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

Miguel A. Aguiló[a,*]

[a]*Morphorm LLC, One Sun Plaza, 100 Sun Avenue Northeast Suite 650, Albuquerque, NM 87109-4670, USA*

Abstract

We present a trust-region algorithm for large-scale PDE-constrained optimization with bound constraints that uses proper orthogonal decomposition (POD) to accelerate the iterative solution. The algorithm samples high-fidelity state and Lagrange-multiplier data along the optimization trajectory and automatically computes and updates the POD basis used to construct reduced-order models, which replace computationally intensive high-fidelity finite element evaluations. A trust-region acceptance criterion detects loss of predictive accuracy in the reduced model and triggers adaptive enrichment of the POD basis, providing a sound mathematical metric to control inexactness during optimization. Bound constraints on the control variables are handled natively through a projected-gradient method.

We also present a linear Hessian formulation for compliance minimization in topology optimization. The quadratic model used inside the trust-region sub-problem requires second-order derivative information; a nonlinear Hessian formulation incurs additional finite element evaluations at each inner iteration. The proposed linear formulation eliminates these extra solves. On three benchmark topology optimization problems, namely the symmetric MBB beam, the Michell beam, and the cantilever beam, it is shown to deliver substantial speedups over the high-fidelity baseline without degrading the optimal topology.

Keywords: trust region, proper orthogonal decomposition, topology optimization, Newton algorithm, projected gradient

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

[☆]An earlier version of this work was released as a Sandia National Laboratories technical report in 2015 (OSTI ID 1512909). The present manuscript revises the prose, corrects several technical errors, and updates the literature review.

[*]Corresponding author

Email address: maguilo@morphorm.com (Miguel A. Aguiló)

problems are essential in many engineering design applications. However, the iterative nature and computational requirements of optimization problems governed by PDE constraints limits the number of optimization iterations that can be performed. 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 applies model reduction techniques to accelerate optimization. The proposed algorithmic framework reduces computational time by replacing high-fidelity finite element model (FEM) evaluations with accurate reduced-order model evaluations during optimization.

Projection-based model reduction has matured into a broad and well-developed methodology [1] for compressing large-scale dynamical systems into low-dimensional surrogates that retain the essential input-output behavior of the high-fidelity model. Proper orthogonal decomposition (POD) is the most widely adopted of these techniques and is used to construct low-fidelity models of high-fidelity FEMs [2, 3, 4]. POD is based on projecting the system response onto subspaces of basis elements that contain characteristics of the expected state solution [5]. Several approaches have been proposed in the literature for surrogate-based optimization [6]. Galbally [7] employed masked projection techniques to compute approximations of nonlinear terms using a subset of interpolation terms to construct a hyper-reduced-order model. The reduced model was then applied to solve a statistical inverse problems and characterize two uncertain control variables; the reduced-order model was kept constant throughout the optimization process. Kahlbacher [8] applied POD surrogate models to solve an optimal control problem governed by a bilinear elliptic equation. A sequential quadratic programming algorithm was applied 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 [9]. The method sampled 441 distinct control (design variable) sets to generate 441 state snapshots, from which the POD basis functions used to construct the reduced-order model were computed. All 441 model evaluations were therefore performed prior to solving the optimization problem. Sachs [10] used POD *a-posteriori* error estimates based on a perturbation method to control the inexactness induced by the reduced solution. A sequential quadratic programming algorithm was applied to characterize four control variables. The algorithms used in [8, 10] ignored bound constraints.

Fahl [11, 12] 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 at every trust-region sub-problem iteration. Kragel [13] extended the work done in [11, 12] by applying a multilevel strategy that relied on coarser FEM evaluations during the trust-region sub-problem

iterations. The POD method was also applied to the coarser FEM evaluations to further expedite optimization. Fewer than 10 control variables were estimated in this study, so the TRPOD framework of [11, 12, 13] was only applied to small-scale unconstrained optimization problems.

More recent work has placed adaptive trust-region ROM methods on firmer theoretical and algorithmic footing. Zahr and Farhat [14] introduced a progressive construction of a parametric ROM, in which the reduced basis is enriched at the solution of a sequence of trust-region sub-problems and the optimization trajectory is constrained to a region of parameter space where the ROM is accurate; this strategy reduced the number of high-fidelity queries by a factor of four to five in subsonic aerodynamic shape optimization. Keil et al. [15] and Banholzer et al. [16] developed adaptive trust-region reduced-basis (TR-RB) methods for PDE-constrained parameter optimization that come with rigorous convergence guarantees, the latter incorporating projected Newton steps suited to bound-constrained problems. Yano [17] proposed a goal-oriented adaptive RB method whose dual-weighted residual estimators directly control the ROM error in the optimization output of interest. The algorithm presented here is closely related in spirit to these efforts but is tailored to large-scale topology optimization with bound constraints, using a ratio-based adaptive enrichment criterion and a projected-gradient handling of the bounds.

Until recently, model reduction had seen comparatively little use in topology optimization, with early efforts focused on dynamic problems. 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 is not applicable to general optimization problems outside the scope of dynamic problems in topology optimization. No mathematical metric was suggested for adaptively updating the Padé functions during optimization. Yoon [19] investigated the application of mode superposition, Ritz vector, and quasi-static Ritz vector methods in the context of topology optimization of dynamic problems. In [19], the orthogonal functions were updated at every optimization iteration. Thus, no adaptive updating scheme was proposed to control the computational cost associated with these orthogonal function updates. Additional computational savings could have been achieved in [19] by avoiding redundant updates. Gerzen *et al.* [20] used singular value decomposition to analyze the first derivative of the equality constraint with respect to the state and control fields. This information was used to construct a reduced subspace for the control field during optimization. However, the state field was not reduced during optimization and thus high-fidelity FEM evaluations were performed during optimization. No adaptive updating scheme was proposed to control the computational cost associated with the singular value decompositions, and the factorizations were recomputed at every optimization iteration.

Several efforts appearing shortly before, concurrently with, or after the work reported here are particularly close in spirit. Gogu [21] introduced an on-the-fly reduced-order construction for large-scale topology optimization by sampling state snapshots along the optimization trajectory and enriching a global POD basis when the surrogate became inaccurate. Xiao *et al.* [22] subsequently developed a principal-components-based on-the-fly

model reduction tailored to compliance minimization, again sampling along the optimization trajectory and updating the basis when sensitivity accuracy degraded. Choi *et al.* [23] proposed a database of linear ROMs for gradient-based optimization with linear PDE constraints, an approach particularly well-suited to compliance problems. Extensions to buckling-constrained topology optimization [24] suggest the broader applicability of on-the-fly ROM strategies in this setting. Table 1 summarizes how the algorithm presented here relates to these efforts along four axes: how the surrogate basis is enriched, whether bound constraints are treated natively, whether topology optimization is the target setting, and whether convergence is established on a rigorous trust-region footing.

Method	Enrichment	Bound constr.	Topology opt.	TR control
Fahl [11, 12]	TR-driven	no	no	yes
Zahr-Farhat [14]	TR-driven	no	no	yes
Keil et al. [15]	TR, certified	no	no	yes
Banholzer et al. [16]	TR, certified	yes	no	yes
Yano [17]	goal-oriented	no	no	—
Gogu [21]	accuracy monitor	no	yes	no
Xiao et al. [22]	sensitivity monitor	no	yes	no
Choi et al. [23]	offline database	no	yes	no
This work	TR-driven	yes	yes	yes

Table 1: Positioning of the proposed algorithm against closely related adaptive reduced-order trust-region methods and on-the-fly ROM approaches for topology optimization. "TR-driven" indicates that basis enrichment is triggered by a trust-region acceptance test rather than a heuristic monitor.

The algorithm presented here differs from these works in three respects: (i) the basis update is embedded inside a trust-region acceptance test rather than driven by a heuristic accuracy monitor, providing a direct mathematical metric for when to enrich; (ii) bound constraints are handled natively through a projected-gradient framework; and (iii) the ROM is paired with a linear Hessian formulation that avoids the additional adjoint solves required by full second-order methods.

Beyond the topology-optimization-specific thread, POD-based reduction has been applied broadly to calibration and shape optimization problems [25, 26, 27]. Although the number of control variables was limited in [25, 26, 27], the POD technique was effective at reducing the computational requirements of the optimization problem. Earlier efforts to scale model reduction to large-scale optimization settings include the surrogate-based statistical inverse problem of Lieberman *et al.* [28, 29], who combined POD with greedy methods to construct reduced-order models for both state and control variables. The reduced-order models were generated offline (before solving the optimization problem) by randomly sampling 1,000 control fields from a Gaussian prior, and both reduced models remained constant throughout optimization.

A complementary line of work circumvents the intrusiveness of projection-based ROMs altogether. Peherstorfer and Willcox [30] introduced operator inference, a non-intrusive method in which the reduced operators are learned from snapshot data by regression while preserving the polynomial structure of the underlying equations. The computational cost of nonlinear residual evaluations is a known bottleneck of projection-based ROMs. This cost has been addressed through hyperreduction methods including the discrete empirical interpolation method [31], energy-conserving sampling and weighting [32], and the empirical quadrature procedure [33]. Multi-fidelity strategies that combine surrogates of different accuracies have also matured into a substantial body of work [34]. The framework presented in this paper is most naturally instantiated with Galerkin-POD, but as emphasized in Section 3 it is not specific to that choice and is compatible with any of these alternatives.

This work expands on previous seminal work on surrogate-based optimization and trust region POD by developing a trust-region algorithm for large-scale PDE-constrained optimization with bound constraints using reduced-order modeling. The algorithm applies Galerkin-POD techniques to construct reduced-order models during optimization and minimize computational requirements. An adaptive POD-function updating scheme based on the ratio between the actual and predicted reduction in the objective function is employed to improve the performance of the reduced-order models online. This allows the algorithm to control the inexactness induced by the reduced solutions and adaptively improve the surrogates during optimization. The algorithm serves simultaneously as an optimization engine and as a sampling strategy: Offline samples are not done to generate the reduced order models. Instead, the samples created by the algorithm online (during optimization) are used to construct the reduced-order models. The algorithm also employs a projected gradient algorithm to model the bound constraints on the control variables [35]. This feature enables the algorithm to effectively solve bound constraint optimization problems, which are common in engineering design applications as well as other optimization applications. Finally, the algorithm is not specific to Galerkin-POD projection methods. Alternative reduced-order modeling techniques can be successfully applied without major modifications to the main algorithmic framework.

This paper also presents a viable Hessian formulation for structural topology optimization. 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 needs accurate second-order derivative information to properly predict the behavior of the objective function within the trust region. If a nonlinear Hessian formulation is used, the computational effort will increase due to additional FEM evaluations. Recently, Liu and Tovar [36] presented a nonlinear Hessian formulation for compliance minimization problems, which was later corrected by Evgrafov [37]. The nonlinear Hessian formulation for compliance minimization problems requires one additional FEM evaluation to compute the application of the trial step to the reduced Hessian operator. In contrast, applying the linear Hessian formulation omits these additional FEM evaluations, thereby minimizing the computational requirements. Results will show the effectiveness of the proposed linear

Hessian formulation for compliance minimization topology optimization problems. Finally, for completeness, the detailed nonlinear Hessian formulation derivation is shown in the Appendix since these details are excluded in [37, 36].

The principal contributions of this paper are the following.

1. A trust-region algorithm for large-scale PDE-constrained optimization with bound constraints that couples Galerkin-POD model reduction with a projected-gradient framework, so that bound constraints are enforced natively rather than handled through penalties or augmented Lagrangians.
2. An online adaptive POD-update scheme driven by the trust-region acceptance ratio: the basis is enriched only when the surrogate fails the ratio test, providing a direct mathematical criterion for when to query the high-fidelity model rather than a heuristic accuracy monitor.
3. A snapshot strategy that doubles as the optimization sampler: all snapshots are generated on the optimization trajectory, eliminating the offline sampling stage common to earlier surrogate-based optimization approaches.
4. A linear Hessian formulation for compliance minimization in topology optimization that omits the two finite-element solves per inner iteration required by the nonlinear formulation of [36, 37], together with a complete derivation in the appendix.
5. An empirical evaluation on three benchmark topology-optimization problems (the symmetric MBB beam, the Michell beam, and the cantilever beam) demonstrating wall-clock speedups of up to 5.4× over the high-fidelity baseline at the same iteration budget, and an additional speedup of more than 11.5× from the linear Hessian formulation.

The remainder of the paper is organized as follows. Section 2 reviews the reduced-space formulation, the topology-optimization model problem, and the POD basis construction. Section 3 develops the trust-region algorithm, the bound-constraint treatment, and the adaptive POD-update scheme. Section 4 reports numerical experiments, Section 6 concludes, and the appendix contains the second-order derivative operators required by the nonlinear Hessian formulation.

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 $g \colon U \times Z \to Y$. Let us now consider the 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 [35, 38]. 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 [39].

2.1. Reduced Space Formulation

Let us 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 $\{(\hat{u}(z), z) \mid z \in Z\} = \{(u, z) \in U \times Z \mid g(u, z) = 0\}$ [40]. This allows the redefinition of the optimization problem defined in Equation 1 as a reduced optimization problem (with the equality constraint eliminated) 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

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

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

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

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

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

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

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

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 from Equation 8 that δu and $\delta\lambda$ are required to compute the application of the trial step to the Hessian operator. These quantities can be computed by solving two independent system of equations.

Let $g(\hat{u}(z), z) = 0 \ \forall \ z \in Z$. Differentiating with respect to z and applying the chain rule gives, for all $(z, \delta z) \in Z \times Z$,

$$\frac{d}{dz}\big[g(\hat{u}(z), z)\big]\delta z = g_u(\hat{u}(z), z)\delta u + g_z(\hat{u}(z), z)\delta z = 0, \tag{9}$$

where g_u and g_z on the right-hand side denote the partial derivatives. 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

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

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

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

In summary, for reduced space PDE-constrained optimization problems, 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

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

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. In contrast, if second-order optimization algorithms are applied, steps 1 through 6 are necessary.

2.2. Structural Topology Optimization

Let $\Omega \subseteq \mathbb{R}^d$, $d \in \{1, 2, 3\}$ denote the computational domain with boundary $\partial\Omega$. Let us now define the Lebesgue space $\mathbb{H} = L^2(\Omega; \mathbb{R}^{\mathsf{n}})$ of measurable and square integrable functions endowed with inner product $\langle \phi, \psi \rangle_{\mathbb{H}} = \int_\Omega \phi\, \psi$ for $\phi, \psi \in \mathbb{H}$ and norm $\|\phi\|_{\mathbb{H}} = \langle \phi, \phi \rangle_{\mathbb{H}}^{1/2}$. Let us 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 functional $V(\mathbf{z})\colon \hat{\mathbb{Z}} \to \mathbb{R}_+^*$ quantifies the volume of the structure and is given by

$$V(\mathbf{z}) = \int_\Omega \mathbf{z}\, d\Omega. \tag{19}$$

In Equation 17, $V_0 \in \mathbb{R}_+^*$ denotes 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 [41] was used for the regularization functional

$$R(\mathbf{z}) = \frac{\zeta}{\tau} \int_\Omega (|\nabla \mathbf{z}|^2 + \nu^2)^\tau \, d\Omega, \tag{20}$$

where $|\nabla \mathbf{z}|^2 = \nabla \mathbf{z} \cdot \nabla \mathbf{z}$ is the pointwise squared gradient. This expression accommodates several regularizers as special cases: setting $\tau = 1$, $\nu = 0$ recovers Tikhonov (H^1 seminorm) regularization; $\tau = 1/2$, $\nu = 0$ recovers total variation; and $\tau = 1/2$, $0 < \nu \leq 1$ recovers a smoothed (Huber-like) total variation. 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 [42]. Equation 20 is analogous to the perimeter functional introduced by Haber *et al.* [43]. However, the surface integral is ignored from the perimeter functional proposed in [43]. Future work will investigate which functional is better suited for surrogate-based optimization.

The modified solid isotropic material with penalization (SIMP) density-based method [44] was used to penalize the stiffness of the material. The modified SIMP formulation is given by

$$E(\mathbf{z}) = E_{\min} + \mathbf{z}^p (E_0 - E_{\min}) \quad | \quad E(\mathbf{z})\colon \hat{\mathbb{Z}} \to \mathbb{R}, \tag{21}$$

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 [44] for a discussion on the advantages obtained with the modified SIMP formulation in topology optimization problems over the classical SIMP approach.

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 is given by

$$\mathcal{L}(\mathbf{u}, \mathbf{z}, \mathbf{v}) = J(\mathbf{u}(\mathbf{z}), \mathbf{z}) + \langle \mathbf{v}, \mathbf{K}\mathbf{u} - \mathbf{f} \rangle_{\mathbb{Y}^*, \mathbb{Y}}. \tag{22}$$

The first-order necessary optimality conditions for Equation 22 are given by

$$\mathcal{L}_{\mathbf{u}}(\mathbf{u}, \mathbf{z}, \mathbf{v}) = \varepsilon_{\mathbf{K}} \mathbf{K}(\mathbf{z})\mathbf{u} + \mathbf{K}(\mathbf{z})\mathbf{v} = 0 \tag{23}$$

$$\mathcal{L}_{\mathbf{z}}(\mathbf{u}, \mathbf{z}, \mathbf{v}) = \frac{\varepsilon_{\mathbf{K}}}{2} (\mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u})\mathbf{u} + (\mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u})\mathbf{v} + \varepsilon_V (V(\mathbf{z}) - V_0) V_{\mathbf{z}}(\mathbf{z}) + R_{\mathbf{z}}(\mathbf{z}) = 0, \tag{24}$$

where the subscripts $\mathbf{u}$ and $\mathbf{z}$ respectively denote derivatives with respect to the state and control. The Fréchet derivative of the regularization functional with respect to $\mathbf{z}$ is the linear functional whose action on a variation $\delta\mathbf{z}$ is

$$R_{\mathbf{z}}(\mathbf{z})\,\delta\mathbf{z} = 2\zeta \int_{\Omega} \left(|\nabla\mathbf{z}|^2 + \nu^2\right)^{\tau-1} \nabla\mathbf{z}\cdot\nabla\delta\mathbf{z}\,d\Omega. \tag{25}$$

After discretization with the basis $\{\psi^b\}$ this functional is represented by the vector $R_{\mathbf{z}}(\mathbf{z}) = 2\zeta\,\mathbf{B}_{\varphi}(\mathbf{z})\,\mathbf{z}$, where $\mathbf{B}_{\varphi}(\mathbf{z})$ is the weighted scalar stiffness matrix

$$\mathbf{B}_{\varphi}(\mathbf{z})_{ij} = \int_{\Omega} \left(|\nabla\mathbf{z}|^2 + \nu^2\right)^{\tau-1} \nabla\psi_i\cdot\nabla\psi_j\,d\Omega. \tag{26}$$

The unweighted scalar stiffness matrix $\mathbf{B}$ used as a reference object below is given by

$$\mathbf{B}_{ij} = \int_{\Omega} \nabla\psi_i\cdot\nabla\psi_j\,d\Omega. \tag{27}$$

For the compliance minimization problem defined in Equation 16, the Lagrange multipliers are given by

$$\mathbf{v} = -\varepsilon_{\mathbf{K}}\mathbf{u}. \tag{28}$$

This explicit expression is derived from Equation 23 and it is due to the self-adjoint property of $\mathbf{K}$, *i.e.* $\mathbf{K}^* = \mathbf{K}$. Substituting Equation 28 into Equation 24 yields a reduced gradient operator of the form

$$\nabla J(\mathbf{u}(\mathbf{z}),\mathbf{z}) = -\frac{\varepsilon_{\mathbf{K}}}{2}(\mathbf{K}_{\mathbf{z}}(\mathbf{z})\mathbf{u})\mathbf{u} + \varepsilon_V(V(\mathbf{z}) - V_0)V_{\mathbf{z}}(\mathbf{z}) + R_{\mathbf{z}}(\mathbf{z}). \tag{29}$$

Equation 28 enables the computation of the reduced gradient operator without solving the computationally intensive adjoint problem.

The quadratic model

$$m(\mathbf{w}_k) = \langle\nabla J(\mathbf{u}(\mathbf{z}_k),\mathbf{z}_k),\mathbf{w}_k\rangle_{\mathbb{H}} + \frac{1}{2}\langle\mathbf{w}_k,\nabla^2 J(\mathbf{u}(\mathbf{z}_k),\mathbf{z}_k)\mathbf{w}_k\rangle_{\mathbb{H}}. \tag{30}$$

is evaluated at every *k-th* trust-region sub-problem iteration to determine if the trial control $\mathbf{z}$ is accepted. The Hessian operator, or an approximation to it, is required to evaluate Equation 30. 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 increases the computational requirements since this calculation is carried out several times during a given 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 [35, 45, 38]. Quasi-Newton methods have been successfully used in many applications. However, in

this work, the second-order derivative information is approximated by a linear Hessian formulation of the form

$$\nabla^2 J(\mathbf{u}(\mathbf{z}),\mathbf{z}) \equiv \mathcal{L}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v}) = J_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z}) + g_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^*\mathbf{v}, \tag{31}$$

where Equation 31 is given by

$$\nabla^2 J(\mathbf{u}(\mathbf{z}),\mathbf{z}) = -\frac{\varepsilon_{\mathbf{K}}}{2}\mathbf{u}\,\mathbf{K}_{\mathbf{zz}}(\mathbf{z})\mathbf{u} + \varepsilon_V V_{\mathbf{z}}(\mathbf{z})V_{\mathbf{z}}(\mathbf{z}) + \varepsilon_V(V(\mathbf{z})-V_0)V_{\mathbf{zz}}(\mathbf{z}) + R_{\mathbf{zz}}(\mathbf{z}). \tag{32}$$

This linear approximation does not require the solution of Equations 10 and 12 and thus reduces the computational requirements. This approach also avoids the storage of additional snapshot ensembles and the associated symmetric eigenvalue solves, further accelerating optimization. Results will show that significant speedups are attained by using the linear Hessian formulation over its nonlinear counterpart to solve Equation 16.

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 a 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

$$\underset{\Phi_q \in \mathbb{H}}{\text{minimize}} \quad \sum_{p=1}^{N_s} \|\widehat{\mathbf{u}}_p - \sum_{q=1}^{\ell} \langle \widehat{\mathbf{u}}_p, \Phi_q\rangle_{\mathbb{H}} \Phi_q\|_{\mathbb{H}}^2 \quad \text{s.t.} \quad \langle \Phi_q, \Phi_r\rangle_{\mathbb{H}} = \delta_{qr}, \tag{33}$$

where $\ell \leq \dim(\hat{\mathbb{U}})$ and $1 \leq r \leq \ell$. The solution to Equation 33 yields the POD basis that is used to generate the reduced-order models during optimization.

The optimization problem defined in Equation 33 is equivalent to solving a symmetric eigenvalue problem or a singular value problem [46, 47]. 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

$$\Psi_{pq}\Phi_q = \sum_{p=1}^{N_s} \langle \widehat{\mathbf{u}}_p, \Phi_q\rangle \widehat{\mathbf{u}}_p = \Upsilon_q \Phi_q, \quad 1 \leq q \leq \ell, \tag{34}$$

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.

Let us 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

$$\bar{\mathbf{u}} = \sum_{q=1}^{N_\varphi} (\theta_i)_q (\widehat{\varphi}_i)_q, \tag{35}$$

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

$$(\widehat{\varphi}_i)_q = \Upsilon_q^{-1/2} \sum_{p=1}^{N_s} (\mathbf{u}_i)_p (\varphi_p)_q \quad \text{for} \quad 1 \leq q \leq N_\varphi. \tag{36}$$

This POD function ensemble is used to effectively build accurate reduced-order models and reduce the number of high-fidelity FEM evaluations performed during 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 version on which the proposed algorithm is based was presented by Lin [48]. This work adapts the trust-region algorithm presented by Lin *et al.* to PDE-constrained optimization problems where the high-fidelity PDE is replaced by a reduced-order model. Specifically, the Galerkin-POD method is used in this work to construct reduced-order models that replace high-fidelity FEM evaluations during optimization. However, the algorithm is not specific to Galerkin-POD methods. The proposed algorithm can be effectively implemented to enable the use of other reduced-order modeling techniques. The main algorithmic framework is described in Algorithm **A.1**.

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, a new trial control is given by $\mathbf{z}_{n+1} = \mathbf{z}_{n=0} + \mathbf{s}_n \mid \|\mathbf{s}_n\| \leq \Delta_n,\ \mathbf{s}_n \in \mathbb{Z}$. 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 30.

The control $\mathbf{z}_n$ and trust-region radius Δ_n are updated through standard rules in trust-region methods for unconstrained optimization [49]. Thus, given a trial step $\mathbf{w}_n$ and

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

a trial control $\mathbf{z}_{n+1}$, the ratio between the actual reduction in the objective function and the predicted reduction in the quadratic model

$$\rho_n = \frac{J(\mathbf{u}(\mathbf{z}_{n+1}), \mathbf{z}_{n+1}) - J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})}{m(\mathbf{s}_n)} \tag{37}$$

is evaluated. A trial step $\mathbf{s}_n$ with $\rho_n > 0$ yields a reduction in the objective function. The trial control $\mathbf{z}_{n+1}$ 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

$$\begin{aligned} &\Delta_{n+1} \in [\sigma_1 \min\{\|\mathbf{s}_n\|, \Delta_n\}, \sigma_2 \Delta_n] && \text{if} \quad \rho_n \leq \eta_1, \\ &\Delta_{n+1} \in [\sigma_1 \Delta_n, \sigma_3 \Delta_n] && \text{if} \quad \rho_n \in (\eta_1, \eta_2), \\ &\Delta_{n+1} \in [\Delta_n, \sigma_3 \Delta_n] && \text{if} \quad \rho_n \geq \eta_2 \end{aligned} \tag{38}$$

where $\eta_1, \eta_2 \in \mathbb{R}_+^*$ with $\eta_1 < \eta_2 < 1$ 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

$$\mathbf{w}_n^C(\alpha) = \mathbf{P}_{\hat{\mathbb{Z}}}[\mathbf{z}_{n=0} - \alpha \nabla J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})] - \mathbf{z}_{n=0}, \tag{39}$$

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

$$\mathbf{P}_{\hat{\mathbb{Z}}}(\mathbf{z}_\imath) = \begin{cases} lb_\imath & \text{if} \quad \mathbf{z}_\imath \leq lb_\imath \\ \mathbf{z}_\imath & \text{if} \quad lb_\imath < \mathbf{z}_\imath < ub_\imath \\ ub_\imath & \text{if} \quad \mathbf{z}_\imath \geq ub_\imath \end{cases} \tag{40}$$

for $1 \leq \imath \leq \mathsf{n}_{\mathbf{z}}$. Here, $\mathsf{n}_{\mathbf{z}}$ denotes the number of control variables. The scalar $\alpha \in \mathbb{R}_+^*$ is selected such that

$$m(\mathbf{w}_n^C(\alpha)) \leq \mu_0 \langle \nabla J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0}), \mathbf{w}_n^C(\alpha) \rangle_{\mathbb{H}} \mid \|\mathbf{w}_n^C(\alpha)\| \leq \mu_1 \Delta_n, \tag{41}$$

where $\mu_0 < \frac{1}{2}$ and $\mu_1 \in \mathbb{R}_+^*$. The criteria in Equation 41 can be satisfied in a finite number of iterations [50, 48, 51].

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

$$\underset{\mathbf{w}_n \in \mathbb{Z}}{\text{minimize}}\ m(\mathbf{w}_n) : \mathbf{w}_\imath = 0,\ \imath \in \mathcal{A}(\mathbf{z}),\ \|\mathbf{S}\mathbf{w}_n\| \leq \Delta_n, \tag{42}$$

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

$$\mathcal{A}(\mathbf{z}) = \{\imath \in \mathcal{I} : \mathbf{z}_\imath \in \{lb_\imath, ub_\imath\}\} \tag{43}$$

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 [52]. However, a proper implementation of the trust region framework described in Algorithm **A.2** enables the use of other Krylov solvers [53]. Different trust-region methods (dogleg [54] or double dogleg [55]) can also be applied to solve the trust-region sub-problem in Equation 42.

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 satisfies

$$\mathbf{z}_+ = \mathbf{P}_{\hat{\mathbb{Z}}}[\mathbf{z}_{n=0} + \beta \mathbf{w}_n] \in \hat{\mathbb{Z}}, \quad \|\mathbf{s}_n\| \leq \mu_1 \Delta_n, \quad \mathbf{z}_{n+1} = \mathbf{z}_{n=0} + \mathbf{s}_n \in \hat{\mathbb{Z}}, \tag{44}$$

together with the fraction-of-Cauchy-decrease condition

$$m(\mathbf{s}_n) \leq \mu_0\, m(\mathbf{w}_n^C), \tag{45}$$

following Lin and Moré [48]. The reader is referred to [56] for the underlying projected-search theory. Given a trial step $\mathbf{s}_n$ that satisfies Equations 44 and 45, the ratio between the actual reduction in the objective function and the predicted reduction in the quadratic model is computed to determine if the trial control $\mathbf{z}_{n+1}$ is accepted. If $\rho_n > \eta_1$, $\mathbf{z}_{k+1} = \mathbf{z}_{n+1}$; else, $\mathbf{z}_{n+1} = \mathbf{z}_{n=0}$. The trust-region radius is updated according to the rules presented in Equation 38. Algorithm **A.2** describes the trust-region sub-problem framework.

Data: **If** $M \leq N$ **then** reduced-order model is inactive **else** reduced order 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.

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

Step 2: Compute descent direction.

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

Step 3: Compute trial step.

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

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

3.2. Adaptive POD functions update

The trust-region algorithm for PDE-constrained optimization with bound constraints described in Section 3.1 applies model reduction techniques to enable faster FEM evaluations during optimization (**online**). The algorithm samples high-fidelity state and Lagrange multipliers snapshots and applies a Galerkin-POD projection method to generate the respective reduced-order models. This approach avoids ineffective sampling of state and Lagrange multipliers snapshots before solving the optimization problem (**offline**). The algorithm will rely on the trust-region framework to effectively generate state and Lagrange multipliers snapshots **online**. These snapshot ensembles are leveraged by the algorithm to construct effective reduced-order models **online**. The conjecture is that the optimization algorithm will generate optimal data sets **online** due to the explicit enforcement of the desired problem requirements (constraints) through the optimization formulation.

At each successful *k-th* iteration of the trust-region algorithm, the ratio between the actual reduction in the objective function and the predicted reduction in the quadratic model is computed. This ratio is leveraged by the algorithm **online** to discern how well the reduced-order model is performing. Based on this ratio, the algorithm applies an adaptive POD function updating scheme to enhance the predictive accuracy of the reduced-order model **online**.

Two counters drive the activation logic of the reduced-order model and are referenced throughout Algorithms **A.1**–**A.3**: M is the number of high-fidelity state/Lagrange-multiplier snapshots collected so far during the optimization run, and N is the user-specified initial training-set size (equal to N_s in Tables 5–7). While $M < N$ the reduced-order model is inactive and the algorithm relies exclusively on high-fidelity FEM evaluations; once $M = N$ the initial POD basis is constructed and the reduced-order model becomes active for subsequent iterations.

Given a number of snapshots N_s, the algorithm initially relies on high-fidelity FEM evaluations to compute state and Lagrange multipliers snapshots while progress is made towards the optimal solution. Once the initial state and Lagrange multipliers ensembles are gathered, the algorithm creates a reduced-order model for each quantity of interest (state and Lagrange multipliers). This approach avoids computationally intensive FEM evaluations **offline**. Thus, high-fidelity state and Lagrange multipliers snapshots are only computed **online**. This leads to significant speedups since high-fidelity FEM evaluations are only performed as needed during optimization.

The proposed algorithm relies on ρ_k to detect loss of predictive accuracy in the reduced order models **online**. If $\rho_k \leq \upsilon$, the reduced-order models are not accurate for optimization and a new 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 stored in their respective snapshot 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 updated snapshot ensembles $\widehat{\mathbf{u}}$ and $\widehat{\mathbf{\Lambda}}$ are used to restore the corresponding sets of eigenvalues and eigenvectors by solving

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

and

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

The parameter $\upsilon \in \mathbb{R}$ denotes 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 enhance the accuracy of the reduced-order models **online**.

The number of state eigenvectors (N_φ) and Lagrange multipliers eigenvectors (N^{λ}_{φ}) conserved to generate the Galerkin-POD basis functions is based on the amount of energy captured by each basis function. The energy measures are computed from the eigenvalue ensembles $\{\Upsilon_q\}_{q=1}^{\ell}$ and $\{\Upsilon^{\lambda}_q\}_{q=1}^{\ell_\lambda}$ and are given by

$$\epsilon_u = \frac{\sum_{q=1}^{N_\varphi}\Upsilon_q}{\sum_{q=1}^{\ell}\Upsilon_q}, \qquad \epsilon_\lambda = \frac{\sum_{q=1}^{N^{\lambda}_{\varphi}}\Upsilon^{\lambda}_q}{\sum_{q=1}^{\ell_\lambda}\Upsilon^{\lambda}_q}. \tag{46}$$

Here, ϵ_u and ϵ_λ are the energy measures for the state and Lagrange multipliers ensembles, respectively. It is often observed that a finite number of eigenvalues and eigenvectors are necessary to compute the Galerkin-POD functions used to construct the 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 $\widehat{\mathbb{W}}_\lambda = \text{span}\{(\widehat{\varphi}^{\lambda}_i)_q\}_{q=1}^{N^{\lambda}_{\varphi}} \subset \mathbb{H}$, the Galerkin-POD function ensembles $\bar{\mathbf{u}}$ and $\mathbf{\Lambda}$ are computed by applying Equations 35 and 36. As emphasized earlier, the algorithm is not specific to Galerkin-POD projection methods. Other projection methods can be used during optimization to generate the reduced-order models. The key is to design and implement a general algorithmic framework that enables the use of different projection methods for model reduction. Algorithm **A.3** describes the adaptive Galerkin-POD updating scheme used to control the inexactness induced by the reduced solution **online**.

3.3. Algorithmic Extensions

The trust-region sub-problem algorithm proposed in Section 3.1 can be modified to accept non-monotone steps. For highly nonlinear programming problems, this strategy enables the algorithm to avoid becoming trapped in local minima during optimization. The idea of non-monotone trust-region algorithms is to modify the acceptance criteria, **Step 4**

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 46
        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 46
            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

of Algorithm **A.2**, 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

$$J_{\min} = \min_{\hat{j} \in \{0,n\}} J(\mathbf{u}(\mathbf{z}_{\hat{j}}), \mathbf{z}_{\hat{j}}). \tag{47}$$

Let us also define $\hat{j}$ successful iterations since $J_{\min}$ was first computed. The reference objective function value J_r is updated if $\hat{j}$ 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, let us define parameters σ_c and σ_r. The sum of the predicted model decrease over all previous successful iterations is denoted by σ_c. The sum of the predicted model decrease over all previous successful iterations since the reference iteration is denoted by σ_r.

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 $J_{\min} = J_r = J_c = J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})$ 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 described in Algorithm **A.4**.

A non-monotone trust-region algorithm proved more effective than the monotone counterpart for solving Equation 16. The non-monotone strategy required fewer iterations and objective function evaluations than the monotone strategy. This reduced the number of high fidelity FEM evaluations and thus accelerated optimization. Algorithm **A.2** can be easily modified to enable a non-monotone trust-region step by limiting the number of trust region sub-problem iterations. The trial control is then accepted without enforcing descent in the objective function. The interested reader is referred to [49] for detailed discussions on the implementation of non-monotone trust-region steps as well as 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 often used to test new formulations and algorithms for topology optimization. Figure 1 shows the design domain, boundary conditions, and external load for each of the topology optimization test problems considered herein. Table 2 shows the corresponding values used for the topology optimization parameters. These values will remain constant throughout, unless explicitly specified. In Table 2, $\max(L_e)$ 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 denotes the volume fraction. The parameter $\text{diag}(v_f)$ denotes a diagonal

Step 4: Acceptance of the trial control.

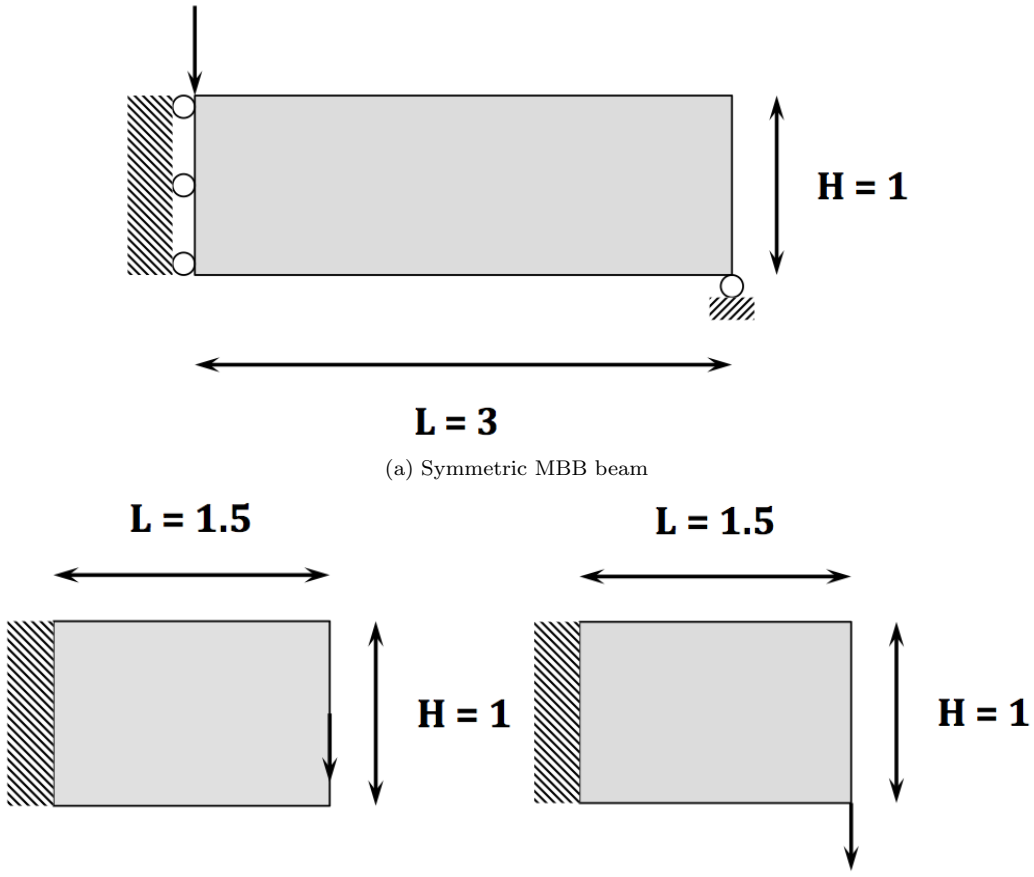
1. Compute $J(\mathbf{u}(\mathbf{z}_{n+1}), \mathbf{z}_{n+1})$ and

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

$$\rho_n^2 = \frac{J(\mathbf{u}(\mathbf{z}_{n+1}), \mathbf{z}_{n+1}) - J(\mathbf{u}(\mathbf{z}_{n=0}), \mathbf{z}_{n=0})}{m(\mathbf{s}_n)},$$

   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 $\mathbf{z}_{k+1} = \mathbf{z}_{n+1}$ and update $\sigma_r \leftarrow \sigma_r + m(\mathbf{s}_n)$ and $\sigma_c \leftarrow \sigma_c + m(\mathbf{s}_n)$
   end
2. Update best objective function value
   if $J(\mathbf{u}(\mathbf{z}_{n+1}), \mathbf{z}_{n+1}) < J_{\min}$
      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 $J(\mathbf{u}(\mathbf{z}_{n+1}), \mathbf{z}_{n+1}) > J_c$
      Set $J_c = J(\mathbf{u}(\mathbf{z}_{n+1}), \mathbf{z}_{n+1})$ 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



(a) Symmetric MBB beam

(b) Michell beam

(c) Cantilever beam

Figure 1: Design domains, boundary conditions, and applied loads for the three benchmark problems. (a) Symmetric MBB beam with pinned left edge and roller support at the bottom-right corner ($L = 3$, $H = 1$). (b) Michell beam ($L = 1.5$, $H = 1$) and (c) cantilever beam ($L = 1.5$, $H = 1$), both with clamped left edge. Hatching denotes clamped boundaries; circles denote pinned/roller supports; arrows are unit point loads ($F = 1$).

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 $lb = 1 \times 10^{-2}$, respectively.

The proposed algorithm was implemented in MATLAB scientific package [57]. Table 3 shows the corresponding values used for the trust-region sub-problem parameters. The $\hat{\epsilon}_u$ parameter is the energy threshold on the state POD functions. Recall that $\hat{\epsilon}_u$ is used to determine the number of eigenvectors conserved for the generation of the reduced-order model. The computational domain was discretized using piecewise linear finite elements on

Parameter	Value
E_0	$1\,\mathrm{Pa}$
$E_{\min}$	$1 \times 10^{-6}\,\mathrm{Pa}$
p	3
ν	1×10^{-8}
τ	0.5
ζ	$0.5\ \max(L_e)$
v_f	0.3
$\mathbf{z}_0$	$\mathrm{diag}(v_f) \cdot \mathbf{1}$
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 2: Corresponding values for the topology optimization parameters.

a regular grid. The Intrepid PDE discretization package from Trilinos [58] was used for the implementation of the linear elastostatics equality constraint. This was enabled through MATLAB's MEX Library, which allows external functions to be called 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 problems is that the reduced gradient operator can be computed without solving the adjoint system of equations. This reduces the computational requirements to assemble the reduced gradient operator during optimization. Storage requirements are likewise minimized: only the state snapshot ensembles are needed to generate the reduced-order model for the state equations.

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 problem. The results were compared against the results gathered by solving Equation 16 using the nonlinear Hessian formulation. Second, the algorithm was applied to three compliance minimization problems; the symmetric MBB beam, Michell beam, and cantilever beam, to investigate the feasibility of accelerating large-scale PDE constrained optimization through reduced-order modeling. Finally, the sensitivity of the algorithm to the initial sampling size was investigated.

Table 4 previews the main outcome of the numerical study by reporting, for each test problem, the best wall-clock speedup obtained over the high-fidelity baseline together with the corresponding initial training-set size and number of high-fidelity finite-element evaluations performed during optimization. The detailed breakdown by training-set size appears in Tables 5–7.

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

Table 3: Corresponding values for the trust-region algorithm parameters.

Problem	DoFs ($\mathbf{u}/\mathbf{z}$)	HFM (s)	Best N_s	ROM (s)	Best $\mathcal{S}$
Symmetric MBB beam	22,082 / 11,041	2,381	13	440	5.41×
Michell beam	22,620 / 11,310	2,542	16	485	5.24×
Cantilever beam	22,620 / 11,310	1,081	16	412	2.62×

Table 4: Headline results across the three benchmark problems. The HFM-baseline times for the MBB and Michell beams correspond to the iteration cap $N_{itr}^{\max} = 150$ (none of the five convergence criteria of Section 4.2 were met within the cap), so the reported speedups are lower bounds on the true time-to-convergence speedup; the cantilever baseline did converge within the cap.

4.1. Linear versus Nonlinear Hessian Formulation

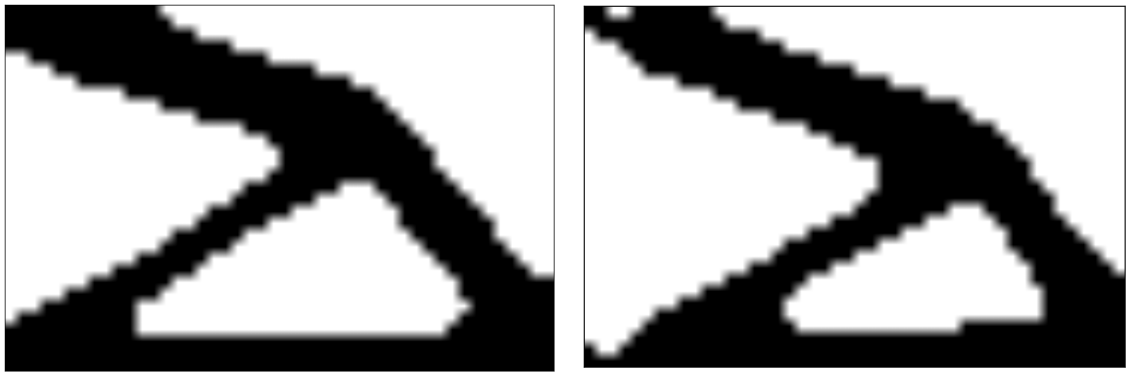
The feasibility of accelerating compliance minimization problems by applying a linear Hessian formulation is investigated in this section. A small-scale cantilever beam example is presented to demonstrate the effectiveness of the proposed linear Hessian formulation. Results will show that significant speedups are attained by omitting the additional FEM evaluations that are performed to compute the application of the trial step to the nonlinear Hessian operator, see Section 2.1. Furthermore, the proposed formulation did not hinder the accuracy of the optimal solution.

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 this computation is required, Equations 10 and 12 are solved. These additional FEM evaluations are performed multiple times during the trust-region sub-problem iterations, increasing the computational demands of large-scale problems.

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 accuracy that is associated with the nonlinear Hessian formulation.

Figures 2a and 2b show the topologies obtained with the linear and nonlinear Hessian formulations, respectively. The problem was discretized using a regular grid with 2,700 ($45 \times 30 \times 2$) triangles. Here, the first number denotes the number of intervals in the x-direction; the second, the number of intervals in the y-direction; and the third, the number of triangles per square. This discretization led to 2,852 and 1,426 state and control variables, respectively. The convergence criterion for this example was $\delta J < 1 \times 10^{-6}$, where δJ is the change in the objective function between two subsequent iterations.

The linear Hessian formulation reached this criterion in 51 iterations, whereas the nonlinear Hessian formulation had not yet converged after the same 51 iterations. Wall-clock cost differs even more sharply: the algorithm took 67 seconds for 51 optimization iterations with the linear Hessian formulation, compared with 773 seconds with the nonlinear formulation – a per-iteration speedup of $\mathcal{S} \approx 11.5\times$. Because the nonlinear run was truncated before reaching the stopping criterion, $11.5\times$ is a lower bound on the time-to-convergence speedup; the true factor in favor of the linear formulation is larger.



(a) Linear Hessian formulation

(b) Nonlinear Hessian formulation

Figure 2: Optimal density fields for the cantilever beam on a 45×30 element mesh (2,700 triangles) with the (a) linear and (b) nonlinear Hessian formulations. Black = solid material ($\rho = 1$), white = void ($\rho = 0$).

The performance of the nonlinear formulation strategy can be further improved. In this study, the linear systems of equations were prone to ill-conditioning due to small positive control values. This affects the performance of the algorithm since the inaccuracies induced by ill-conditioned systems, *e.g.* Equations 10 and 12, will produce inaccurate solutions. Applying an effective preconditioner can help control the inaccuracies induced by these

ill-conditioned systems [59]. This will result in accurate solutions to Equations 10 and 12, improving the performance of the algorithm. However, the linear Hessian formulation strategy will also benefit from the use of a preconditioner. Therefore, the linear Hessian formulation strategy is still expected to outperform its nonlinear counterpart. Effective preconditioning strategies will be explored in the future to further understand the impact on both the linear and nonlinear Hessian formulations. Finally, future research will investigate if the linear Hessian formulation strategy outperforms quasi-Newton approximation methods.

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, Michell 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.

The 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. $\|\mathbf{w}_k^C\| < 1 \times 10^{-4}$
5. $\max(\delta \mathbf{z}_\iota) < 1 \times 10^{-3}$.

Here, N_{itr} is the total number of optimization iterations and $N_{itr}^{\max}$ is the maximum number of optimization iterations. The parameter $\max(\delta \mathbf{z}_\iota)$ is the maximum component-wise change in the control between two subsequent iterations. Finally, a kernel-based filtering operator [60, 61] with a filter radius of $r = 2$ element edge lengths was used to generate the results presented in this section. These types of filtering operators are often used in topology optimization problems to reduce numerical artifacts. These numerical artifacts include the checkerboard effect and mesh-dependent topologies [62, 63, 64]. The filter imposes a minimum length scale on the control field, which suppresses these artifacts without altering the trust-region update or the reduced-order modeling strategy described above.

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 led to 22,082 and 11,041 state and control variables, respectively. Table 5 shows the results obtained for the symmetric MBB beam test problem. The baseline case is the high-fidelity model (HFM) used throughout the optimization process. This baseline ran for 2,381 seconds and exhausted the iteration budget at $N_{itr}^{\max} = 150$ without satisfying any of the convergence criteria listed in Section 4.2;

Parameter	HFM	$N_{s=13}$	$N_{s=14}$	$N_{s=15}$	$N_{s=16}$
N_{itr}	150	46	150	91	47
N_{HFM}	—	23	51	45	32
N_{UPD}	—	10	37	30	16
Time (s)	2,381	440	1,723	864	452
$\mathcal{S}$	—	5.41	1.38	2.76	5.26
$\mathcal{C}$	$N_{itr}^{\max}$	$\max(\delta \mathbf{z}_i)$	$N_{itr}^{\max}$	$\max(\delta \mathbf{z}_i)$	$\max(\delta \mathbf{z}_i)$

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

the reported wall-clock time therefore reflects a fixed iteration budget rather than time-to-convergence. The topology produced by the baseline at the iteration cap is shown in Figure 3(a). The baseline case was compared against the cases where the reduced-order model was used during optimization, all of which did satisfy a convergence criterion within the budget. Table 5 shows that noticeable speedups were attained by using reduced-order models during optimization. In most cases the algorithm also required fewer optimization iterations to produce an optimal topology. For instance, for cases $N_s = 13$, $N_s = 15$, and $N_s = 16$, significant speedups were attained, as seen in Table 5. Even for $N_s = 14$, which required the same amount of iterations as the baseline case, the computational time was less than the baseline case.

Figure 3 shows the optimal topologies for the symmetric MBB beam test problem. The optimal topologies obtained using the surrogate-based approach match the expected topology. The kernel-based filter applied throughout this study imposes a minimum length scale on the control field, so the recovered topologies are free of the checkerboard pattern and the mesh-dependent artifacts that can otherwise emerge at this resolution. The algorithm nonetheless shows a dependence on the initial sampling size: the predictive accuracy of the reduced-order model, and with it the number of POD function updates (N_{UPD}) and the number of HFM evaluations (N_{HFM}) performed during optimization, varies with N_s. This dependence reflects the reliability of the POD basis rather than any numerical artifact in the control field, and is addressed by the POD-enrichment and sampling strategies discussed for the Michell beam (Section 4.2.2) and in the conclusions (Section 6).

Regardless of the discernible dependence on the initial sampling size, the proposed algorithm produced topologies that matched the expected topology, with significant speedups 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 problem are provided in Figure 4.

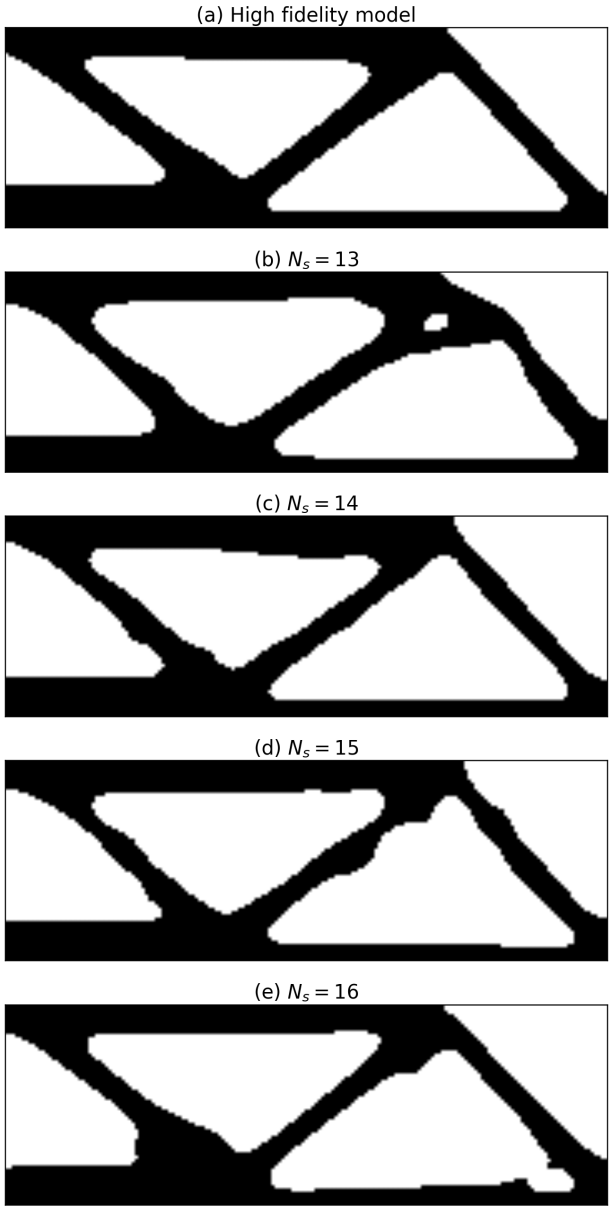


Figure 3: Optimal density fields for the symmetric MBB beam on a 180×60 element mesh. (a) High-fidelity baseline. (b)–(e) Surrogate-based trust-region optimization with $N_s \in \{13, 14, 15, 16\}$ initial POD snapshots. Black = solid material ($\rho = 1$), white = void ($\rho = 0$).

4.2.2. Michell Beam

Figure 1b shows the design domain, boundary conditions, and external load used for the Michell beam test problem. The problem is discretized using a regular grid with 22,188

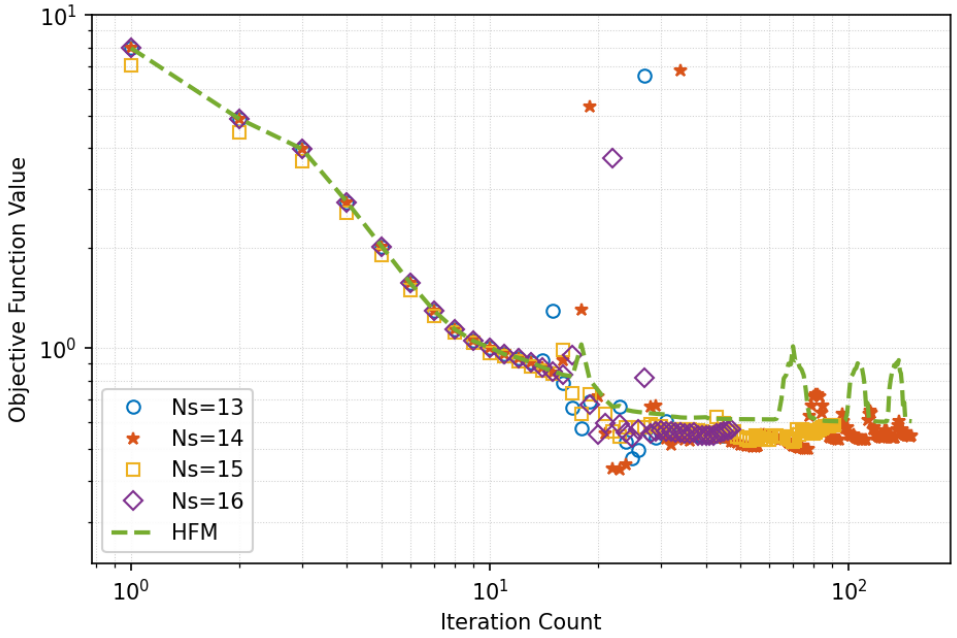


Figure 4: Objective-function convergence for the symmetric MBB beam. Dashed green: high-fidelity baseline. Markers: surrogate-based runs at $N_s \in \{13, 14, 15, 16\}$ initial POD snapshots.

$(129 \times 86 \times 2)$ triangles. This discretization led to 22,620 and 11,310 state and control variables, respectively. Table 6 shows the results obtained for the Michell beam test problem. The baseline case ran for 2,542 seconds and likewise exhausted the iteration budget at $N_{itr}^{\max} = 150$ without satisfying a convergence criterion (the same caveat as in Section 4.2.1 applies to the reported speedups here). The corresponding topology at the iteration cap is shown in Figure 5(a). The baseline results were compared against the results generated by applying reduced-order modeling techniques during optimization. The surrogate-based approach produced optimal results while reducing computational time. In contrast to the symmetric MBB beam case study, all surrogate-based cases also converged in fewer optimization iterations than the baseline.

Significant speedups, $1.7 \leq \mathcal{S} \leq 5.24$, were attained for the Michell beam test problem. The performance of the algorithm improved as the initial sampling size increased, exhibiting the same sensitivity to that size observed in the MBB case. Figure 5 shows the optimal topologies obtained using the surrogate-based optimization. These results matched the expected optimal topology. However, the quality of the solution depended on the initial sampling size. This could be improved by enhancing the predictive reliability of the POD functions through enrichment methods. Future work will explore techniques to enrich POD functions and improve their effectiveness during optimization. Nevertheless, the

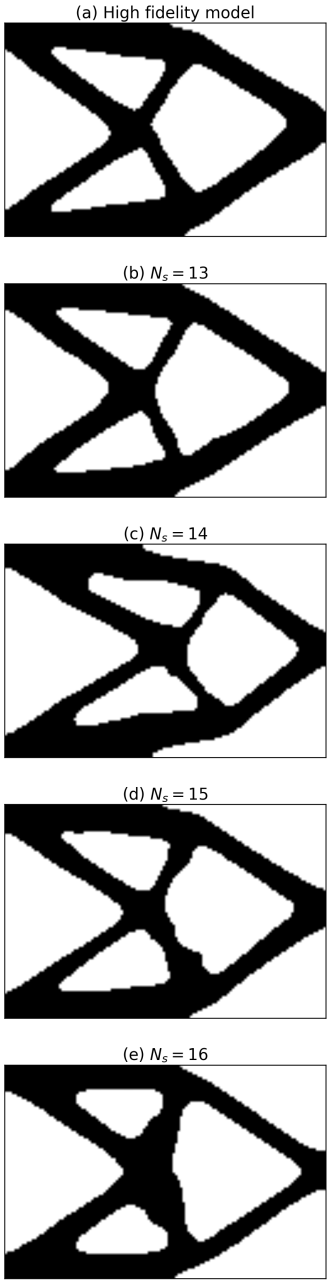


Figure 5: Optimal density fields for the Michell beam on a 129×86 element mesh. (a) High-fidelity baseline. (b)–(e) Surrogate-based optimization with N_s initial POD snapshots.

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 (s)	2,542	1,493	1,202	830	485
$\mathcal{S}$	—	1.70	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 6: Results for the Michell beam test problem with respect to different initial snapshot sampling size.

results suggest that the surrogate-based trust-region algorithm can be used to accelerate large-scale topology optimization problems and produce accurate results. For completeness, the objective function histories for the Michell beam test problem are provided in Figure 6.

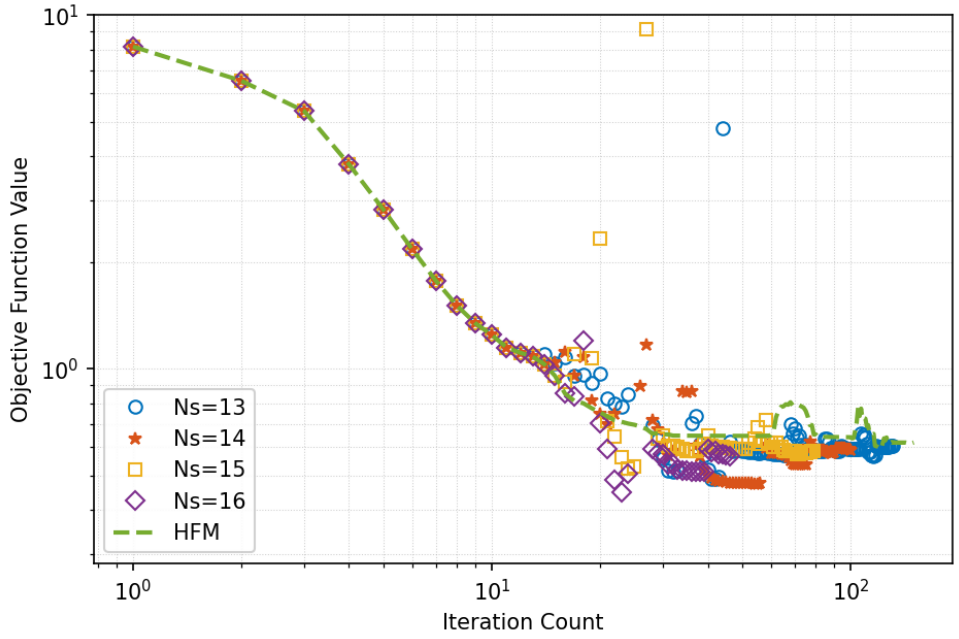


Figure 6: Objective-function convergence for the Michell beam. Dashed green: high-fidelity baseline. Markers: surrogate-based runs at $N_s \in \{13, 14, 15, 16\}$ initial POD snapshots.

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 (s)	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 7: Results for the cantilever beam test problem with respect to different initial snapshot sampling size.

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 led to 22,620 and 11,310 state and control variables, respectively. Table 7 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 7(a). The kernel-based filter applied throughout this study imposes a minimum length scale on the control field. As a result, the optimal topology is free of the checkerboard pattern and the mesh-dependent features that can otherwise emerge at this resolution as the element size is decreased. The topologies obtained with the proposed surrogate-based optimization algorithm exhibit the same length-scale control and converge to the same expected optimal topology as the high-fidelity baseline.

The baseline results were compared against the results obtained with the surrogate-based trust-region algorithm. As before, the performance of the algorithm depended on the initial sampling size. However, Table 7 shows that significant speedups, $1.71 \leq \mathcal{S} \leq 2.62$, were still attained for the cantilever beam test problem. Thus, the proposed surrogate-based trust-region algorithm shows its ability to produce optimal results in less time. The algorithm can be further accelerated by decreasing the number of POD function updates done during optimization. Future work will focus on developing a sound strategy that reduces the number of adaptive POD function updates during optimization. For completeness, the objective function histories for the cantilever beam test problem are plotted in Figure 8.

5. Limitations

The numerical evidence supports the proposed algorithm on the problem class for which it was designed, but several limitations should temper its broader interpretation. First, the experiments are confined to compliance minimization on three two-dimensional benchmarks under plane-strain linear elasticity; the behavior of the algorithm on three-dimensional problems, on multi-physics couplings, or on objectives whose adjoint cost dominates the

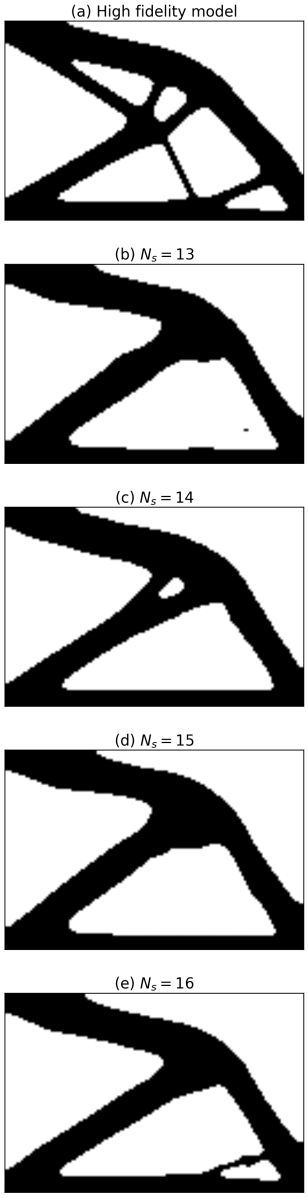


Figure 7: Optimal density fields for the cantilever beam on a 129×86 element mesh. (a) High-fidelity baseline. (b)–(e) Surrogate-based optimization with N_s initial POD snapshots.

primal cost is not characterized here. Second, only bound constraints on the control are treated natively; general nonlinear inequality constraints, volume-bound or stress-bound formulations beyond the linear volume equality used in this work, are outside the present

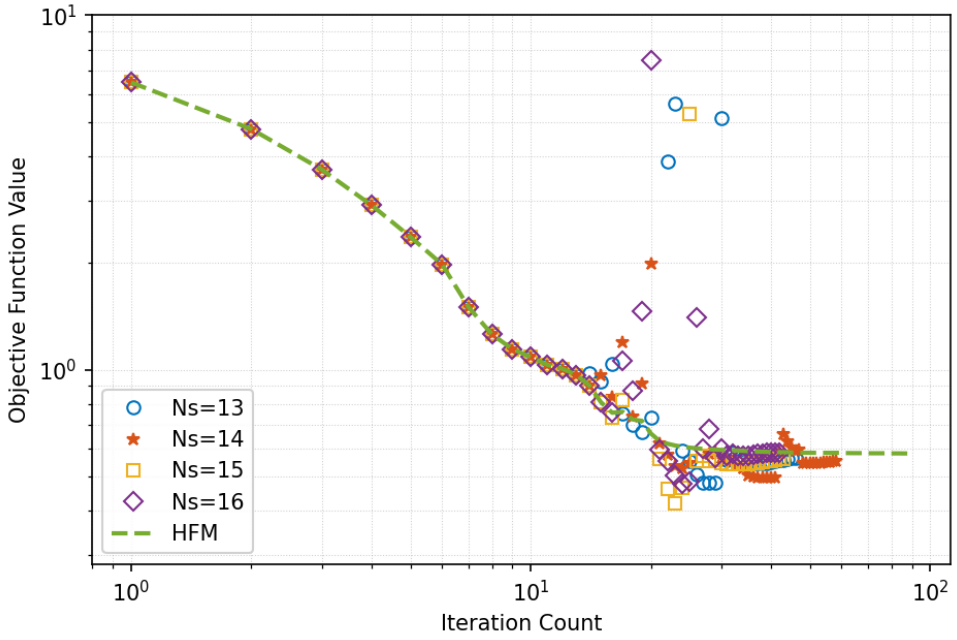


Figure 8: Objective-function convergence for the cantilever beam. Dashed green: high-fidelity baseline. Markers: surrogate-based runs at $N_s \in \{13, 14, 15, 16\}$ initial POD snapshots.

scope and would require the interior-point extension noted in Section 6. Third, the reported results display a discernible sensitivity to the initial training-set size N_s that the adaptive scheme reduces but does not eliminate, and the rule for choosing N_s remains empirical. Fourth, the algorithm inherits the cost of high-fidelity adjoint solves whenever the ratio test triggers an enrichment; problems in which a single high-fidelity adjoint solve already exhausts the compute budget will not benefit unless the enrichment criterion is itself weakened. Fifth, the convergence guarantees of the underlying trust-region Newton method [48] carry over modulo the assumption that the reduced gradient is sufficiently accurate at acceptance time; we do not establish certified *a posteriori* bounds on this accuracy, as is done by Keil et al. [15] and Banholzer et al. [16] for the parametric reduced-basis setting. Finally, the implementation reduces only the state and Lagrange-multiplier fields; reduction of the control field, which would be valuable for very high-dimensional designs, is left to future work.

6. Conclusions

This paper presented a novel trust-region algorithm that applies proper orthogonal decomposition techniques to enable PDE-constrained optimization using reduced-order

modeling techniques. The algorithm initially samples high-fidelity state information to compute a POD basis. This POD basis is used to generate a low-fidelity finite element model that enables faster optimization iterations. The algorithm relies on an adaptive scheme based on the ratio between the actual and predicted reduction in the objective function, to control the inexactness induced by the reduced order model during optimization. Furthermore, the adaptive updating scheme enables the algorithm to add additional modes to improve predictive reliability of the orthogonal functions during optimization. However, additional research is needed to understand how to properly select the initial training set. Regardless, results showed that the proposed surrogate-based optimization algorithm has the potential of accelerating the solution of large-scale PDE-constrained optimization problems.

The algorithm was applied to three compliance minimization problems in topology optimization. Results showed that the algorithm consistently produced optimal topologies in fewer optimization iterations while converging to the expected solution. Furthermore, the algorithm reduced the number of high-fidelity FEM evaluations performed during optimization. This produced noticeable speedups, which is crucial for large-scale PDE-constrained optimization problems due to the computational demands of high-fidelity model evaluations. The algorithm is also general and can be effectively applied in multiple PDE-constrained optimization settings, and can be combined with Krylov subspace recycling methods to further expedite the optimization problem. Future research focuses on combining Krylov subspace recycling methods with the proposed surrogate-based optimization algorithm.

This paper also presented a linear Hessian formulation for compliance minimization problems in topology optimization. The trust-region framework relies on a quadratic model to predict the behavior of the objective function within a suitable neighborhood. If a nonlinear Hessian formulation is applied, the computational effort will increase due to the additional finite element evaluations performed during the trust-region sub-problem iterations. Replacing the nonlinear Hessian formulation with its linear counterpart led to significant speedups (over $11.5\times$) without negatively impacting the quality of the solution.

The proposed algorithm can be further improved. First, more work is needed to enable optimal sampling strategies to reduce the sensitivity of the solution with respect to the initial sampling size. Second, mathematically sound techniques will be investigated to improve the predictive reliability of the orthogonal functions used to create the reduced-order models. This will allow the algorithm to rely longer on the reduced-order models and avoid computationally intensive singular value or eigenvalue problems. Third, further speedups can be attained by applying reduced-order models to both state and control variables. Therefore, expanding the current trust-region framework to allow the reduction of both state and control variables will be explored in the future. A general constraint modeling scheme based on interior-point methods will also be incorporated to support accurate modeling of general constraints.

Acknowledgements

The early phase of this work was supported by the DOE NNSA Advanced Simulation & Computing (ASC) program at Sandia National Laboratories.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the author used Claude in order to format a complete draft, check for errors in derivations, and improve quality of typeset figures. After using this tool/service, the author reviewed and edited the content as needed and takes full responsibility for the content of the publication.

Code and data availability

A MATLAB implementation of the algorithm and the input data required to reproduce the numerical experiments reported in Section 4 are available from the corresponding author on reasonable request.

Appendix A. Notation

For reference, Table 8 collects the principal symbols used in the paper.

Appendix B. Second-order derivative operators

Equation 8 shows the 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 for completeness. This set of derivative operators were used to solve the high-fidelity, nonlinear compliance minimization topology optimization problem in Section 4.1.

Throughout this section, $\mathbf{K}_{\mathbf{z}}(\mathbf{z})$ denotes the third-rank tensor $(\partial K_{ij}/\partial z_k)$. The contraction $\mathbf{K}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z}$ returns the matrix-valued directional derivative $\sum_k (\partial \mathbf{K}/\partial z_k)\delta z_k$ (contracting the k-index), and $\mathbf{K}_{\mathbf{zz}}(\mathbf{z})\delta\mathbf{z}$ is defined analogously. The decoration * has two distinct uses: on a matrix it denotes the ordinary transpose, while on the tensor $\mathbf{K}_{\mathbf{z}}(\mathbf{z})$ itself it denotes the tensor transpose viewed as a linear map (i.e., a contraction on a different axis). In the compliance setting $\mathbf{K}(\mathbf{z})$ is symmetric, so $\mathbf{K}(\mathbf{z})^* = \mathbf{K}(\mathbf{z})$ and $(\mathbf{K}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z})^* = \mathbf{K}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z}$ for every $\delta\mathbf{z}$; the * is retained in the general expressions below so they remain valid outside the symmetric case.

The necessary equality constraint derivative operators are given by

$$g_{\mathbf{z}}(\mathbf{u}(\mathbf{z}), \mathbf{z})\delta\mathbf{z} = (\mathbf{K}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z})\mathbf{u} \tag{48}$$

$$g_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z})\delta\mathbf{u} = \mathbf{K}(\mathbf{z})\delta\mathbf{u} \tag{49}$$

Symbol	Meaning
$\mathbf{z}$	Control (design) variable
$\hat{\mathbb{Z}}$	Feasible (bound-constrained) control set
$\mathbf{u}$	State variable; $\bar{\mathbf{u}}$ denotes its POD approximation
$\mathbf{v}$, $\mathbf{\Lambda}$	Lagrange multiplier and its POD approximation
$J(\mathbf{u}, \mathbf{z})$	Objective functional
$g(\mathbf{u}, \mathbf{z})$	PDE (equality) constraint
$\mathcal{L}(\mathbf{u}, \mathbf{z}, \mathbf{v})$	Lagrangian
$m(\mathbf{s}_n)$	Quadratic trust-region model of J
ρ_n	Ratio of actual to predicted reduction
$\mathbf{s}_n$, $\mathbf{w}_n$, $\mathbf{w}_n^C$	Trial step, descent direction, Cauchy step
Δ_n	Trust-region radius
η_1, η_2	Acceptance / very-good-step thresholds for ρ_n
$\sigma_1, \sigma_2, \sigma_3$	Trust-region radius update factors
μ_0, μ_1	Projected-search parameters
υ	ROM-active acceptance threshold
$\mathbf{K}(\mathbf{z})$, $\mathbf{D}(\mathbf{z})$	Stiffness matrix and elasticity matrix
$E_0, E_{\min}, p$	SIMP parameters (full / void modulus, penalization)
ζ, ν, τ	Regularization coefficient, smoothing parameter, exponent in $R(\mathbf{z})$
$\varepsilon_{\mathbf{K}}, \varepsilon_V$	Compliance and volume-misfit weights
v_f	Target volume fraction
$(\widehat{\varphi}_i)_q, (\theta_i)_q$	POD basis functions and modal coefficients for the state
N_s	Initial training-set (snapshot) size; M counts collected snapshots, N is the target
$\hat{\epsilon}_u$	POD energy threshold for basis truncation
$N_{itr}, N_{HFM}, N_{UPD}$	Outer iterations, high-fidelity FEM evaluations, POD updates
$\mathcal{S}$	Wall-clock speedup relative to the HFM baseline
$\mathcal{C}$	Active convergence criterion at termination

Table 8: Principal symbols used in the paper.

$$g_{\mathbf{u}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^* \mathbf{v} = \mathbf{K}(\mathbf{z})^* \mathbf{v} \tag{50}$$

$$g_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^* \delta\mathbf{u} = \mathbf{0} \tag{51}$$

$$g_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^* \delta\mathbf{z} = (\mathbf{K}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z})^* \mathbf{v} \tag{52}$$

$$g_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^* \delta\mathbf{z} = ((\mathbf{K}_{\mathbf{zz}}(\mathbf{z})\delta\mathbf{z})^* \mathbf{v})\mathbf{u} \tag{53}$$

$$g_{\mathbf{zu}}(\mathbf{u}(\mathbf{z}), \mathbf{z})^* \delta\mathbf{u} = (\mathbf{K}_{\mathbf{z}}(\mathbf{z})^* \mathbf{v})\delta\mathbf{u}. \tag{54}$$

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

$$J_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}), \mathbf{z})\delta\mathbf{u} = \mathbf{K}(\mathbf{z})\delta\mathbf{u} \tag{55}$$

$$J_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}), \mathbf{z})\delta\mathbf{z} = \varepsilon_{\mathbf{K}}(\mathbf{K}_{\mathbf{z}}(\mathbf{z})\delta\mathbf{z})\mathbf{u} \tag{56}$$

$$J_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = \frac{\varepsilon_{\mathbf{K}}}{2}\mathbf{u}(\mathbf{K}_{\mathbf{zz}}(\mathbf{z})\delta\mathbf{z})\mathbf{u} + \varepsilon_V\, V_{\mathbf{z}}(\mathbf{z})\big(V_{\mathbf{z}}(\mathbf{z})\cdot\delta\mathbf{z}\big) + \varepsilon_V\big(V(\mathbf{z})-V_0\big)V_{\mathbf{zz}}(\mathbf{z})\delta\mathbf{z} + R_{\mathbf{zz}}(\mathbf{z})\delta\mathbf{z} \tag{57}$$

$$J_{\mathbf{zu}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{u} = \varepsilon_{\mathbf{K}}\big(\mathbf{K}_{\mathbf{z}}(\mathbf{z})^*\delta\mathbf{u}\big)\mathbf{u}. \tag{58}$$

The second-order derivative operators of the volume misfit term and the regularization functional in Equation 57 are respectively given by

$$V_{\mathbf{zz}}(\mathbf{z}) = \mathbf{0} \tag{59}$$

and (the exact second Fréchet derivative, applied to a variation $\delta\mathbf{z}$)

$$\begin{aligned} R_{\mathbf{zz}}(\mathbf{z})\,\delta\mathbf{z} = {} & 4\zeta(\tau-1)\int_\Omega \big(|\nabla\mathbf{z}|^2+\nu^2\big)^{\tau-2}(\nabla\mathbf{z}\cdot\nabla\delta\mathbf{z})\,\nabla\mathbf{z}\cdot\nabla\psi\,d\Omega \\ & + 2\zeta\int_\Omega \big(|\nabla\mathbf{z}|^2+\nu^2\big)^{\tau-1}\nabla\delta\mathbf{z}\cdot\nabla\psi\,d\Omega. \end{aligned} \tag{60}$$

After discretization, this corresponds to the matrix

$$R_{\mathbf{zz}}(\mathbf{z}) = 2\zeta\,\mathbf{B}_\varphi(\mathbf{z}) + 4\zeta(\tau-1)\,\mathbf{C}_\varphi(\mathbf{z}), \tag{61}$$

where $\mathbf{B}_\varphi(\mathbf{z})$ is given by Equation 26 and $\mathbf{C}_\varphi(\mathbf{z})$ is the rank-update matrix

$$\mathbf{C}_\varphi(\mathbf{z})_{ij} = \int_\Omega \big(|\nabla\mathbf{z}|^2+\nu^2\big)^{\tau-2}(\nabla\mathbf{z}\cdot\nabla\psi_i)(\nabla\mathbf{z}\cdot\nabla\psi_j)\,d\Omega. \tag{62}$$

Parameters ζ, ν, and τ were defined in Section 2.2.

The stiffness matrix $\mathbf{K}$ for linear elastostatics is given by

$$\mathbf{K}(\mathbf{z}) = \int_\Omega \nabla\phi_i\,\mathbf{D}(\mathbf{z})\nabla\phi_j\,d\Omega, \tag{63}$$

where the elasticity matrix $\mathbf{D}(\mathbf{z})$ is given by

$$\mathbf{D}(\mathbf{z}) = \frac{E_{\min}+\mathbf{z}^p(E_0-E_{\min})}{(1+\mathtt{v})(1-2\mathtt{v})}\begin{bmatrix} 1-\mathtt{v} & \mathtt{v} & \mathtt{v} & 0 & 0 & 0 \\ \mathtt{v} & 1-\mathtt{v} & \mathtt{v} & 0 & 0 & 0 \\ \mathtt{v} & \mathtt{v} & 1-\mathtt{v} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\mathtt{v}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\mathtt{v}}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\mathtt{v}}{2} \end{bmatrix}, \tag{64}$$

where $\mathtt{v}$ denotes the Poisson's ratio. The expression above is the full three-dimensional isotropic stiffness matrix in Voigt notation; the two-dimensional test problems in Section 4 use the corresponding plane-strain restriction obtained by retaining the upper-left 2×2

block of the normal entries together with the in-plane shear entry $\frac{1-2\mathrm{v}}{2}$. The first and second-order derivative operators for $\mathbf{K}_{\mathbf{z}}(\mathbf{z})$ and $\mathbf{K}_{\mathbf{zz}}(\mathbf{z})$ are respectively given by

$$\mathbf{K}_{\mathbf{z}}(\mathbf{z}) = \int_{\Omega} \nabla\phi_i \, \mathbf{D}'(\mathbf{z}) \nabla\phi_j \, d\Omega \tag{65}$$

and

$$\mathbf{K}_{\mathbf{zz}}(\mathbf{z}) = \int_{\Omega} \nabla\phi_i \, \mathbf{D}''(\mathbf{z}) \nabla\phi_j \, d\Omega. \tag{66}$$

where,

$$\mathbf{D}'(\mathbf{z}) = \frac{p\,\mathbf{z}^{p-1}(E_0 - E_{\min})}{(1+\mathrm{v})(1-2\mathrm{v})} \begin{bmatrix} 1-\mathrm{v} & \mathrm{v} & \mathrm{v} & 0 & 0 & 0 \\ \mathrm{v} & 1-\mathrm{v} & \mathrm{v} & 0 & 0 & 0 \\ \mathrm{v} & \mathrm{v} & 1-\mathrm{v} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\mathrm{v}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\mathrm{v}}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\mathrm{v}}{2} \end{bmatrix}, \tag{67}$$

and

$$\mathbf{D}''(\mathbf{z}) = \frac{p(p-1)\,\mathbf{z}^{p-2}(E_0 - E_{\min})}{(1+\mathrm{v})(1-2\mathrm{v})} \begin{bmatrix} 1-\mathrm{v} & \mathrm{v} & \mathrm{v} & 0 & 0 & 0 \\ \mathrm{v} & 1-\mathrm{v} & \mathrm{v} & 0 & 0 & 0 \\ \mathrm{v} & \mathrm{v} & 1-\mathrm{v} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\mathrm{v}}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\mathrm{v}}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\mathrm{v}}{2} \end{bmatrix}. \tag{68}$$

Recall from Section 2.1 that steps 4–6 are required to compute the application of the trial step to the nonlinear Hessian operator given by Equation 8. Substituting Equation 10 into Equation 12 yields

$$\begin{aligned} \delta\mathbf{v} = -\mathcal{L}_{\mathbf{uv}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})^{-1}[\mathcal{L}_{\mathbf{uu}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})(-g_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^{-1} g_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z}) \\ + \mathcal{L}_{\mathbf{uz}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{z}]. \end{aligned} \tag{69}$$

An explicit expression for $\delta\mathbf{v}$ is derived by substituting Equations 51 and 55 into Equation 14 and Equations 52 and 56 into Equation 15. This explicit expression is given by

$$\delta\mathbf{v} = -g_{\mathbf{u}}(\mathbf{u}(\mathbf{z}),\mathbf{z})^{-1}(g_{\mathbf{z}}(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z}) \equiv \delta\mathbf{u}. \tag{70}$$

The application of the trial step to the nonlinear Hessian operator is obtained by substituting Equations 28, 54, 58, and 70 into Equation 8. This expression is given by

$$\nabla^2 J(\mathbf{u}(\mathbf{z}),\mathbf{z})\delta\mathbf{z} = \mathcal{L}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{z} + \mathcal{L}_{\mathbf{zv}}(\mathbf{u}(\mathbf{z}),\mathbf{z},\mathbf{v})\delta\mathbf{u}, \tag{71}$$

where $\mathcal{L}_{\mathbf{zz}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v})$ is given by Equation 32 and $\mathcal{L}_{\mathbf{zv}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v})$ is given by

$$\mathcal{L}_{\mathbf{zv}}(\mathbf{u}(\mathbf{z}), \mathbf{z}, \mathbf{v}) = \mathbf{K}_{\mathbf{z}}(\mathbf{z})^{*}\mathbf{u}. \tag{72}$$

For compliance minimization topology optimization problems, one additional FEM evaluation is required to compute the application of the trial step to the nonlinear Hessian operator.

References

[1] P. Benner, S. Gugercin, K. Willcox, A survey of projection-based model reduction methods for parametric dynamical systems, SIAM Review 57 (4) (2015) 483–531. doi:10.1137/130932715.

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

[3] K. Carlberg, C. Farhat, J. Cortial, 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) (2013) 623 – 647.

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

[5] 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 (1987) 561–571.

[6] Z. Ostrowski, R. Bialecki, A. 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.

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

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

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

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

[11] E. Arian, M. Fahl, E. W. Sachs, Trust-region proper orthogonal decomposition for flow control, Tech. rep., DTIC Document (2000).

[12] M. Fahl, Trust-region methods for flow control based on reduced order modeling, Ph.D. thesis, Ph. D. thesis, Trier university (2000).

[13] B. Kragel, Streamline diffusion pod models in optimization, PhD, University of Trier.

[14] M. J. Zahr, C. Farhat, Progressive construction of a parametric reduced-order model for PDE-constrained optimization, International Journal for Numerical Methods in Engineering 102 (5) (2015) 1077–1110. doi:10.1002/nme.4770.

[15] T. Keil, L. Mechelli, M. Ohlberger, F. Schindler, S. Volkwein, A non-conforming dual approach for adaptive trust-region reduced basis approximation of PDE-constrained parameter optimization, ESAIM: Mathematical Modelling and Numerical Analysis 55 (3) (2021) 1239–1269. doi:10.1051/m2an/2021019.

[16] S. Banholzer, T. Keil, L. Mechelli, M. Ohlberger, F. Schindler, S. Volkwein, An adaptive projected Newton non-conforming dual approach for trust-region reduced basis approximation of PDE-constrained parameter optimization, Pure and Applied Functional Analysis 7 (5) (2022) 1561–1596.

[17] M. Yano, Goal-oriented model reduction of parametrized nonlinear partial differential equations: Application to aerodynamics, International Journal for Numerical Methods in Engineering 121 (23) (2020) 5320–5347. doi:10.1002/nme.6395.

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

[19] G. H. Yoon, Structural topology optimization for frequency response problem using model reduction schemes, Computer Methods in Applied Mechanics and Engineering 199 (25) (2010) 1744–1763.

[20] N. Gerzen, F.-J. Barthold, Enhanced analysis of design sensitivities in topology optimization, Structural and Multidisciplinary Optimization 46 (4) (2012) 585–595.

[21] C. Gogu, Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction, International Journal for Numerical Methods in Engineering 101 (4) (2015) 281–304. doi:10.1002/nme.4797.

[22] M. Xiao, D. Lu, P. Breitkopf, B. Raghavan, S. Dutta, W. Zhang, On-the-fly model reduction for large-scale structural topology optimization using principal components analysis, Structural and Multidisciplinary Optimization 62 (1) (2020) 209–230. doi:10.1007/s00158-019-02485-3.

[23] Y. Choi, G. Boncoraglio, S. Anderson, D. Amsallem, C. Farhat, Gradient-based constrained optimization using a database of linear reduced-order models, Journal of Computational Physics 423 (2020) 109787. doi:10.1016/j.jcp.2020.109787.

[24] N. Ferro, M. Bertin, K. Lampoh, M. Tarallo, Efficient buckling constrained topology optimization using reduced order modeling, Structural and Multidisciplinary Optimization 66 (2023) 171. doi:10.1007/s00158-023-03616-7.

[25] D. Amsallem, M. Zahr, Y. Choi, C. Farhat, Design optimization using hyper-reduced-order models, Structural and Multidisciplinary Optimization (2014) 1–22.

[26] Y. Yue, K. Meerbergen, Accelerating optimization of parametric linear systems by model order reduction, SIAM Journal on Optimization 23 (2) (2013) 1344–1370.

[27] M. J. Zahr, D. Amsallem, C. Farhat, Construction of parametrically-robust cfd-based reduced-order models for pde-constrained optimization, AIAA Paper 2845 (2013) 26–29.

[28] C. Lieberman, K. Willcox, O. Ghattas, Parameter and state model reduction for large-scale statistical inverse problems, SIAM Journal on Scientific Computing 32 (5) (2010) 2523–2542.

[29] T. Bui-Thanh, K. Willcox, O. Ghattas, Model reduction for large-scale systems with high-dimensional parametric input space, SIAM Journal on Scientific Computing 30 (6) (2008) 3270–3288.

[30] B. Peherstorfer, K. Willcox, Data-driven operator inference for nonintrusive projection-based model reduction, Computer Methods in Applied Mechanics and Engineering 306 (2016) 196–215. doi:10.1016/j.cma.2016.03.025.

[31] S. Chaturantabut, D. C. Sorensen, Nonlinear model reduction via discrete empirical interpolation, SIAM Journal on Scientific Computing 32 (5) (2010) 2737–2764. doi:10.1137/090766498.

[32] C. Farhat, P. Avery, T. Chapman, J. Cortial, Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency, International Journal for Numerical Methods in Engineering 98 (9) (2014) 625–662. doi:10.1002/nme.4668.

[33] M. Yano, A. T. Patera, An LP empirical quadrature procedure for reduced basis treatment of parametrized nonlinear PDEs, Computer Methods in Applied Mechanics and Engineering 344 (2019) 1104–1123. doi:10.1016/j.cma.2018.02.028.

[34] B. Peherstorfer, K. Willcox, M. Gunzburger, Survey of multifidelity methods in uncertainty propagation, inference, and optimization, SIAM Review 60 (3) (2018) 550–591. doi:10.1137/16M1082469.

[35] D. Bertsekas, Nonlinear Programming, Athena Scientific, 1999.

[36] K. Liu, A. Tovar, An efficient 3d topology optimization code written in matlab, Structural and Multidisciplinary Optimization 50 (6) (2014) 1175–1196.

[37] A. Evgrafov, On the reduced hessian of the compliance, Structural and Multidisciplinary Optimization 50 (6) (2014) 1197–1199. doi:10.1007/s00158-014-1204-x.
URL http://dx.doi.org/10.1007/s00158-014-1204-x

[38] J. Nocedal, S. J. Wright, Numerical Optimization, Springer, 2006.

[39] R. T. Rockafellar, Lagrange multipliers and optimality, SIAM Review 35 (1993) 183–238.

[40] V. Danilov, Encyclopedia of Mathematics, Springer, 2001, Ch. Implicit function (in algebraic geometry).

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

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

[43] R. Haber, C. Jog, M. P. Bendsøe, A new approach to variable-topology shape design using a constraint on perimeter, Structural Optimization 11 (1-2) (1996) 1–12.

[44] O. Sigmund, Morphology-based black and white filters for topology optimization, Structural and Multidisciplinary Optimization 33 (4-5) (2007) 401–424. doi:10.1007/s00158-006-0087-x.
URL http://dx.doi.org/10.1007/s00158-006-0087-x

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

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

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

[48] C.-J. Lin, J. J. Moré, Newton's method for large bound-constrained optimization problems, SIAM Journal on Optimization 9 (4) (1999) 1100–1127.

[49] A. R. Conn, N. I. M. Gould, P. L. Toint, Trust region methods, Vol. 1, SIAM, 2000.

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

[51] J. J. Moré, Trust regions and projected gradients, in: System Modelling and Optimization, Springer, 1988, pp. 1–13.

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

[53] Y. Saad, Iterative methods for sparse linear systems, Siam, 2003.

[54] M. J. Powell, A new algorithm for unconstrained optimization, Nonlinear programming (1970) 31–65.

[55] J. Dennis Jr, H. Mei, Two new unconstrained optimization algorithms which use function and gradient values, Journal of Optimization Theory and Applications 28 (4) (1979) 453–482.

[56] J. J. Moré, G. Toraldo, On the solution of large quadratic programming problems with bound constraints, SIAM Journal on Optimization 1 (1) (1991) 93–113.

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

[58] 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, A. Williams, An Overview of Trilinos, SAND2003-2927, Sandia National Laboratories, P.O. Box 5800, Albuquerque, NM 87185-1110 (2003).

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

[60] B. Bourdin, Filters in topology optimization, International Journal for Numerical Methods in Engineering 50 (9) (2001) 2143–2158. doi:10.1002/nme.116.

[61] T. E. Bruns, D. A. Tortorelli, Topology optimization of non-linear elastic structures and compliant mechanisms, Computer Methods in Applied Mechanics and Engineering 190 (26–27) (2001) 3443–3459. doi:10.1016/S0045-7825(00)00278-4.

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

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

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