the orthogonal functions. Finally, the objective function histories for the cantilever beam test problems are plotted in Figure 8.

5. Conclusions

This paper presented a novel trust region algorithm that relies on proper orthogonal decomposition techniques to construct accurate reduced order models during optimization. The algorithm initially samples high fidelity state information to compute the POD functions that are used to generate the reduced order model. The reduced order model is then used to replace the computationally intensive high fidelity finite element evaluations during optimization. The algorithm relies on an adaptive updating scheme, which is based on the ration between the actual and predicted reduction in the objective func-



(a) High fidelity model

(b) $N_s = 13$

(c) $N_s = 14$

(d) $N_s = 15$

(e) $N_s = 16$

Figure 7: Results for the Cantilever beam test problem using reduced order modeling.

tion, to improve the predictive accuracy of the reduced order model during optimization. Results showed that the proposed algorithm reduces the computational effort required to large-scale PDE-constrained optimization problems in topology optimization.

The algorithm was applied to three compliance minimization topology optimization test problems. The algorithm consistently produce optimal topologies in fewer optimization iterations. Furthermore, the algorithm reduces the number of high fidelity model evaluations needed to compute the derivative information during optimization. This produce noticeable speedups, which is crucial for large-scale PDE-constrained optimization settings due to the reliance on computationally intensive finite element models. The algorithm is general and thus applicable to any PDE-constrained optimization setting. Future work will

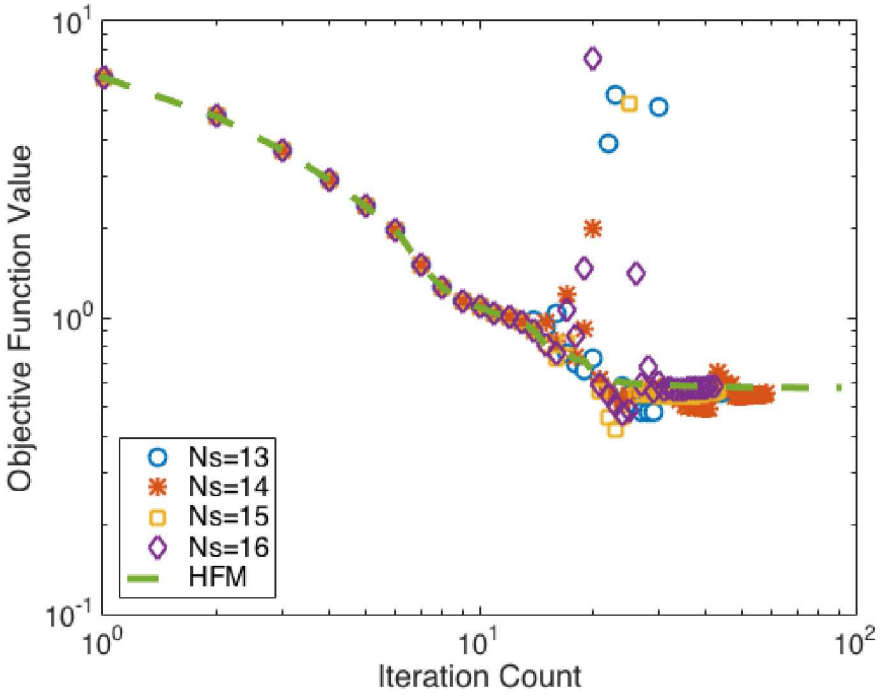


Figure 8: Objective function values for the cantilever beam test problem.

investigate if the algorithm can be applied to solve large-scale inverse problems.

This paper also presented an effective linear programming Hessian formulation for compliance minimization topology optimization problems. The trust region framework relies on a quadratic model to predict the behavior of the objective function within a suitable neighborhood. The quadratic model relies on second order derivative information to improve the performance of the trust region algorithm. If a nonlinear Hessian formulation is applied, the computational effort will increase due to additional finite element evaluations done during the trust region sub-problem iterations. Replacing the nonlinear Hessian formulation with its linear counterpart led to significant speedups, 20x. Furthermore, the performance of the algorithm and quality of the optimal solution were not negatively affected by using the linear Hessian formulation for the quadratic model.

The algorithm proposed herein can be further improved. First, more work is needed to enable optimal sampling strategies and reduce the sensitivity of the algorithm to the initial sampling size. Second, mathematically sounds techniques will be investigated to improve the predictive accuracy of the orthogonal functions used to create the reduced order model. This will allow the algorithm to rely longer on the reduced order model and avoid computationally intensive singular value decompositions or eigenvalue problems.

Finally, further speedups can be attained by applying reduced order models for both state and control variables. Therefore, expanding the current framework to allow reduction in both state and control variables will be explore in future work.

Acknowledgement

Miguel A. Aguiló thanks the DOE NNSA Advanced Simulation & Computing (ASC) program for their support and Raquel Ruiz, Miguel G. Aguiló, and Vivian Valentín for their advice and guidance.

Appendix

Equation 8 shows the second order derivative operators required to compute the application of the trial step to the nonlinear Hessian operator. These second order derivative operators are explicitly defined in this section. These operators were used to solve the nonlinear programming compliance minimization topology optimization problem during the high fidelity optimization iterations.

The necessary equality constraint derivative operators are given by

$$\langle g_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z}),\delta\mathbf{z}\rangle_{\mathbb{H}} = \langle \mathbf{u}\,\mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u},\delta\mathbf{z}\rangle_{\mathbb{H}} \tag{44}$$

$$\langle g_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z}),\delta\mathbf{u}\rangle_{\mathbb{H}} = \langle \mathbf{u}\,\mathbf{K}(\mathbf{z}),\delta\mathbf{u}\rangle_{\mathbb{H}} \tag{45}$$

$$\langle g_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*,\delta\mathbf{u}\rangle_{\mathbb{H}} = \langle \mathbf{v}\,\mathbf{K}(\mathbf{z})^*,\delta\mathbf{u}\rangle_{\mathbb{H}} \tag{46}$$

$$\langle g_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*,\delta\mathbf{u}\rangle_{\mathbb{H}} = \langle \mathbf{0},\delta\mathbf{u}\rangle_{\mathbb{H}} \tag{47}$$

$$\langle g_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*,\delta\mathbf{z}\rangle_{\mathbb{H}} = \langle \mathbf{v}(\mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u})^*,\delta\mathbf{z}\rangle_{\mathbb{H}} \tag{48}$$

$$\langle g_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*,\delta\mathbf{z}\rangle_{\mathbb{H}} = \langle \mathbf{v}(\mathbf{K}_{\mathbf{zz}}(\mathbf{z})\mathbf{u})^*,\delta\mathbf{z}\rangle_{\mathbb{H}} \tag{49}$$

$$\langle g_{\mathbf{zu}}(\mathbf{z}(\mathbf{u}),\mathbf{z})^*,\delta\mathbf{u}\rangle_{\mathbb{H}} = \langle \mathbf{v}(\mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u})^*,\delta\mathbf{u}\rangle_{\mathbb{H}}. \tag{50}$$

The second order derivative operators for the objective function are given by

$$\langle J_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*,\delta\mathbf{u}\rangle_{\mathbb{H}} = \langle \mathbf{K}(\mathbf{z}),\delta\mathbf{u}\rangle_{\mathbb{H}} \tag{51}$$

$$\langle J_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*,\delta\mathbf{z}\rangle_{\mathbb{H}} = \langle (\mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u})^*,\delta\mathbf{z}\rangle_{\mathbb{H}} \tag{52}$$

$$\langle J_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*,\delta\mathbf{z}\rangle_{\mathbb{H}} = \langle \mathbf{u}(\mathbf{K}_{\mathbf{zz}}(\mathbf{z})\mathbf{u})^* + V_{\mathbf{zz}}(\mathbf{z}) + R_{\mathbf{zz}}(\mathbf{z}),\delta\mathbf{z}\rangle_{\mathbb{H}} \tag{53}$$

$$\langle J_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*,\delta\mathbf{u}\rangle_{\mathbb{H}} = \langle (\mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u})^*,\delta\mathbf{u}\rangle_{\mathbb{H}}, \tag{54}$$

where

$$V_{\mathbf{zz}}(\mathbf{z}) = \mathbf{0} \tag{55}$$

$$R_{\mathbf{zz}}(\mathbf{z}) = \frac{\zeta}{2}((\tau-1)(\langle\nabla\mathbf{z},\nabla\mathbf{z}\rangle_{\mathbb{H}}+\nu^2)^{\tau-2})\langle\nabla\mathbf{z},\nabla\mathbf{z}\rangle_{\mathbb{H}}\hat{\mathbf{K}} + (\langle\nabla\mathbf{z},\nabla\mathbf{z}\rangle_{\mathbb{H}}+\nu^2)^{\tau-1}\hat{\mathbf{K}}. \tag{56}$$

Recall that the lagrange multipliers are explicitly given by Equation 24. Thus, the adjoint system of equations are not solved during optimization. Parameters ζ, ν, and τ were defined in Section 2.2. Here, $\hat{\mathbf{K}}$ is the scalar stiffness matrix and is given by

$$\hat{\mathbf{K}} = \int_{\Omega} \nabla\psi\nabla\psi \; \partial\Omega, \tag{57}$$

where Ω is the computational domain with boundary $\partial\Omega$. The basis function ψ was defined in Section 2.2. The first order derivative operator for the modified total variation regularization functional, Equation 20, is given by

$$R_{\mathbf{z}}(\mathbf{z}) = \frac{\zeta}{2}(\tau - 1)(\langle\nabla\mathbf{z}, \nabla\mathbf{z}\rangle_{\mathbb{H}} + \nu^2)^{\tau-1}\hat{\mathbf{K}}\,\mathbf{z}. \tag{58}$$

Applying steps 4-6 from Section 2.1 and substituting Equations 44-54 into Equations 8, 10, and 12 enables the application of the trial step to the nonlinear Hessian operator. The reader is referred to Section 2.1 to review the details concerning this calculation.

References

[1] A. Agarwal and L.T. Biegler. A trust-region framework for constrained optimization using reduced order modeling. *Optimization and Engineering*, 14(1):3–35, 2013.

[2] D. Amsallem, M.J. Zahr, and C. Farhat. Nonlinear model order reduction based on local reduced-order bases. *International Journal for Numerical Methods in Engineering*, 92(10):891–916, 2012.

[3] David Amsallem, Matthew Zahr, Youngsoo Choi, and Charbel Farhat. Design optimization using hyper-reduced-order models. *Structural and Multidisciplinary Optimization*, pages 1–22, 2014.

[4] Eyal Arian, Marco Fahl, and Ekkehard W Sachs. Trust-region proper orthogonal decomposition for flow control. Technical report, DTIC Document, 2000.

[5] D.P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

[6] Tan Bui-Thanh, Karen Willcox, and Omar Ghattas. Model reduction for large-scale systems with high-dimensional parametric input space. *SIAM Journal on Scientific Computing*, 30(6):3270–3288, 2008.

[7] James V Burke, Jorge J Moré, and Gerardo Toraldo. Convergence properties of trust region methods for linear and convex constraints. *Mathematical Programming*, 47(1-3):305–336, 1990.

[8] K. Carlberg, C. Farhat, J. Cortial, and D. Amsallem. The GNAT method for nonlinear model reduction: Effective implementation and application to computational fluid dynamics and turbulent flows. *Journal of Computational Physics*, 242(0):623 – 647, 2013.

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

[10] V.I. Danilov. *Encyclopedia of Mathematics*, chapter Implicit function (in algebraic geometry). Springer, 2001.

[11] J.E. Dennis and R.B. Schnabel. *Numerical methods for unconstrained optimization and nonlinear equations*. Prentice-Hall, Englewood Cliffs, N.J., 1983.

[12] JE Dennis Jr and HHW Mei. Two new unconstrained optimization algorithms which use function and gradient values. *Journal of Optimization Theory and Applications*, 28(4):453–482, 1979.

[13] I. Epanomeritakis, V. Akcelik, O. Ghattas, and J Bielak. A newton-cg method for large-scale three-dimensional elastic full-waveform seismic inversion. *Inverse Problems*, 24(3):1–26, 2008.

[14] Marco Fahl. *Trust-region methods for flow control based on Reduced Order Modeling*. PhD thesis, Ph. D. thesis, Trier university, 2000.

[15] D. Galbally, K. Fidkowski, K. Willcox, and O. Ghattas. Non-linear model reduction for uncertainty quantification in large-scale inverse problems. *International Journal for Numerical Methods in Engineering*, 81(12):1581–1608, 2010.

[16] 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.

[17] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, 1996.

[18] Jakob S Jensen. Topology optimization of dynamics problems with padé approximants. *International journal for numerical methods in engineering*, 72(13):1605–1630, 2007.

[19] M. Kahlbacher and S. Volkwein. POD a-posteriori error based inexact SQP method for bilinear elliptic optimal control problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46:491–511, 3 2012.

[20] Bret Kragel. Streamline diffusion pod models in optimization. *PhD, University of Trier*, 2005.

[21] O. Lass and S. Volkwein. Adaptive POD basis computation for parametrized nonlinear systems using optimal snapshot location. *Computational Optimization and Applications*, pages 1–33, 2014.

[22] Chad Lieberman, Karen Willcox, and Omar Ghattas. Parameter and state model reduction for large-scale statistical inverse problems. *SIAM Journal on Scientific Computing*, 32(5):2523–2542, 2010.

[23] Chih-Jen Lin and Jorge J Moré. Newton's method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, 9(4):1100–1127, 1999.

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

[25] Jorge J Moré. Trust regions and projected gradients. In *System Modelling and Optimization*, pages 1–13. Springer, 1988.

[26] Jorge J Moré and Gerardo Toraldo. On the solution of large quadratic programming problems with bound constraints. *SIAM Journal on Optimization*, 1(1):93–113, 1991.

[27] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 2006.

[28] Z. Ostrowski, R.A. Bialecki, and A.J. Kassab. Advances in application of proper orthogonal decomposition in inverse problems. In *Proceedings of the 5th International Conference on Inverse Problems in Engineering: Theory and Practice, Cambridge, UK*, 2005.

[29] Michael JD Powell. A new algorithm for unconstrained optimization. *Nonlinear programming*, pages 31–65, 1970.

[30] R. T. Rockafellar. Lagrange multipliers and optimality. *SIAM Review*, 35:183–238, 1993.

[31] Yousef Saad. *Iterative methods for sparse linear systems*. Siam, 2003.

[32] E.W. Sachs and S. Volkwein. POD-Galerkin approximations in PDE-constrained optimization. *GAMM-Mitteilungen*, 33(2):194–208, 2010.

[33] O. Sigmund and K. Maute. Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055, 2013.

[34] O. Sigmund and J. Petersson. Numerical instabilities in topology optimization: A survey on procedures dealing with checkerboards, mesh-dependencies and local minima. *Structural optimization*, 16(1):68–75, 1998.

[35] Ole Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5):401–424, 2007.

[36] Ole Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4-5):401–424, 2007.

[37] L. Sirovich. Turbulence and the dynamics of coherent structures. I-Coherent structures. II-Symmetries and transformations. III-Dynamics and scaling. *Quarterly of applied mathematics*, 45:561–571, 1987.

[38] L. Sirovich. Turbulence and the dynamics of coherent structures: Part i-iii. *Quart. Appl. Math.*, XLV:561–590, 1987.

[39] T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM Journal on Numerical Analysis*, 20(3):626–637, 1983.

[40] F. Tröltzsch and S. Volkwein. POD a-posteriori error estimates for linear-quadratic optimal control problems. *Computational Optimization and Applications*, 44(1):83–115, 2009.

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

[42] S. Wang, E. Sturler, and G. H. Paulino. Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering*, 69(12):2441–2468, 2007.

[43] Gil Ho Yoon. Structural topology optimization for frequency response problem using model reduction schemes. *Computer Methods in Applied Mechanics and Engineering*, 199(25):1744–1763, 2010.

[44] Yao Yue and Karl Meerbergen. Accelerating optimization of parametric linear systems by model order reduction. *SIAM Journal on Optimization*, 23(2):1344–1370, 2013.

[45] Matthew J Zahr, David Amsallem, and Charbel Farhat. Construction of parametrically-robust cfd-based reduced-order models for pde-constrained optimization. *AIAA Paper*, 2845:26–29, 2013.