LAWRENCE
LIVERMORE
NATIONAL
LABORATORY

LLNL-CONF-771564

Accelerating topology optimization using reduced order models

Y. Choi, G. Oxberry, D. White, T. Kirchdoerfer

April 9, 2019

World Congress of Structural and Multidisciplinary Optimization
Beijing, China
May 20, 2019 through May 24, 2019

Accelerating topology optimization using reduced order models

Youngsoo Choi[1]*, **Geoffrey Oxberry**[1], **Daniel White**[1], **Trenton Thomas Kirchdoerfer**[1]

[1]*Lawrence Livermore National Laboratory, 7000 East Ave, Livermore, CA 94551, USA*
* *Corresponding author: choi15@llnl.edu*

Abstract
We present a ROM-based topology optimization algorithm that enables a fast design process. The algorithm replaces high-fidelity models with reduced order models in the beginning of the optimization process, allowing inexact linear system solutions. When the optimization gets near the end, the Krylov subspace method with ROM-recycling is invoked for a precise solution with reduced number of iterations. The stage of the optimization process is quantified by the norm of the KKT optimality condition. Our method is general enough to be applicable to not only topology optimization, but also PDE-constrained optimization, i.e., a broader class of physics-based optimization. We have applied our method to two 3D design problems, showing a considerable speed-up and reduction in linear iterations and achieving an optimal design that satisfies the KKT optimality condition.
Keywords: *reduced order model, topology optimization, Krylov subspace method with recycling, inexact linear solve*

1. Introduction

Topology optimization is a powerful design tool that enables an automatic process of obtaining an optimal design with the help of sophisticated computational models. It is widely used in industries, academia, laboratories for various applications, mainly including aerospace, structural, mechanical engineering, but expanding to fluids, acoustics, electromagnetics, and optics. The optimal designs from topology optimization can be directly manufactured thanks to the recent advances in additive manufacturing. However, the topology optimization process is slow because an expensive solution process of physics model is involved. The most expensive part of the solution process is a sequence of large-scale linear solves:

$$\boldsymbol{A}_{(k)}\boldsymbol{x}_{(k)} = \boldsymbol{b}_{(k)}. \tag{1}$$

Although a sparse linear system arises, the size of the system hinders a rapid design process. Therefore, developing a method of accelerating the sequence of large-scale linear solve is essential. There have been many attempts to reduce the cost of large-scale linear solves (1). They can be grouped to two categories: i) allowing inexact linear solves and ii) reducing the number of iterations in Krylov subspace methods.

As for allowing inexact linear solves in topology optimization, Amir and his coauthors in [4] studied how to utilize the inexact solution of linear systems in the optimization process by considering specific convergence criteria to compliance and compliant mechanism problems. In order to extend their approach to other applications such as stress constrained problems, appropriate convergence criteria need to be developed. Gogu [5] replaces high-fidelity linear solve with Reduced Order Models (ROMs). Basis is constructed on-the-fly by Gram-Schmidt process whenever the ROM residual is bigger than a tolerance set by a user. In order to set the tolerance relatively a large value, e.g., 0.1, the sensitivity correction terms are added to the ROM sensitivity as in [4]. However, the sensitivity correction terms require n linear solves with size N, where n is the dimension of ROM and N is High-Fidelity Model (HFM) size. Those solves involve system matrices whose factorizations are already known if direct solver is used. However, if linear systems are solved with iterative solvers, those factorizations are not available. Additionally, this ROM method is compared with the academic version of topology optimization algorithm written in MATLAB. Therefore, it is likely that his reported speed-up will be degraded when his ROM-based method is compared with the high-performance computing (HPC)-based HFMs. Gil Ho Yoon in [6] uses various model reduction techniques for frequency response problems. They include the mode superposition, Ritz vector, and quasi-static Ritz vector. All those ROMs mentioned above do not consider the reduced basis from the proper orthogonal decomposition, which is known to give you an optimal basis, given a solution data.

The Krylov subspace methods with recycling has been a way of reducing the number of iterations. In topology optimization, Wang and his coauthors in [1] used MINRES with recycling to accelerate the solution process of both symmetric positive-definite and indefinite systems. Scaling of stiffness was used to bring down the condition number of stiffness matrix. They used incomplete Cholesky factorization as a preconditioner. However, the recycling subspace was taken from the solutions of previous linear solves, not from reduced basis of previous solution, resulting in a bigger recycling space. On the other hand, in the fields other than topology optimization, the Krylov subspace methods with recycling have been used to accelerate the linear solve process. Especially the following two works, [2] and [3], use the subspace spanned by reduced basis of previous solutions as the recycling space.

The gradient-based optimization solvers start with an initial design and explore the design space until it finds an optimal solution that satisfies the KKT conditions. In the beginning of the optimization process, the change of design variables is large, indicating that the linear solves at this stage do not have to be solved precisely. This is exactly the motivation for various gradient-based optimization algorithms using various inexactness (e.g., see [7,8,9]). Thus, we will replace HFM with a ROM in the beginning of the optimization process. On the other hand, as the optimization process gets near the end, the change in design variables is small and a precise solution of each linear solve is required for a guarantee of the convergence to a local optimum. For this stage of the optimization process, we will use Krylov subspace methods with ROM-recycling space that is able to solve for an accurate enough solution with a reduced number of iterations. The combination of inexact ROM in the beginning and the Krylov subspace methods with ROM-recycling in the end will accelerate the overall topology optimization process.

The rest of the paper is organized in the following way: Section 2 will present the details of our method, the ROM-based topology optimization algorithm. The numerical results will demonstrate several advantages of our method in Section 3. Then, we will summarize and conclude the paper in Section 4.

2. ROM-based topology optimization

To accelerate the overall topology optimization, we develop a ROM-based topology optimization process that applies two machineries: i) inexact linear solves by ROMs and ii) Krylov subspace methods with ROM-recycling. The overall ROM-based topology optimization flow chart is shown in Figure 1.

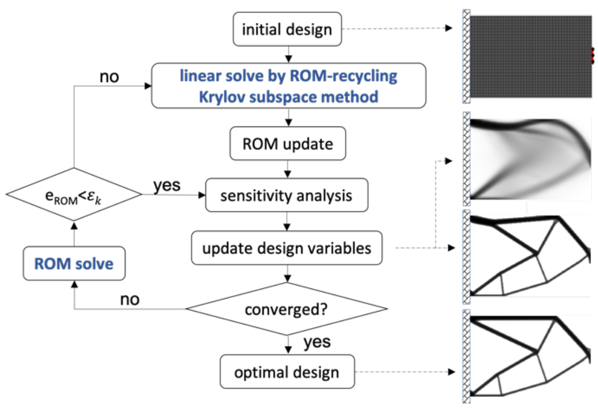

Figure 1. ROM-based topology optimization flow chart

The topology optimization solves a sequence of linear system of equations (1) that generates a sequence of solutions, $\boldsymbol{x}_{(k)}$. The sequence of solutions converges to an optimal solution, $\boldsymbol{x}_{\star}$, for a feasible problem. Additionally, as the optimization process is close to the end, the sequence of solutions does not change much, i.e., $\|\boldsymbol{x}_{(k)} - \boldsymbol{x}_{(k+1)}\|_2 \leq \epsilon$ for $k > K > 0$ with a number, K, and a small number, ϵ. Therefore, finding the solution $\boldsymbol{x}_{(k)}$ within the subspace spanned by the previous ℓ solutions, $\mathcal{A}_{k-1}^{\ell} := \mathrm{span}\{\boldsymbol{x}_{(k-\ell)}, \ldots, \boldsymbol{x}_{(k-1)}\}$, must give a good approximation to $\boldsymbol{x}_{(k)}$.

2.1 Proper Orthogonal Decomposition (POD)-based ROM

For the last two decades, there have been tremendous development in projection-based model order reduction techniques that achieve an order of ten to thousand speedups without losing much accuracy for various physics simulations, including, but not limited to, computational fluid and solid mechanics. Among many, the POD-based ROM has shown promising capability of accelerating physics simulations, e.g., [10, 11]. POD [12] generates a basis matrix, $\boldsymbol{\Phi} \in \mathbb{R}^{N\times n}$, that minimizes the difference between solution snapshots and projected ones. In other words, POD generates the optimal subspace of dimension, $n \ll N$. The solution to the POD problem is well-known, i.e., a truncated orthogonal left singular matrix from Singular Value Decomposition (SVD) of the solution snapshot matrix. For our context, we can build $\boldsymbol{\Phi}_{(k)}$ by the POD with the previous ℓ solution snapshot matrix, i.e., $\boldsymbol{S}_{k-1}^{\ell} = \begin{bmatrix} \boldsymbol{x}_{(k-\ell)} & \cdots & \boldsymbol{x}_{(k-1)} \end{bmatrix}$. However, computing SVD whenever it needs to update the reduced basis can be computationally expensive. Therefore, we use the incremental SVD [13], shown in Algorithm 3 and 4, that enables a fast update of $\boldsymbol{\Phi}_{(k)}$ on-the-fly and in a memory efficient way. Alternative to incremental SVD, the incremental QR, i.e., Gram-Schmidt process, can be used as in [5]. However, the incremental SVD outperforms the incremental QR in Algorithm 2, which will be demonstrated in Section 3. It is because the incremental SVD gives more optimal basis than incremental QR in the POD sense. Once the basis is available, the corresponding ROM to (1) is built and solved at the stage of **ROM solve** in the optimization flow chart, i.e., Figure 1:

$$\boldsymbol{\Phi}_{(k)}^T \boldsymbol{A}_{(k)} \boldsymbol{\Phi}_{(k)} \hat{\boldsymbol{x}}_{(k)} = \boldsymbol{\Phi}_{(k)}^T \boldsymbol{b}_{(k)}, \tag{2}$$

where $\hat{\boldsymbol{x}}_{(k)} \in \mathbb{R}^n$ is a generalized reduced coordinate vector of the ROM. Then the approximate full solution is restored by $\tilde{\boldsymbol{x}}_{(k)} = \boldsymbol{\Phi}_{(k)}\hat{\boldsymbol{x}}_{(k)}$. The residual function of the ROM solution can be defined as $\boldsymbol{r}_{(k)}(\tilde{\boldsymbol{x}}_{(k)}) = \boldsymbol{b}_{(k)} - \boldsymbol{A}_{(k)}\tilde{\boldsymbol{x}}_{(k)}$, which is not generally zero due to the approximation introduced by the ROM. The norm of the residual can be considered as an error indicator, i.e., $\|\boldsymbol{x}_{(k)} - \tilde{\boldsymbol{x}}_{(k)}\|_2 = \|\boldsymbol{A}_{(k)}^{-1}\boldsymbol{r}_{(k)}(\tilde{\boldsymbol{x}}_{(k)})\| \leq \|\boldsymbol{A}(k)^{-1}\|_2\|\tilde{\boldsymbol{r}}_{(k)}(\tilde{\boldsymbol{x}}_{(k)})\|_2$. Therefore, we use the relative residual norm as an error indicator, i.e., $e_{\mathrm{ROM}} := \|\boldsymbol{r}_{(k)}\|/\|\boldsymbol{b}_{(k)}\|$ in Figure 1. Although we allow an inexact solve by the ROM, the accuracy must be good enough to ensure a sufficient descent search direction. We impose the condition $e_{\mathrm{ROM}} < \epsilon_k$, where ϵ_k is determined by a scalar product of the KKT residual norm, i.e., $\epsilon_k = r_{\mathrm{ROM}} \cdot \|\boldsymbol{r}_{\mathrm{KKT}}\|$ with a reduction factor, $r_{\mathrm{ROM}} < 1 \in \mathbb{R}$. The motivation for this choice is that the early stage of the optimization process can be found by the KKT residual norm. If the KKT residual norm is large, it implies that the current design variables are far from the optimal. Otherwise, it implies that the current design variables are near the optimal. Additionally, we introduce the safeguard threshold, $\epsilon_{\max}$, with which we set $\epsilon_k = \min(r_{\mathrm{ROM}} \cdot \|\boldsymbol{r}_{\mathrm{KKT}}\|, \epsilon_{\max})$ that avoids the adversarial case of $\|\boldsymbol{r}_{\mathrm{KKT}}\|$ being large. Finally, if $e_{\mathrm{ROM}} < \epsilon_k$, we consider that the ROM solution is good enough. Otherwise, we invoke the ROM-recycling Krylov subspace method to compute a better solution than the ROM solution.

2.2 ROM-recycling Krylov subspace method

Various recycling Krylov subspace methods are available in literature, e.g., [1,2,3,14]. For the simplicity of the presentation, we present only the Preconditioned Conjugate-Gradient (PCG) with recycling that can solve symmetric positive-definite systems. The Krylov subspace is defined as $\mathcal{K}_m := \{\boldsymbol{P}^{-1}\boldsymbol{r}, \boldsymbol{P}^{-1}\boldsymbol{A}\boldsymbol{r}, \ldots, \boldsymbol{P}^{-1}\boldsymbol{A}^m\boldsymbol{r}\}$ with a preconditioner $\boldsymbol{P}^{-1}$ and the residual vector $\boldsymbol{r}$. Then the PCG iterative solver solves the following minimization problem in m-th PCG iteration:

$$\boldsymbol{x}^{(m)} = \arg \min_{\boldsymbol{x} \in \boldsymbol{x}^{(0)} + \mathcal{K}_m} \|\boldsymbol{x}^\star - \boldsymbol{x}\|_{\boldsymbol{A}}, \tag{3}$$

where $\boldsymbol{A}\boldsymbol{x}^\star = \boldsymbol{b}$ and $\|\boldsymbol{q}\|_{\boldsymbol{A}} := \sqrt{\boldsymbol{q}^T\boldsymbol{A}\boldsymbol{q}}$ for a symmetric positive definite matrix, $\boldsymbol{A}$. For the brevity of the presentation, we drop the subscript index (k) that represents k-th optimization iteration. Algorithm 1 without the red blocks shows a PCG algorithm. With the red block, the ROM-recycling PCG constructs the Krylov subspace that includes the ROM subspace $\mathcal{A}_{k-1}^{\ell}$ and the subsequent subspace generated by the PCG process. The ROM-recycling PCG enforces the subsequent subspace to be orthogonal to $\mathcal{A}_{k-1}^{\ell}$. As a result, the solution generated by the ROM-recycling PCG can jump start with a ROM solution from **ROM solve** of Figure 1 and the subsequently generated solutions can minimize the error in (3) within the complementary subspace to $\mathcal{A}_{k-1}^{\ell}$. This is motivated by the fact that the ROM solution $\tilde{\boldsymbol{x}}$ from **ROM solve** of Figure 1 is the optimal solution of (3) with $\mathcal{K}_m$ replaced by $\mathcal{A}_{k-1}^{\ell}$. Therefore, if $e_{\mathrm{ROM}} \geq \epsilon_k$, then we use $\tilde{\boldsymbol{x}}$ as an initial guess to the ROM-recycling PCG with the currently available reduced basis $\boldsymbol{\Phi}$. If the topology optimization process is in beginning stage, then the ROM-recycling PCG does not need to solve for a precise solution. Thus, we set ϵ_{rel}, one of the convergence criteria parameters for the PCG to be a scalar product of the KKT residual norm, i.e., $\epsilon_{\mathrm{rel}} = r_{\mathrm{CG}}\|\boldsymbol{r}_{\mathrm{KKT}}\|$ with a reduction factor, $r_{\mathrm{CG}} < 1$. Additionally, we introduce the safeguard threshold, ϵ_{CG}, with which we set $\epsilon_{\mathrm{rel}} = \min(r_{\mathrm{CG}}\|\boldsymbol{r}_{\mathrm{KKT}}\|, \epsilon_{\mathrm{CG}})$ that avoids the adversarial case of $\|\boldsymbol{r}_{\mathrm{KKT}}\|$ being large.

Algorithm 1 ROM-recycling_pcg

Input: $\boldsymbol{A}$, $\boldsymbol{b}$, $\boldsymbol{x}$, ϵ_{rel}, ϵ_{abs}, N_{maxit}, $\boldsymbol{P}^{-1}$, $\boldsymbol{\Phi}$
Output: $\boldsymbol{x}$

$\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}$
$\boldsymbol{p} = \boldsymbol{P}^{-1}\boldsymbol{r}$
solve $\boldsymbol{\Phi}^T\boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{\mu} = \boldsymbol{\Phi}^T\boldsymbol{A}\boldsymbol{p}$
$\boldsymbol{p} = \boldsymbol{p} - \boldsymbol{\Phi}\boldsymbol{\mu}$
$r_0 = \max\{\epsilon_{\mathrm{rel}}^2\boldsymbol{b}^T\boldsymbol{b}, \epsilon_{\mathrm{abs}}^2\}$
$\zeta = \boldsymbol{r}^T\boldsymbol{p}$
if $\zeta \le r_0$ **then**
  converged
end if
$\boldsymbol{z} = \boldsymbol{A}\boldsymbol{p}$
$\gamma = \boldsymbol{z}^T\boldsymbol{p}$
if $\gamma = 0$ **then**
  fail to converge
end if
for $i = 1, \ldots, N_{\mathrm{maxit}}$ **do**
  $\alpha = \frac{\zeta}{\gamma}$
  $\boldsymbol{x} = \boldsymbol{x} + \alpha\boldsymbol{p}$
  $\boldsymbol{r} = \boldsymbol{r} - \alpha\boldsymbol{A}\boldsymbol{p}$
  $\boldsymbol{z} = \boldsymbol{P}^{-1}\boldsymbol{r}$
  $\eta = \boldsymbol{r}^T\boldsymbol{z}$
  if $\eta < r_0$ **then**
    converged
  end if
  $\beta = \frac{\eta}{\zeta}$
  $\boldsymbol{p} = \boldsymbol{z} + \beta\boldsymbol{p}$
  solve $\boldsymbol{\Phi}^T\boldsymbol{A}\boldsymbol{\Phi}\boldsymbol{\mu} = \boldsymbol{\Phi}^T\boldsymbol{A}\boldsymbol{z}$
  $\boldsymbol{p} = \boldsymbol{p} - \boldsymbol{\Phi}\boldsymbol{\mu}$
  $\boldsymbol{z} = \boldsymbol{A}\boldsymbol{p}$
  $\gamma = \boldsymbol{p}^T\boldsymbol{z}$
  if $\gamma \le 0$ **then**
    not positive definite
  end if
  $\zeta = \eta$
end for

Algorithm 2 Incremental QR

Input: $\boldsymbol{c}, \epsilon_{\mathrm{QR}}$
Output: $\boldsymbol{\Phi}$

if $k = 0$ **then**
  $\boldsymbol{\Phi} \leftarrow \boldsymbol{c}/\|\boldsymbol{c}\|$
  $k \leftarrow 1$
  return
else
  $\boldsymbol{j} \leftarrow \boldsymbol{c} - \boldsymbol{\Phi}\boldsymbol{\Phi}^T\boldsymbol{c}$
  $\boldsymbol{\Phi} \leftarrow \begin{bmatrix}\boldsymbol{\Phi} & \boldsymbol{j}/\|\boldsymbol{j}\|\end{bmatrix}$
  $k \leftarrow k + 1$
end if

Algorithm 3 Initializing incremental SVD

Input: $\boldsymbol{c}, \epsilon_{\mathrm{SVD}}$
Output: $\boldsymbol{s}, \boldsymbol{\Phi}, \boldsymbol{\Psi}$

if $\|\boldsymbol{c}\| > \epsilon_{\mathrm{SVD}}$ **then**
  $\boldsymbol{s} \leftarrow [\|\boldsymbol{c}\|]$
  $\boldsymbol{\Phi} \leftarrow \boldsymbol{c}/\|\boldsymbol{s}\|$
  $\boldsymbol{\Psi} \leftarrow [1]$
else
  $\boldsymbol{s} \leftarrow []$
  $\boldsymbol{\Phi} \leftarrow []$
  $\boldsymbol{\Psi} \leftarrow []$
end if

Algorithm 4 Incremental SVD

Input: $\boldsymbol{c}, \epsilon_{\mathrm{SVD}}, \boldsymbol{\Phi}, \boldsymbol{s}, \boldsymbol{\Psi}$
Output: $\boldsymbol{\Phi}, \boldsymbol{s}, \boldsymbol{\Psi}$

if $k = 0$ or $k = m_{\Phi}$ **then**
  Apply Algorithm 3
  return $\boldsymbol{\Phi}, \boldsymbol{s}, \boldsymbol{\Psi}$
end if
$\boldsymbol{\ell} \leftarrow \boldsymbol{\Phi}^T\boldsymbol{c}$
$p \leftarrow \sqrt{\boldsymbol{c}^T\boldsymbol{c} - \boldsymbol{\ell}^T\boldsymbol{\ell}}$
$\boldsymbol{j} \leftarrow (\boldsymbol{c} - \boldsymbol{\Phi}\boldsymbol{\ell})/p$
$\boldsymbol{Q} \leftarrow \begin{bmatrix}\mathrm{s} & \boldsymbol{\ell} \\ \boldsymbol{0} & p\end{bmatrix}$
if $p < \epsilon_{\mathrm{SVD}}$ **then**
  $\boldsymbol{Q}_{\mathrm{end,end}} \leftarrow 0$
end if
$[\boldsymbol{\Phi}', \mathrm{s}', \boldsymbol{\Psi}'^T] \leftarrow \mathrm{SVD}(\boldsymbol{Q})$

if $p < \epsilon_{\mathrm{SVD}}$ **then**
  $\boldsymbol{\Phi} \leftarrow \boldsymbol{\Phi}\boldsymbol{\Phi}'_{1:k,1:k}$
  $\boldsymbol{s} \leftarrow \boldsymbol{s}'_{1:k}$
  $\boldsymbol{\Psi} \leftarrow \boldsymbol{\Psi}'_{1:k,:}$
else
  $\boldsymbol{\Phi} \leftarrow \begin{bmatrix}\boldsymbol{\Phi} & \boldsymbol{j}\end{bmatrix}\boldsymbol{\Phi}'$
  $\boldsymbol{s} \leftarrow \boldsymbol{s}'$
  $\boldsymbol{\Psi} \leftarrow \begin{bmatrix}\boldsymbol{\Psi} & \boldsymbol{0} \\ \boldsymbol{0} & 1\end{bmatrix}\boldsymbol{\Psi}'$
  $k \leftarrow k + 1$
end if
if $\boldsymbol{\Phi}_{:,1}^T\boldsymbol{\Phi}_{:,\mathrm{end}} > \min\{\epsilon_{\mathrm{SVD}}, \epsilon \cdot m_{\Phi}\}$ **then**
  $[\boldsymbol{Q}', \boldsymbol{R}] \leftarrow QR\{\boldsymbol{\Phi}\}$
  $\boldsymbol{\Phi} \leftarrow \boldsymbol{Q}'$
end if

Figure 2. Algorithms in ROM-based topology optimization

2.3 Algorithms

Several algorithms in our ROM-based topology optimization are presented in Figure 2. Algorithm 1 describes the ROM-recycling PCG that is invoked in **linear solve by ROM-recycling Krylov subspace method** of Figure 1. For the preconditioner of the PCG, we use the algebraic multi-grid method [15]. Gram-Schmidt process is described in Algorithm 2 and the incremental SVD is described in Algorithm 4. Either the Gram-Schmidt or the incremental SVD is invoked in **ROM update** of Figure 1 to generate $\boldsymbol{\Phi}$. Note that the incremental SVD specifies the maximum basis size, denoted as m_{Φ}. If the dimension of the reduced basis reaches m_{Φ}, then we invoked Algorithm 3 to rebuild the basis. For the incremental QR, we mimic what is done in [5], i.e., if the dimension of the reduced basis reaches m_{Φ}, then the oldest basis vector is removed from the basis set. Algorithm 4 also introduces the threshold, ϵ_{SVD}, to determine the linear dependency of the new input vector to the existing basis vectors. Various optimization solvers are available for topology optimization. The Method of Moving Asymptotes (MMA) is popular due to its simplicity and practicality [16]. However, the convergence criteria in MMA are heuristic, so it does not guarantee an optimal solution. Thus, we use the interior point algorithm, IPOPT [17], in which the KKT residual norm is used as a convergence criterion.

3. Numerical experiments

We consider two 3D design problems: i) cantilever beam and ii) wind turbine blade. Cantilever beam design is the same problem in [1]. The design domains and optimal designs for two problems are shown in Figure 3. We minimize compliance with mass constraints for both problems. The beam design problem has 24,500 design variables, while the blade problem has 414,979 design variables. For the blade problem, we apply two different loads: 10 N in x-direction and 10 N in y-direction. We set $r_{\mathrm{CG}} = r_{\mathrm{ROM}} = \epsilon_{\mathrm{CG}} = \epsilon_{\max} = 10^{-3}$, $\epsilon_{\mathrm{SVD}} = 10^{-9}$, and $m_{\Phi} = 10$ both for the beam and blade designs.

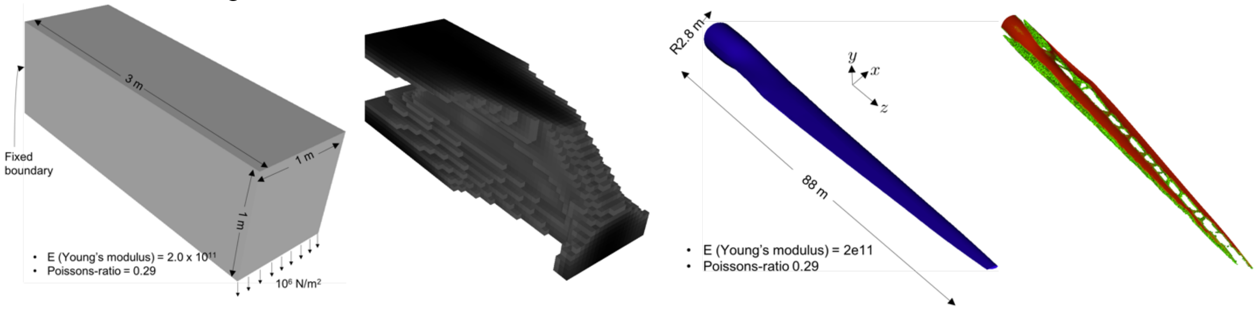

Figure 3. Left Two: design domain and an optimal design for the cantilever beam, Right Two: design domain and an optimal design for the wind turbine blade

Table 1 and 2 show the performance comparison of the default and ROM-based topology optimizations for cantilever beam and blade design, respectively. The default method is to use PCG with AMG preconditioner for all the linear solves with a fixed convergence threshold.

Table 1. Performance comparison for cantilever beam 3D design

	Default	*ROM-based top.opt. incremental QR*	*ROM-based top.opt. incremental SVD*
Optimal compliance	13.59	13.59	13.59
Total CPU time of linear solve (sec.)	**47.1**	**14.4**	**12.1**
Total iters. of linear solve	**10,555**	**396**	**323**
Avg. iters. of linear solve	**34.8**	**2.1**	**2.1**
IPOPT iter.	282	183	151
KKT norm	6.25e-7	6.93e-7	2.13e-7
Speed-up of total linear solve	1.0	**3.3**	**3.9**
Avg. iter. reduction of linear solve	1.0	**16.6**	**16.6**
Total iter. reduction of linear solve	1.0	**26.7**	**32.7**

Table 2. Performance comparison for wind turbine blade 3D design

	Default	*ROM-based topopt incremental QR*	*ROM-based topopt incremental SVD*
Optimal compliance	2.61	2.62	2.61
Total CPU time of linear solve (hour)	**3.1**	**0.9**	**0.7**
Total iters. of linear solve	**706,943**	**154,732**	**123,750**
Avg. iters. of linear solve	**394.9**	**79.6**	**85.0**
IPOPT iter.	781	970	726
KKT norm	6.99e-7	9.29e-7	8.58e-7
Speed-up of total linear solve	1.0	**3.4**	**4.4**
Avg. iter. reduction of linear solve	1.0	**5.0**	**4.6**
Total iter. Reduction of linear solve	1.0	**4.6**	**5.7**

Note that the three different methods produce the same optimal compliance value both for beam and blade designs. However, this is guaranteed in general because the approximation introduced by the ROMs may alter the optimization path and can converge to a different local minimum. Nevertheless, the optimal shape obtained by our method satisfies the KKT optimality condition, implying that the quality of the design is not degraded by the approximation introduced by the ROMs. Our method achieves considerable speed-ups of linear solve, i.e., 3.9 for beam and 4.4 for blade design. Additionally, a considerable iteration reduction in linear solves is accomplished by our method, i.e., 32.7 for beam and 5.7 for blade design. The incremental SVD outperforms the incremental QR in terms of the speed-up of linear solves and the total number of iterations of linear solves. This is because the incremental SVD gives an optimal basis set in the POD.

4. Conclusions

We have introduced the ROM-based topology optimization approach that achieves the reduction of linear solver iterations from four to thirty-two and the overall linear solve CPU time reduction a factor of four for our numerical experiments. Additionally, the method does not suffer from the approximation introduced by the ROM because the accuracy of ROM is carefully monitored and treated appropriately throughout the optimization process, resulting in an optimal design that satisfies the KKT optimality condition. Our ROM-based topology optimization method is easily extendable to problems other than compliance minimization problems, e.g., stress-constraint and nonlinear material design problems. We will apply our method to those other types of problems as well as larger-scale problems than the ones considered here.

Acknowledgments: This work was performed at Lawrence Livermore National Laboratory and was supported by the LDRD program (project 17-SI-005). Lawrence Livermore National Laboratory is operated by Lawrence Livermore National Security, LLC, for the U.S. Department of Energy, National Nuclear Security Administration under Contract DE-AC52-07NA27344 and LLNL-JRNL-xxxxxx.

References

1. Wang S and de Sturler E and Paulino G H. Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. Int. J Numer. Meth. Eng., 2007, 69(12):2441-2468
2. Carlberg K and Forstall V and Tuminaro R. Krylov-subspace recycling via the POD-augmented conjugated-gradient method. SIAM J. Matrix Anal. Appl., 2016, 37(3):1304-1336
3. Nguyen N C and Chen Y. Reduced-basis method for the iterative solution of parameterized symmetric positive-definite linear systems. arXiv preprint arXiv:1804.06363, 2018
4. Amir O and Stolpe M and Sigmund O. Efficient use of iterative solvers in nested topology optimization. Struct. Multidiscip. Optim., 2010, 42(1):55-72
5. Gogu C. Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction. Int. J Numer. Meth. Eng., 2015, 101(4):281-304
6. Yoon G. Structural topology optimization for frequency response problem using model reduction schemes. Comput. Methods Appl. Mech. Eng., 2010, 199(25-28):1744-1763
7. Heinkenschloss M and Vicente L.N. Analysis of inexact trust-region SQP algorithms. SIAM J. Optim., 2002, 12(2):283-302
8. Bellavia S. Inexact interior-point method. J. Optimiz. Theory App., 1998, 96(1):109-121
9. Byrd R H and Curtis F E and Nocedal J. An inexact SQP method for equality constrained optimization. SIAM J. Optim., 2008, 19(1):351-369
10. Choi Y and Carlberg K. Space–time least-squares Petrov–Galerkin projection for nonlinear model reduction. SIAM J. Sci. Comput. 2019, 41(1):A26-A58
11. Amsallem D and Zahr M and Choi Y and Farhat C. Design optimization using hyper-reduced-order models. Struct. Multidiscip. Optim., 2015, 51(4):919-940
12. Berkooz G and Holmes P and Lumley J L. The proper orthogonal decomposition in the analysis of turbulent flows. Annu. Rev. Fluid Mech., 1993, 25(1):539-575
13. Oxberry G.M. and Kostova-Vassilevska T and Arrighi W and Chand K. Limited-memory adaptive snapshot selection for proper orthogonal decomposition. Int. J Numer. Meth. Eng., 2017, 109(2):198-217
14. Parks M L and de Sturler E and Mackey G and Johnson D D and Maiti S. Recycling Krylov subspaces for sequences of linear systems. SIAM J. Sci. Comput., 2006, 28(5):1651-1674
15. Baker A H and Falgout R D and Kolev T V and Yang U M. Scaling Hypre's multigrid solvers to 100,000 cores. High-Performance Scientific Computing, 2010:261-279
16. Svanberg K, Method of moving asymptotes – a new method for structural optimization. Int. J Numer. Meth. Eng., 1998, 41(8):1417-1434
17. Wachter A and Biegler L T. On the implementation of a primal-dual interior point filter line search algorithm for large-scale nonlinear programming. Math. Program., 2006, 106(1):25-57