Structural and Multidisciplinary Optimization (2020) 62:209–230
https://doi.org/10.1007/s00158-019-02485-3

RESEARCH PAPER

On-the-fly model reduction for large-scale structural topology optimization using principal components analysis

Manyu Xiao[1] · Dongcheng Lu[1] · Piotr Breitkopf[2] · Balaji Raghavan[3] · Subhrajit Dutta[4] · Weihong Zhang[5]

Received: 30 April 2019 / Revised: 9 December 2019 / Accepted: 18 December 2019 / Published online: 19 February 2020

Abstract

Despite a solid theoretical foundation and straightforward application to structural design problems, 3D topology optimization still suffers from a prohibitively high computational effort that hinders its widespread use in industrial design. One major contributor to this problem is the cost of solving the finite element equations during each iteration of the optimization loop. To alleviate this cost in large-scale topology optimization, the authors propose a projection-based reduced-order modeling approach using proper orthogonal decomposition for the construction of a reduced basis for the FE solution during the optimization, using a small number of previously obtained and stored solutions. This basis is then adaptively enriched and updated *on-the-fly* according to an error residual, until convergence of the main optimization loop. The method of moving asymptotes is used for the optimization. The techniques are validated using established 3D benchmark problems. The numerical results demonstrate the advantages and the improved performance of our proposed approach.

Keywords POD · Meta-modeling · Design · High-performance computing · Additive manufacturing · 3D printing

1 Introduction

Topology optimization, first introduced by Bendsoe (1989) has matured over the last few decades (Xia and Breitkopf 2014, 2017) and has had a significant influence on design optimization research.

The classical topology optimization problem consists of optimizing material distribution in two or three dimensions so as to minimize the structural compliance, i.e., finding the density distribution over a voxel grid for a chosen volume fraction under a prescribed set of external loads and boundary conditions. Density-based methods are today the most widely used by engineers along with level-set methods (Zhou et al. 2019), topological derivative procedures (Allaire et al. 2004; Norato et al. 2007), and phase field techniques (Xia et al. 2018), etc (Ferro et al. 2019). A comprehensive review of developments in topology optimization post 2000 may be found in Deaton and Grandhi (2014).

With the modern day mastery of additive manufacturing techniques, topology optimization is increasingly being applied in the design of engineered materials for aerospace applications (Meng et al. 2019b). However, it is surprisingly far from attaining mainstream popularity among structural engineers, despite nearly two decades of research that have been devoted to the subject. One of the key challenges in topology optimization has been dealing with large-scale or high-dimensional design problems that could involve millions or even billions of degrees of freedom (Aage et al. 2017). During each iteration of the optimization process, we need to solve the equilibrium equations for the computation–intensive numerical/finite element (FE) model characterizing the discretized structure. This central and still

Responsible Editor: Fred van Keulen

✉ Piotr Breitkopf
piotr.breitkopf@utc.fr

[1] Xi'an Key Laboratory of Scientic Computation and Applied Statistics, Department of Applied Mathematics, School of Mathematics and Statistics, Northwestern Polytechnical University, Xi'an, 710072, People's Republic of China

[2] Alliance Sorbonne Universités, Université de Technologie de Compiègne, Laboratoire Roberval FRE UTC-CNRS 2012, Compiègne, France

[3] Laboratoire de Génie Civil et Génie Mécanique EA 3913, Institut National des Sciences Appliquées de Rennes, Rennes, France

[4] Department of Civil Engineering, National Institute of Technology Silchar, Silchar, India

[5] State IJR Center of Aerospace Design and Additive Manufacturing, Northwestern Polytechnical University, Xi'an, China

unresolved issue of prohibitively high computational effort casts an ever-present pall on its large-scale application to industrial design.

High-performance computing approaches have been proposed in the literature surveyed to deal with this problem and are expectedly successful (Mahdavi et al. 2006; Aage and Lazarov 2013; Aage et al., 2015, 2017), but most, if not all, require an increase in computing resources to realize their full potential in reducing the computational time.

Reanalysis methods have been used in topology optimization since the seminal paper of Kirsch and Papalambros (Kirsch and Papalambros 2001) in 2001, where they proposed a unified approach for structural reanalysis in topology optimization. Wang et al. (2007) and Amir et al. (2009) proposed methods based on the use of Krylov subspaces. In a different paper, Amir et al. (2010) proposed the construction of a reduced basis using the combined approximations method. Reanalysis methods were also used in Amir et al. (2012), He and Jiang (2012), and Kirsch and Bogomolni (2004). Yoon (2010) used eigenmodes and Ritz vectors for the reduced basis in topology optimization for vibration response. Gogu (2015) extended the approach of Kirsch and Papalambros (2001) and used Gram–Schmidt orthonormalization to construct a reduced basis *on-the-fly* based on the violation of an error residual. A survey of the available literature reveals a recent resurgence of interest in reanalysis in topology optimization (Zheng et al. 2017; Sun et al. 2018; Senne et al. 2019).

Reduced-order modeling (ROM), in particular, supervised manifold learning has become a popular approach in a variety of fields today including computational mechanics and structural optimization (Dutta et al. 2018). The basic premise of projection-based ROM (Amsallem et al. 2015) involves mapping the higher dimensional physics onto a lower dimensional space through an appropriate reduced basis calculated using various methods depending on the nature of the problem at hand. While the field is still in its infancy (given the magnitude of potential improvements), the results obtained thus far have been more than promising. Principal components analysis (PCA) (Amsallem et al. 2015; Pearson 1901), proper generalized decomposition (PGD) (Chinesta et al. 2011), hyper-reduction (Ryckelynck et al. 2006), and reduced basis methods (Hoang et al. 2016) are the three prominent schools of this field today. Of these, PCA, also called proper orthogonal decomposition or POD (Berkooz et al. 1993; Xiao et al. 2009; Dulong et al. 2007; Raghavan and Breitkopf 2013; Raghavan et al. 2013a; Raghavan et al. 2013b; Meng et al. 2018; Madra et al. 2018; Xiao et al. 2018; Meng et al. 2019a), is an *a posteriori* statistical method that learns the covariance structure of complex multivariate data.

With the very recent exceptions of Alaimo et al. (2018), Ferro et al. (2019), and Choi et al. (2019), to the knowledge of the authors, virtually no work has been done on coupling topology optimization with POD. The work of Ferro et al. (2019) involves applying POD to the density map and yields a very efficient numerical scheme which loses precision depending on the number of modes. Since their ROM was not computed "on-the-fly," i.e., with constant monitoring using the full-field model, could have resulted in the dependence of their obtained optimized topology density map on basis size. In addition, Choi et al. (2019) presented a novel approach to ROM-supplemented topology optimization using inexact linear solutions by incremental SVD during the initial stages of the optimization (when the accuracy is not expected to be as strict), and Krylov subspace methods with ROM recycling closer to convergence, where greater accuracy is expected.

In this work, inspired by Kirsch and Papalambros (2001) and Gogu (2015), we improve the computational efficiency by mapping displacement field quantities of the large-scale problem to a low-dimensional space through an appropriate basis, which we calculate using POD. To render the method more accessible on a workstation, we use an iterative solver for the full-field solution. The method of moving asymptotes (MMA) is used for the optimization as an alternative to the classical optimality criteria (OC) method, based on a dedicated version of sensitivity analysis.

The remainder of the paper is organized in the following manner: in Section 2, the theoretical formulation is formally presented beginning with classical topology optimization, followed by the reduced-order basis construction and sensitivity analysis. In Section 3, we summarize the algorithm for *on-the-fly* basis construction using POD. Section 4 details the numerical investigations using benchmark 3D compliance minimization problems followed by a discussion. The paper closes with concluding comments and recommendations for future work.

Extension to non self-adjoint problems is discussed in the Appendix.

2 Theoretical formulation

The mathematical formulation of the discrete material distribution problem may be expressed as follows:

$$\begin{aligned}
&\min_{\rho} c(\rho) = F^T U = U^T \mathrm{KU}\\
&\qquad \sum_{e=1}^{N} v_e \rho_e = v_{\mathrm{frac}} V < V\\
&0 \leq \rho_e \leq 1, \quad e = 1, \ldots, N\\
&\qquad\qquad \mathrm{KU} = F
\end{aligned} \tag{1}$$

where c is the compliance of the structure, ρ is the vector of design variables consisting of the individual element (e) densities ρ_e, F is the external forces vector, U is the FE displacement vector, K is the global stiffness matrix of the structure, v_e the volume of an element e, and V the maximum prescribed volume for the entire structure. The number of elements in the 2D/3D grid is N.

Using a modified solid isotropic material with penalization model Amir et al. (2010), the density of an element can be expressed as follows:

$$E_e(\rho_e) = E_{\min} + \rho_e^p (E_{\text{nominal}} - E_{\min}) \tag{2}$$

For topology optimization of large-scale structures, the bulk of the computational cost expectedly stems from the requirement to compute the numerical solution of the equilibrium equations at each iteration:

$$\text{KU} = F \tag{3}$$

Computing this full-field solution for large-scale topology optimization problems involves the inversion of a very large system of equations that can consist of up to millions or billions (Aage et al. 2017) of degrees of freedom. To improve the scalability of the approach to allow for implementation on parallel computing systems eventually (not treated in this particular paper), the FEA for the full-field solution is performed using a preconditioned conjugate solver for improved scalability, similar to Mahdavi et al. (2006) except using an incomplete Cholesky decomposition of K as the preconditioner.

The authors must point out that the PCG with incomplete Cholesky is no longer the state of the art solver, and computation times using multi-grid preconditioning (Tatebe 1993), the current gold standard according to the literature, may well be different from those listed in this work.

Algorithm 1 Solution of $KU = F$ using iterative solver (PCG).

```
procedure PCG-FEA
    $L \longleftarrow$ *Incomplete* Cholesky decomposition of $K$
    Preconditioner $\tilde{K} = LL^T$
    $R_0 = F - KU_0$
    $Z_0 = \tilde{K}^{-1} R_0$
    $P_0 = Z_0$
    $i = 0$
    while $||R_0|| > tol$ do
        $\alpha_i = \frac{R_i^T Z_i}{P_i^T K P_i}$
        $U_{i+1} = U_i + \alpha_i P_i$
        $R_{i+1} = R_i - \alpha_i K P_i$
        $Z_{i+1} = \tilde{K}^{-1} R_{i+1}$
        $\beta_i = \frac{Z_{i+1}^T R_{i+1}}{Z_i^T R_i}$
        $P_{i+1} = Z_{i+1} + \beta_i P_i$
        $i = i + 1$
```

The basic operations are given in Algorithm 1, which is a standard procedure that may be found in any textbook on numerical methods. However, the iterative solution is still computationally expensive since it involves a large number of degrees of freedom, but also because of the preconditioning phase due to the poorly conditioned matrix K (large variations between nearly void $E_{\min}$ and solid $E_{\max}$). To alleviate this issue, we propose a ROM procedure in the following subsections.

2.1 Projection-based reduced-order modeling

To reduce the computational effort during an iteration of the optimizer loop, we map the displacement field quantity (i.e., U) of the above large-scale problem (3) to a low-dimensional space through an appropriate *orthonormal* basis Φ (i.e., $\Phi^T \Phi = \boldsymbol{I}$) calculated *on-the-fly* using solution snapshots from the previous iterations.

The basis $\Phi = [\phi_1 \dots \phi_{N_b}]$ is obtained using an effective set of N_b "snapshots" of the displacement field $\boldsymbol{U}_{\text{temp}} = [U_1, U_2, \dots, U_{N_b}]$ each obtained by solving (3) *during* the main optimization, centered around the mean snapshot $\bar{u} = (\sum_{k=1}^{N_b} U_k)/N_b$. (Later on, we will show that Φ may be calculated by singular value decomposition (svd) of $\boldsymbol{U}_{\text{temp}}$).

The problem projected onto the reduced basis transforms into the reduced system:

$$\Phi^T \text{KU}_{\text{rb}} = \Phi^T F \tag{4}$$

where U_{rb} is the approximate solution to the higher dimensional displacement vector, obtained by a linear combination of the projection coefficients (α):

$$U_{\text{rb}} = \Phi\alpha + \bar{u} \tag{5}$$

Equation (4) thus becomes:

$$\Phi^T K(\Phi\alpha + \bar{u}) = \Phi^T F \tag{6}$$

The main consequence is that any of the displacement vector snapshots U_i may be expressed as a finite basis linear combination:

$$U_i \approx U_{\text{rb}}^i = \bar{u} + \sum_{k=1}^{m} \alpha_k^i \phi_k = \bar{u} + \Phi\alpha^i \tag{7}$$

where the α^i depend on the choice of the basis Φ. The error residual is given by the following:

$$\epsilon_{\text{rb}}^2 = \frac{\| \text{KU}_{\text{rb}} - F \|^2}{\| F \|^2} = \frac{\| K(\Phi\alpha + \bar{u}) - F \|^2}{\| F \|^2} \tag{8}$$

corresponding to the relative error between the internal forces stemming from the approximate reduced basis solution and the actually applied forces. If the approximate solution U_{rb} were exact, the residual would be zero because the exact solution would satisfy the equilibrium equations $\text{KU} = F$.

The goal then is to use U_{rb} in place of U for the optimization depending on the error threshold ϵ_{rb}. If this error is unreasonable, we then run the full-field FE, i.e., Equation (3) at that particular loop iteration to get a fresh displacement vector that will then be used to refine the basis. Note that in order to retain generality as far as possible, we will hold off on presenting the exact method of calculating the basis until the end of this section, the reason being that much of this section is relevant regardless of the choice of Φ. The exact basis updation scheme is described in the next subsection

2.2 Sensitivity analysis

When the ROM, i.e., U_{rb} is used in place of the FE solution, the original objective function (compliance) may be expressed as follows:

$$c(\rho_e) = U_{\mathrm{rb}}^T K(\rho_e) U_{\mathrm{rb}} = (\Phi\alpha + \bar{u})^T K(\rho_e)(\Phi\alpha + \bar{u}) \quad (9)$$

The use of this expression, however, entails the verification of some additional constraints. The first constraint represents the Galerkin *projected*, i.e., reduced system of equations (replacing the original FE):

$$K_{rb} U_{\mathrm{rb}} = F_{\mathrm{rb}} \text{ or}$$

$$\Phi^T \mathrm{K} U_{\mathrm{rb}} = \Phi^T K(\Phi\alpha + \bar{u}) = \Phi^T F \quad (10)$$

The second constraint must be on the snapshots $U_1 \dots U_{N_b}$ used for generating the orthogonal basis vectors, having each (by definition) been obtained through the solution of the full equilibrium equation during *the particular iteration* that they were added to the set of snapshots:

$$K_i U_i = F \text{ where } i = 1, 2 \dots N_b \quad (11)$$

where K_i is simply the stiffness matrix for which the snapshot vector U_i was obtained. In the completely general case, the sensitivity of the compliance calculated using the ROM is potentially different from the sensitivity for the original problem.

Following Kirsch and Papalambros (2001) and Gogu (2015), the conventional way to calculate the modified sensitivity is by using the adjoint equation, using Lagrange multipliers $\mu_i, \lambda_i, i = 1 \dots N_b$ for the two constraints in (10) and (11).

The modified objective function may then be represented as follows:

$$c(\rho_e) = (\Phi\alpha + \bar{u})^T K(\Phi\alpha + \bar{u}) - 2\mu^T [\Phi^T K(\Phi\alpha + \bar{u}) - \Phi^T F] - \sum_{i=1}^{N_b} \lambda_i^T (K_i U_i - F) \quad (12)$$

This expression may be simplified as follows:

$$c(\rho_e) = [\alpha^T \Phi^T K \Phi\alpha - 2\mu^T (\Phi^T K \Phi\alpha - \Phi^T F)] + [2\bar{u}^T K \Phi\alpha + \bar{u}^T K \bar{u} - 2\mu^T \Phi^T K \bar{u}] - [\sum_{i=1}^{N} \lambda_i^T (K_i U_i - F)] = c_1(\rho_e) + c_2(\rho_e) + c_3(\rho_e) \quad (13)$$

where c_1, c_2, and c_3 are the terms within the square brackets.

Each of the three terms may then be individually evaluated as follows:

$$\frac{\partial c_1}{\partial \rho_e} = 2(\alpha - 2\mu)^T \frac{\partial \Phi^T}{\partial \rho_e} K \Phi\alpha + 2(\alpha - \mu)^T \Phi^T K \Phi \frac{\partial \alpha}{\partial \rho_e} + (\alpha - 2\mu)^T \Phi^T \frac{\partial K}{\partial \rho_e} \Phi\alpha + 2\mu^T \frac{\partial \Phi^T}{\partial \rho_e} F \quad (14)$$

$$\begin{aligned}\frac{\partial c_2}{\partial \rho_e} &= 2(\alpha - \mu)^T \Phi^T K \frac{\partial \bar{u}}{\partial \rho_e} + 2(\alpha - \mu)^T \Phi^T \frac{\partial K}{\partial \rho_e} \bar{u} + \bar{u}^T \frac{\partial K}{\partial \rho_e} \bar{u} \\ &\quad 2(\alpha - \mu)^T \frac{\partial \Phi^T}{\partial \rho_e} K \bar{u} + 2 \frac{\partial \alpha^T}{\partial \rho_e} \Phi^T K \bar{u} + 2 \frac{\partial \bar{u}^T}{\partial \rho_e} K \bar{u} \\ &= 2[(\alpha - \mu)^T \Phi^T + \bar{u}] K \frac{\partial \bar{u}}{\partial \rho_e} + [2(\alpha - \mu)^T \Phi^T + \bar{u}] \frac{\partial K}{\partial \rho_e} \bar{u} \\ &\quad + 2 \frac{\partial \alpha^T}{\partial \rho_e} \Phi^T K \bar{u} + 2(\alpha - \mu)^T \frac{\partial \Phi^T}{\partial \rho_e} K \bar{u}\end{aligned} \quad (15)$$

and the last term:

$$\frac{\partial c_3}{\partial \rho_e} = -\sum_{i=1}^{N_b} \lambda_i^T \frac{\partial K_i}{\partial \rho_e} U_i - \sum_{i=1}^{N_b} \lambda_i^T K_i \frac{\partial U_i}{\partial \rho_e} \quad (16)$$

In order to solve the adjoint equation, we remember that we are free to choose the Lagrange multipliers as we see fit. A useful substitution is $\mu = (\alpha + \Phi^T \bar{u})$ giving the following:

$$\frac{\partial c_1}{\partial \rho_e} = -2(\alpha + 2\Phi^T \bar{u})^T \frac{\partial \Phi^T}{\partial \rho_e} K \Phi\alpha - 2\bar{u}^T K \Phi \frac{\partial \alpha}{\partial \rho_e} - (\alpha + 2\Phi^T \bar{u})^T \Phi^T \frac{\partial K}{\partial \rho_e} \Phi\alpha + 2(\alpha + \Phi^T \bar{u})^T \frac{\partial \Phi^T}{\partial \rho_e} F \quad (17)$$

and

$$\frac{\partial c_2}{\partial \rho_e} = 2 \frac{\partial \alpha^T}{\partial \rho_e} \Phi^T K \bar{u} - 2\bar{u}^T \Phi \frac{\partial \Phi^T}{\partial \rho_e} K \bar{u} - \bar{u}^T \frac{\partial K}{\partial \rho_e} \bar{u} \quad (18)$$

From the above, we end up with the following:

$$\begin{aligned}\frac{\partial c}{\partial \rho_e} &= -2(\alpha + 2\Phi^T \bar{u})^T \frac{\partial \Phi^T}{\partial \rho_e} K \Phi\alpha - (\alpha + 2\Phi^T \bar{u})^T \Phi^T \frac{\partial K}{\partial \rho_e} \Phi\alpha \\ &\quad + 2(\alpha + \Phi^T \bar{u})^T \frac{\partial \Phi^T}{\partial \rho_e} F - 2\bar{u}^T \Phi \frac{\partial \Phi^T}{\partial \rho_e} K \bar{u} \\ &\quad - \bar{u}^T \frac{\partial K}{\partial \rho_e} \bar{u} - \left[\sum_{i=1}^{N_b} \lambda_i^T \frac{\partial K_i}{\partial \rho_e} U_i + \sum_{i=1}^{N_b} \lambda_i^T K_i \frac{\partial U_i}{\partial \rho_e}\right]\end{aligned} \quad (19)$$

which may further be simplified to the following:

$$\begin{aligned}\frac{\partial c}{\partial \rho_e} &= -U_{\mathrm{rb}}^T \frac{\partial K}{\partial \rho_e} U_{\mathrm{rb}} + 2U_{rb}^T \Phi \frac{\partial \Phi^T}{\partial \rho_e} (F - \mathrm{K} U_{\mathrm{rb}}) \\ &\quad - \left[\sum_{i=1}^{N_b} \lambda_i^T \frac{\partial K_i}{\partial \rho_e} U_i + \sum_{i=1}^{N_b} \lambda_i^T K_i \frac{\partial U_i}{\partial \rho_e}\right] \\ &= -U_{\mathrm{rb}}^T \frac{\partial K}{\partial \rho_e} U_{\mathrm{rb}} + 2U_{\mathrm{rb}}^T \Phi \frac{\partial \Phi^T}{\partial \rho_e} \Delta F - \left[\sum_{i=1}^{N_b} \lambda_i^T \frac{\partial K_i}{\partial \rho_e} U_i + \sum_{i=1}^{N_b} \lambda_i^T K_i \frac{\partial U_i}{\partial \rho_e}\right]\end{aligned} \quad (20)$$

The above equation is a generalized version of the expression obtained by Gogu (2015), in the context of an orthonormal basis Φ and including the effect of the mean snapshot $\bar{u}$, and is valid for any reduced approach in the Galerkin family. (Note that if the mean $\bar{u}$ were assumed to be = 0 (centered snapshots), the second set of terms within parentheses would vanish yielding the same exact expression as in Gogu 2015).

To go further and obtain a final expression, we present the updation strategy in the next subsection.

2.3 On-the-fly reduced basis construction and updation strategy

In the last equation of the previous subsection, we still need to determine $\lambda_1 .. \lambda_{N_b}$ and $\frac{\partial \Phi}{\partial \rho_e}$ so as to obtain $\frac{\partial c}{\partial \rho_e}$, and these will depend on the particular updation strategy, which is explained in detail in this subsection.

After $i \geq N_b$ iterations of a classical topology optimization procedure, we expect to have already calculated N_b displacement vectors ($U_1 \ldots U_{N_b}$) by the usual process of inverting the full equilibrium equations in (3). As hinted earlier, the subspace generated by these N_b previously calculated vectors can be used to calculate a reduced basis Φ that could be used to estimate the displacement vector for the next iteration $(i+1)$.

This means that the corresponding (approximate) displacement vector is obtained using the ROM in (5), which calculates the reduced state variables at the current iteration $(i + 1)$ (and, thus, an approximation of U) by solving the equilibrium equations projected on the subspace generated by the N_b displacement vector snapshots.

At iteration $(i + 2)$, a new approximation of the displacement vector can still be calculated using the ROM with the same subspace generated by the first N_b displacement vectors. This process may be applied until the approximate solution using the ROM is no longer sufficiently accurate, based, for example, on a threshold on the value of the residual ϵ_{rb} in (8), at which point, we use (3) to get a *fresh* snapshot vector to replace the *oldest* stored vector, and thus refine the basis Φ.

Algorithm 2 Topology optimization with *on-the-fly* ROM construction.

```
procedure TopOpt-ROM
    System Initialization (volume fraction, filter, etc)
    Pre FEM operations (mesh parameters, load definitions, etc)
    while $iteration \leq N_b$ do
        $\boldsymbol{U}_{temp} \longleftarrow$ solution of $KU = F$
        if $iteration = N_b$ then
            $\Phi \longleftarrow$ calculated from $\boldsymbol{U}_{temp}$
        else
            $U_{rb} \longleftarrow= $ solution of $\Phi^T K U_{rb} = \Phi^T F$
            $\epsilon_{rb} \longleftarrow \frac{||KU_{rb} - F||}{||F||}$
            if $\epsilon_{rb} > \varepsilon$ then
                Remove oldest snapshot from $\boldsymbol{U}_{temp}$
                $\boldsymbol{U}_{temp} \longleftarrow$ solution of $KU = F$
                $\Phi \longleftarrow$ calculated from $\boldsymbol{U}_{temp}$
            else
                $U \longleftarrow U_{rb}$
        Calculate sensitivities ($\frac{\partial c}{\partial \rho_e}$)
        Apply density filtering to calculated sensitivities
        update element densities ($\rho_e$).
        $iteration = iteration + 1$
```

So whenever the ROM is used, we have N_b basis vectors that are only updated as and when the residual exceeds our pre-specified tolerance, by re-running (3) and replacing the *oldest* snapshot vector.[1] When the residual is below the tolerance, we use U_{rb} instead.

This means that we do not use a *continuously evolving* basis Φ in this work past the *first* N_b iterations (that are used to determine the initial basis), rather our basis is only updated using a fresh FE solution to modify $\boldsymbol{U}_{\text{temp}}$ when the error residual ϵ_{rb} in (8) is unacceptably high. If the residual is within the tolerance, we reuse the existing Φ.

The basic approach is given in Algorithm 2.

In addition, when the reduced basis Φ is used to get U_{rb}, $K_1 \ldots K_{N_b}$ and $U_1 \ldots U_{N_b}$ are not *continuous* functions of the *current* density ρ_e (having been previously obtained during the basis-changing iterations). This in turn applies to the basis Φ (obtained from the snapshots U_i). So most of the terms in the previously obtained expression will vanish.

This ultimately means that we recover the classical expression for the sensitivity for our particular approach, i.e.,

$$\frac{\partial c}{\partial \rho_e} = -U_{rb}^T \frac{\partial K}{\partial \rho_e} U_{\text{rb}} \tag{21}$$

In the next subsection, we complete this section by describing the procedure of constructing Φ from the FE solutions $U_1 \ldots U_{N_b}$ using PCA.

2.4 Construction of ROM (Φ and U_{rb}) using PCA

As explained earlier, we map the displacement field quantity of the above large-scale problem (i.e., U) to a low-dimensional space through an appropriate *orthonormal* basis Φ. The higher dimensional data may then be reconstructed by linear combination of the projection coefficients α using (5), thus leading to the reconstruction error in (8). The PCA approach in this paper uses singular value decomposition to calculate Φ using the matrix of the M displacement vector snapshots to minimize this reconstruction error.

The basic idea behind "economical" singular value decomposition (SVD) of a real matrix $\mathbf{D}_{N \times M}$ where $N > M$ is expressing it as under:

$$\mathbf{D} = \boldsymbol{\Psi} \boldsymbol{\Sigma} \boldsymbol{V}^T \tag{22}$$

where $\boldsymbol{\Psi}_{N \times M}$ and $\boldsymbol{V}_{M \times M}$ are both *unitary/orthogonal* matrices and $\boldsymbol{\Sigma}_{M \times M}$ is a diagonal matrix (i.e., $\boldsymbol{\Sigma}_{\text{ij}} = \delta_{\text{ij}}$). It can be easily shown that $\boldsymbol{\Psi}$ is the matrix of eigenvectors of the square covariance matrix $\boldsymbol{C^v} = \mathbf{D}\mathbf{D}^T$ while the elements along the "diagonal" of $\boldsymbol{\Sigma}$ squared are its eigenvalues.

[1] Refining the basis by discarding the older less relevant information in favor of more recent information is a fairly standard strategy, also used by Gogu (2015)

Constructing the *centered* snapshot matrix $\boldsymbol{D}$ using M stored FE solutions centered around the mean snapshot $\bar{u}$:

$$\boldsymbol{D} = [U_1 - \bar{u} \dots U_M - \bar{u}] \tag{23}$$

gives the reduced basis Φ composed of the first N_b columns of $\boldsymbol{\Psi}$, where the number of modes N_b is selected according to the energy criterion:

$$\epsilon_{\text{PCA}} = 1 - \frac{\sum_{i=1}^{N_b} s_i}{\sum_{j=1}^{M} s_j} \tag{24}$$

Note here that since the actual calculation process here involves a relatively small N_b (total number of snapshots) in the first place, compared to the number of degrees of freedom in the full-field model, we can use all the N_b modes without truncation, i.e., $N_b = M$.

Algorithm 2 is then completed with details about the construction of Φ, and therefore U_{rb} as shown below in algorithm 3.

Algorithm 3 Topology optimization with iterative (PCG) solver and *on-the-fly* reduced basis construction using PCA.

```
procedure TopOpt-PCA
    System Initialization (volume fraction, filter, etc)
    Pre FEM operations (mesh parameters, load definitions, etc)
    while $iteration \leq N_b$ do
        $\boldsymbol{U}_{temp} \longleftarrow$ solution of $KU = F$
        if $iteration = N_b$ then
            $\bar{u} \longleftarrow$ mean $(\boldsymbol{U}_{temp})$
            $\Phi \longleftarrow svd(\boldsymbol{U}_{temp} - \bar{u})$
        else
            $\alpha \longleftarrow=$ solution of $\Phi^T K \Phi \alpha = \Phi^T F - \Phi^T K \bar{u}$
            $\epsilon_{rb} \longleftarrow \frac{||K(\Phi\alpha+\bar{u})-F||}{||F||}$
            if $\epsilon_{rb} > \varepsilon$ then
                Remove oldest snapshot from $\boldsymbol{U}_{temp}$
                $\boldsymbol{U}_{temp} \longleftarrow$ solution of $KU = F$
                $\bar{u} \longleftarrow$ mean $\boldsymbol{U}_{temp}$
                $\Phi \longleftarrow SVD(\boldsymbol{U}_{temp} - \bar{U})$
            else
                $U_{rb} \longleftarrow \Phi\alpha + \bar{u}$
        Calculate sensitivities $(\frac{\partial c}{\partial \rho_e})$
        Apply filtering to calculated sensitivities
        update element densities $(\rho_e)$.
        $iteration = iteration + 1$
```

3 Benchmark tests

To demonstrate the effectiveness of the approach presented in this paper, we first compare the PCA-based approach with an ROM based on Gram–Schmidt orthonormalization (Gogu 2015) for a 2D benchmark compliance minimization problem. Next, we use two benchmark 3D tests and minimize the structural compliance with the classical SIMP (single isotropic material with penalization) assumption. The elastic parameters: maximum and minimum (dimensionless) Young's moduli $E_{\text{nominal}} = 1$ and $E_{\min} = 10^{-9}$, Poisson's ratio $\nu = 0.3$. The penalty factor $p = 3$ and a density filter radius of 1.5 has been applied in both cases.

As an alternative to the frequently used optimality criteria approach (Saxena and Ananthasuresh 2000; Yin and Yang 2001; Sigmund 2001), we have used the method of moving asymptotes (Svanberg 1987, 2002) for the optimization loop in this work. This method is based on a convex representation of the objective function and is conveniently adapted to the problem of topology optimization due to its ease of use. The method has already been demonstrated to work very well on a vast variety of topology optimization problems (Bendsoe and Sigmund 2004; Aage and Lazarov 2013), and lends itself to increased scalability due to the separable nature of the convex approximation.

3.1 2D case: ROM comparison between Gram–Schmidt and PCA

As has been mentioned in the introductory section, a ROM approach for topology optimization using Gram–Schmidt orthonormalization was proposed in Gogu (2015). To compare our proposed approach, i.e., PCA-based on-the-fly reduced-order model, we use the same classical benchmark 2D Messerschmitt–Bolkow–Blohm (MBB) problem (Fig. 1) to assess computational effort, time and accuracy.

The problem parameters have been set as follows: 150×50 and 600×200 (voxel) FE mesh/grid, nominal, and minimum (dimensionless) Young's moduli $\text{E}_{\text{nominal}} = 1$ and $\text{E}_{\min} = 10^{-9}$, Poisson's ratio = 0.3, a maximum allowable volume fraction ν_f of 0.5, a penalization factor $p = 3$, and a density filter radius of 1.5, with the optimization iterations stopped when the density variation within any of the elements is less than 1%.

In order to ensure the convergence of each result of every test, we may set a larger value for the maximum number of iterations: here we set 6000, just to be on the safer side. For both ROM, the number of PCA modes N_b

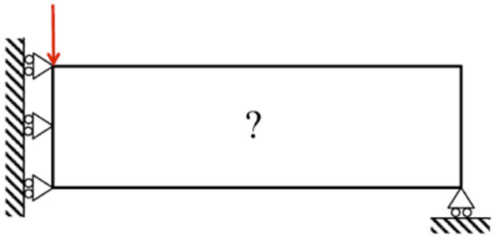

Fig. 1 2D Messerschmitt–Bolkow–Blohm (MBB) benchmark problem

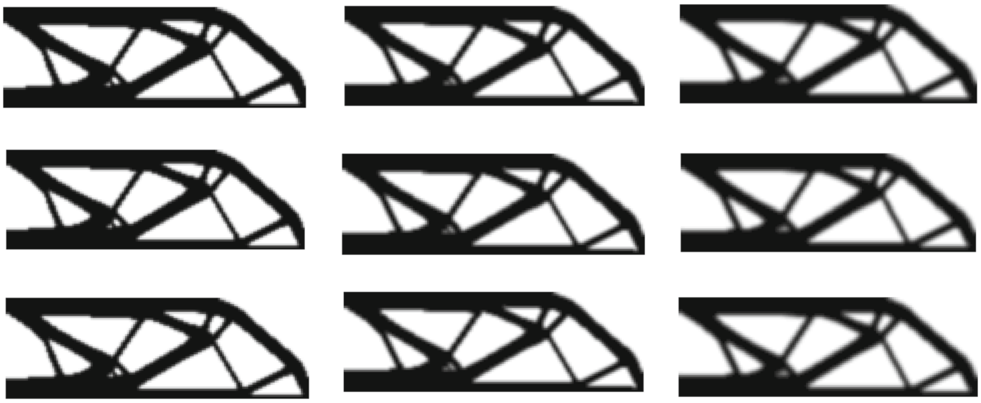

Fig. 2 Optimal topologies generated using the Gram–Schmidt with $r_{\min}$ = (a) 1.5, (b) 2.0, (c) 2.5, PCA with $r_{\min}$ = (d) 1.5, (e) 2.0, (f) 2.5 and reference routine with $r_{\min}$ = (g) 1.5, (h) 2.0, (i) 2.5 for 150 × 50 2D grid

is selected as 4, residual threshold ϵ_{rb} is selected as 0.01. All these parameters are fixed, allowing us to change the filter size $r_{\min}$ on both convergence speed and accuracy of the objective function. The optimized topology and corresponding computing results are summarized in the following discussion.

Figure 2 gives the optimal topologies obtained using the reference routine (i.e., without any ROM), the PCA-based ROM as well as the Gram–Schmidt approach, on a FE mesh of grid resolution 150 × 50 (resolution given in voxels). From the figure, we can see that the three topologies are visually indistinguishable, which means both ROMs yield almost identical design results to that obtained with the reference full order model in the 2D case.

The corresponding results are summarized in Table 1 and Fig. 3.

One can see from Table 1 that various minuscule features (like a tiny hole that appears in the "optimal" topology) fade away before our naked eyes with a slight increase of filter size from 1.5 to 3 for each computation method in each column. However, the boundary of optimal topology for all models gets smoother but fuzzier as we increase the filter size, which may lead to the illusion of the hole getting smaller or even disappearing. We may also draw a conclusion from the table that less optimization time is needed if we use a larger value of filter size (within the adequate range) for any method (reference, PCA and Gram–Schmidt), but larger values of filter size lead to a poorer optimal compliance. It is noteworthy that when using filter size $r_{\min} = 3$, the performance shows a downtick which indicates us there is an optimal filter size.

Moreover, by comparing the PCA approach and Gram–Schmidt routines, we find that the PCA method requires less optimization time and a remarkably fewer number of full solutions (but more iterations) than the Gram–Schmidt method for the same filter size. This validates the PCA ROM as more efficient than the Gram–Schmidt at each iteration step. As far as accuracy of the final objective function is concerned, PCA and Gram–Schmidt methods are basically similar. If we investigate in detail, the former has a slightly higher precision than the latter. To explain the advantage of the PCA approach over Gram–Schmidt

Table 1 Comparison of performance for 150 × 50 2D grid resolution

Method	Filter $r_{\min}$	itrns	Relative error (c)	Density variation	Full FE solutions	Optimal compliance
Reference	1.5	412	0	0.010	412	198.0312
	2.0	391	0	0.010	391	200.8855
	2.5	224	0	0.009	224	208.2528
	3.0	322	0	0.010	322	212.4484
PCA	1.5	408	0.00191	0.009	175	198.0274
	2.0	394	0.0004	0.007	162	200.8847
	2.5	227	0.00149	0.009	89	208.2497
	3.0	395	0.01483	0.010	101	212.4169
Gram–Schmidt	1.5	402	0	0.009	265	198.0312
	2.0	388	0.000060	0.010	246	200.8867
	2.5	224	0.00173	0.008	131	208.2492
	3.0	643	0.04749	0.006	190	212.3475

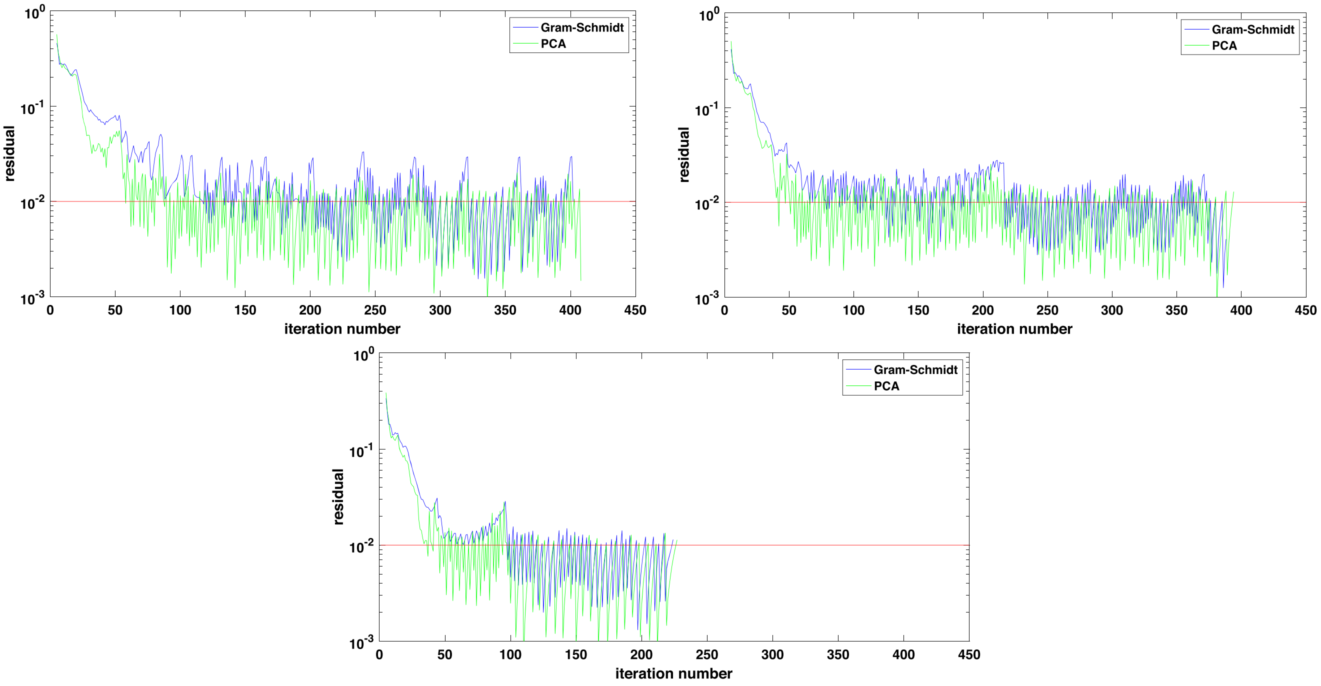

Fig. 3 Residual comparison between Gram–Schmidt and PCA with (a) $\epsilon_{\text{rb}} = 0.1$, $N_b = 4$ and (b) $\epsilon_{\text{rb}} = 0.1$, $N_b = 10$ (c) $N_b = 40$ $\epsilon_{\text{rb}} = 0.1$, (d) $N_b = 4$ $\epsilon_{\text{rb}} = 0.05$, and (e) $\epsilon_{\text{rb}} = 0.05$ $N_b = 10$ (f) $\epsilon_{\text{rb}} = 0.01$ $N_b = 4$

in accuracy, it is instructive to analyze the evolution of the residual throughout the whole iteration process. From Fig. 3, we can very clearly see that PCA method has a clearly lower residual than the Gram–Schmidt method when solving for intermediate displacement vectors during the entire optimization process.

Under the same control precision of design density (1%, here), PCA approach always converges earlier and has a higher convergence accuracy compared to the Gram–Schmidt method for a given $r_{\min}$, a clear improvement in both efficiency and accuracy in this 2D case. We may therefore conclude that the PCA method outperforms the Gram–Schmidt method, at least for this particular 2D benchmark problem.

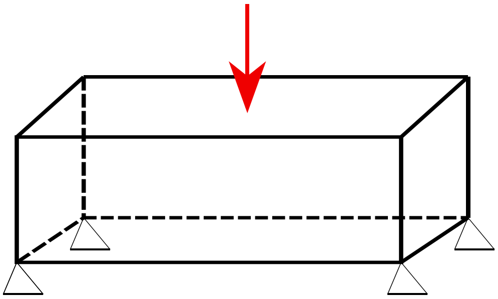

Fig. 4 First 3D test case and boundary conditions

It is important to note that none of this is counter-intuitive, since the Gram–Schmidt is basically an approximation to the POD with the modes directly obtained from the snapshots by orthonormalization rather than going through the procedure of finding the optimal modes through SVD.

3.2 3D case 1: simply supported beam

This test case is a 3D variant of the MBB benchmark problem (Fig. 4)—a simply supported beam under flexion in 3D.

The optimization iterations have been stopped when the density variation within any of the elements is less than 1% (or when 100 iterations have been completed).

We focus on the influence of varying the ROM error threshold ϵ_{rb} and the number of snapshots N_b used to construct the basis Φ, as well as the scalability of the approach with grid resolution.

3.2.1 Scalability of performance with grid resolution

Four different grids were considered here in increasing order of resolution: a coarse 96×24×64 grid, a finer 108×27×72 grid, an even finer grid (132×33×88), and a high-resolution 156×39×104. The 3D topology results are shown in Fig. 5a–d.

Figure 6a and b compare the traditional (without ROM) topology optimization performance with the PCA-

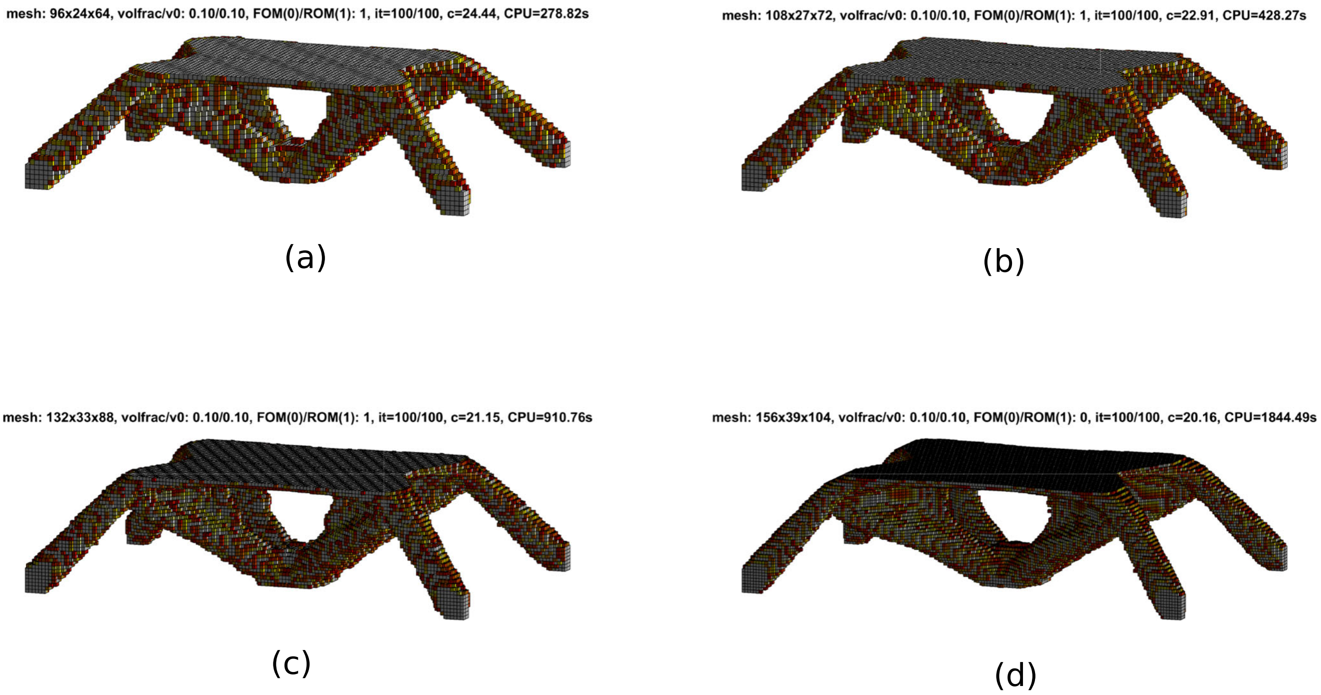
mesh: 96x24x64, volfrac/v0: 0.10/0.10, FOM(0)/ROM(1): 1, it=100/100, c=24.44, CPU=278.82s
(a)
mesh: 108x27x72, volfrac/v0: 0.10/0.10, FOM(0)/ROM(1): 1, it=100/100, c=22.91, CPU=428.27s
(b)
mesh: 132x33x88, volfrac/v0: 0.10/0.10, FOM(0)/ROM(1): 1, it=100/100, c=21.15, CPU=910.76s
(c)
mesh: 156x39x104, volfrac/v0: 0.10/0.10, FOM(0)/ROM(1): 0, it=100/100, c=20.16, CPU=1844.49s
(d)

Fig. 5 Optimized 3D topologies for the MBB beam problem, using four different grids with increasing resolution (**a**) 96×24×64 grid, **b** finer 108×27×72 grid) (**c**) finer grid (132×33×88) and **d** 156×39×104 obtained using PCA

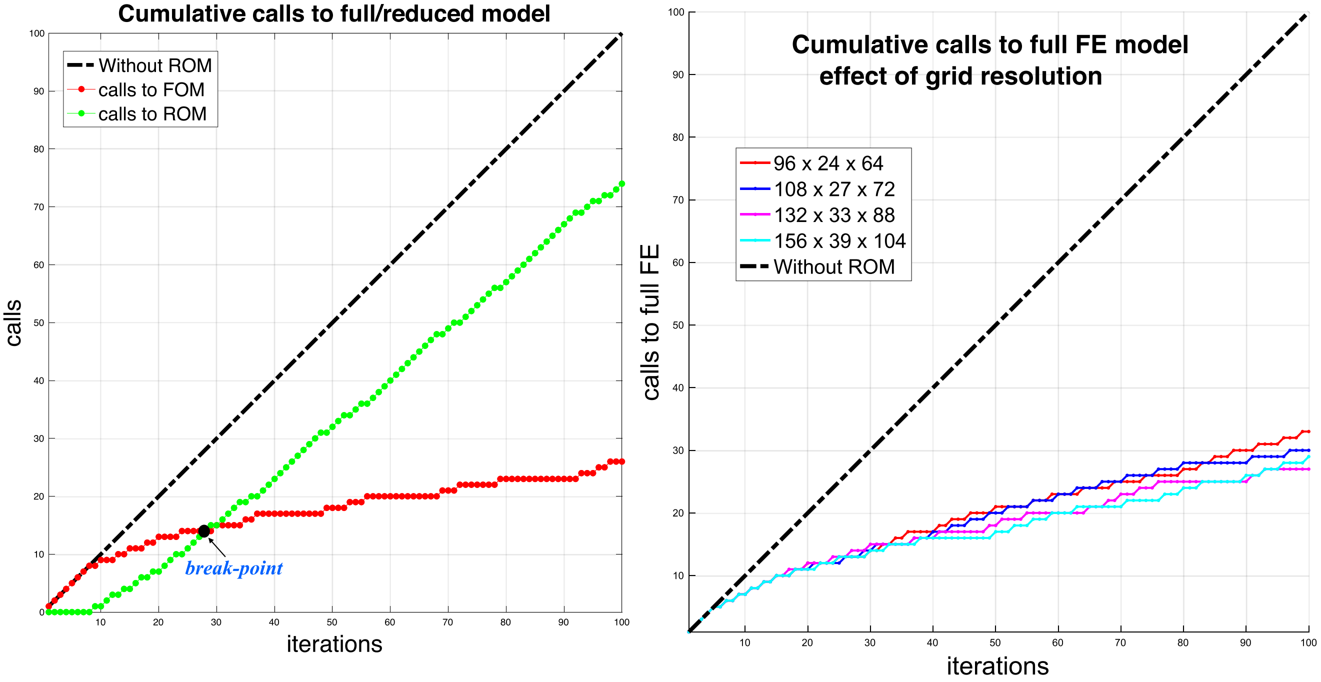

Fig. 6 **a** Typical 132×33×88 grid comparison of computational effort between traditional and PCA approach (**b**) Scalability of ROM performance with grid size using four different grid resolutions - comparison of computational effort with and without ROM

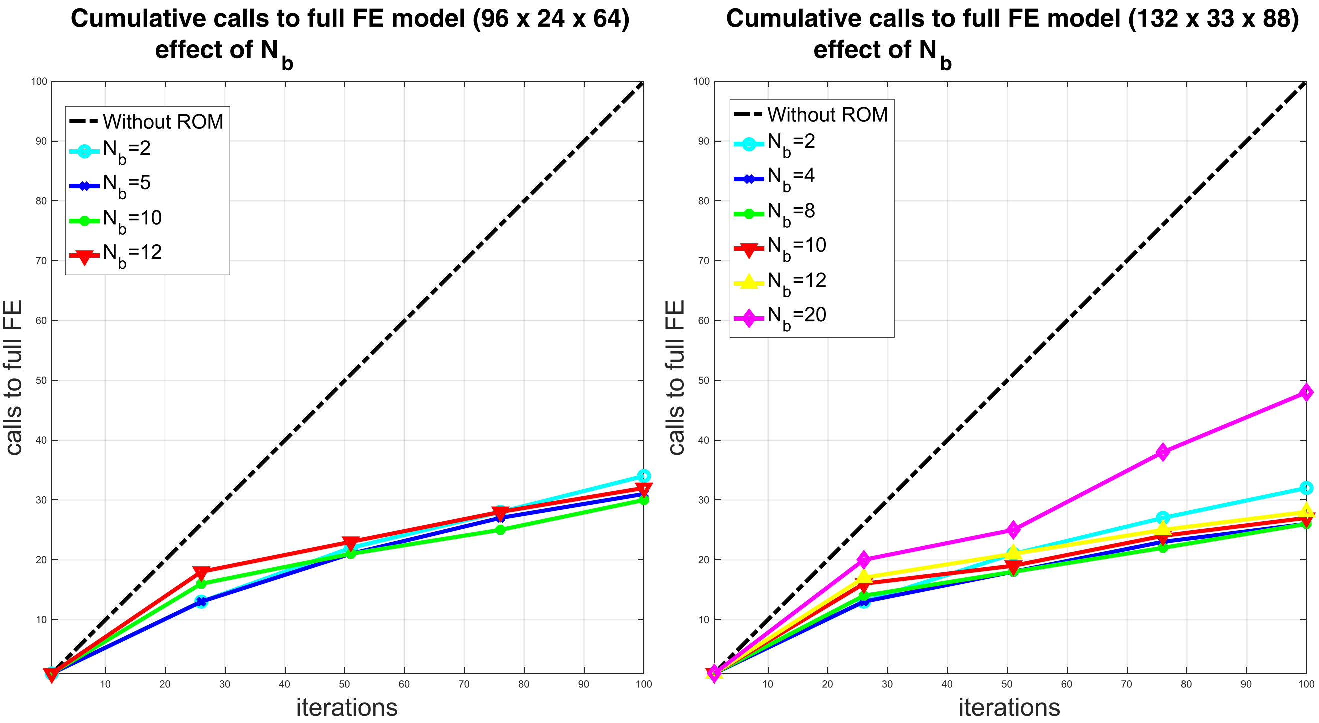

Fig. 7 96×24×64 and 132×33×88 grids—comparison of PCA computational effort for different N_b (no truncation) and $\epsilon_{\mathrm{rb}} = 0.05$

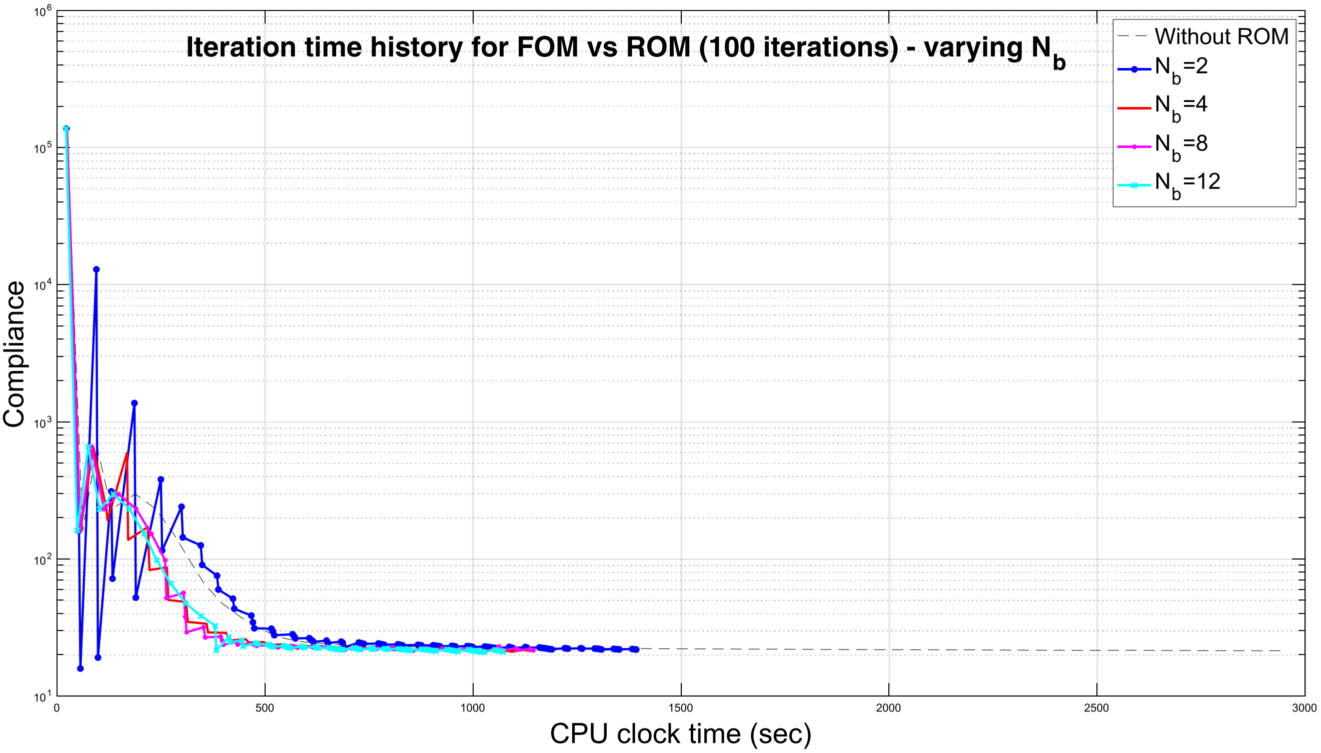

Fig. 8 132×33×88 grids—semilog plot comparison of PCA computational effort for different N_b (no truncation) and $\epsilon_{\mathrm{rb}} = 0.05$

Table 2 Performance comparison for various N_b over 100 iterations for the first 3D test case (using a 132×33×88 grid)

N_b (modes)	calls to FEM	Calls to ROM	Break point	CPU time (sec)	Compliance
2	29	71	24	1930.1	20.846
4	24	76	22	1748.4	20.333
6	22	78	24	1641.1	19.72
8	23	77	28	1590.3	19.553
10	10	90	20	864.29	14.902
14	14	86	28	1064.9	15.188
18	36	74	36	2252.5	24.678
20	42	58	40	2515.0	31.331

coupled approach, for 100 iterations, and the scalability of the savings, respectively. The break point represents the transition where more calls have been made to the ROM rather than the full-field model. It is immediately evident that the number of function calls to the full-field FEM drops off and stabilizes as the number of calls to the significantly less computationally intensive PCA routine increases gradually (after the first N_b iterations and progressively stabilizes). This leads to a dramatic reduction in computational time and effort as seen from the CPU times required for each case.

It is thus clear that coupling the ROM using the on-the-fly calculated PCA basis significantly improves the computational efficiency of the overall optimization routine. This improvement scales up with the grid resolution. Next, we will attempt to identify some ”best practices” for choosing appropriate N_b and ϵ_{rb}.

3.2.2 Performance of ROM with varying N_b and ϵ_{rb}

For this parametric study, we have used all the snapshots without truncation of the basis ($N_b = M$). In the first part, we vary N_b (number of modes/snapshots) from 2 to 20, so

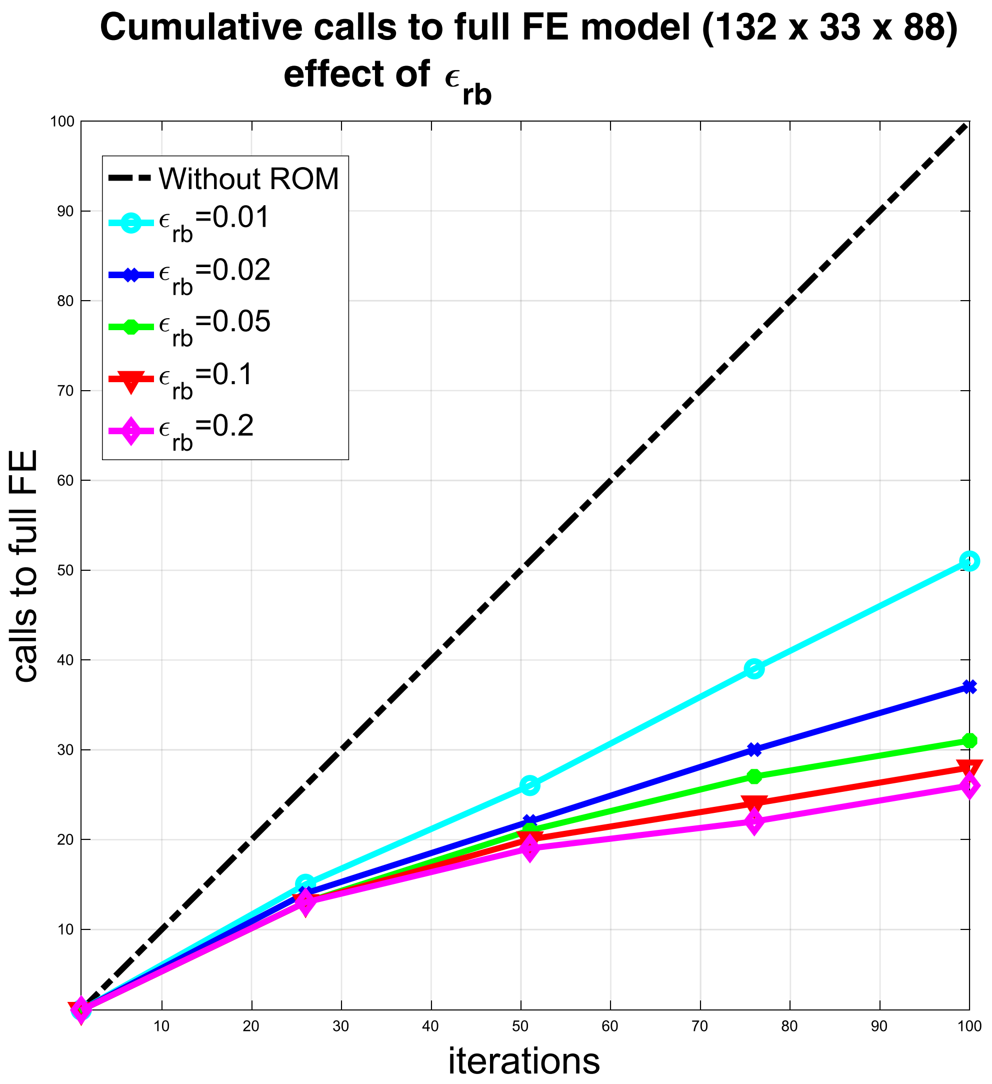

Fig. 9 132×33×88 grid comparison of PCA computational effort for $N_b = 5$ modes and varying ϵ_{rb} (no truncation)

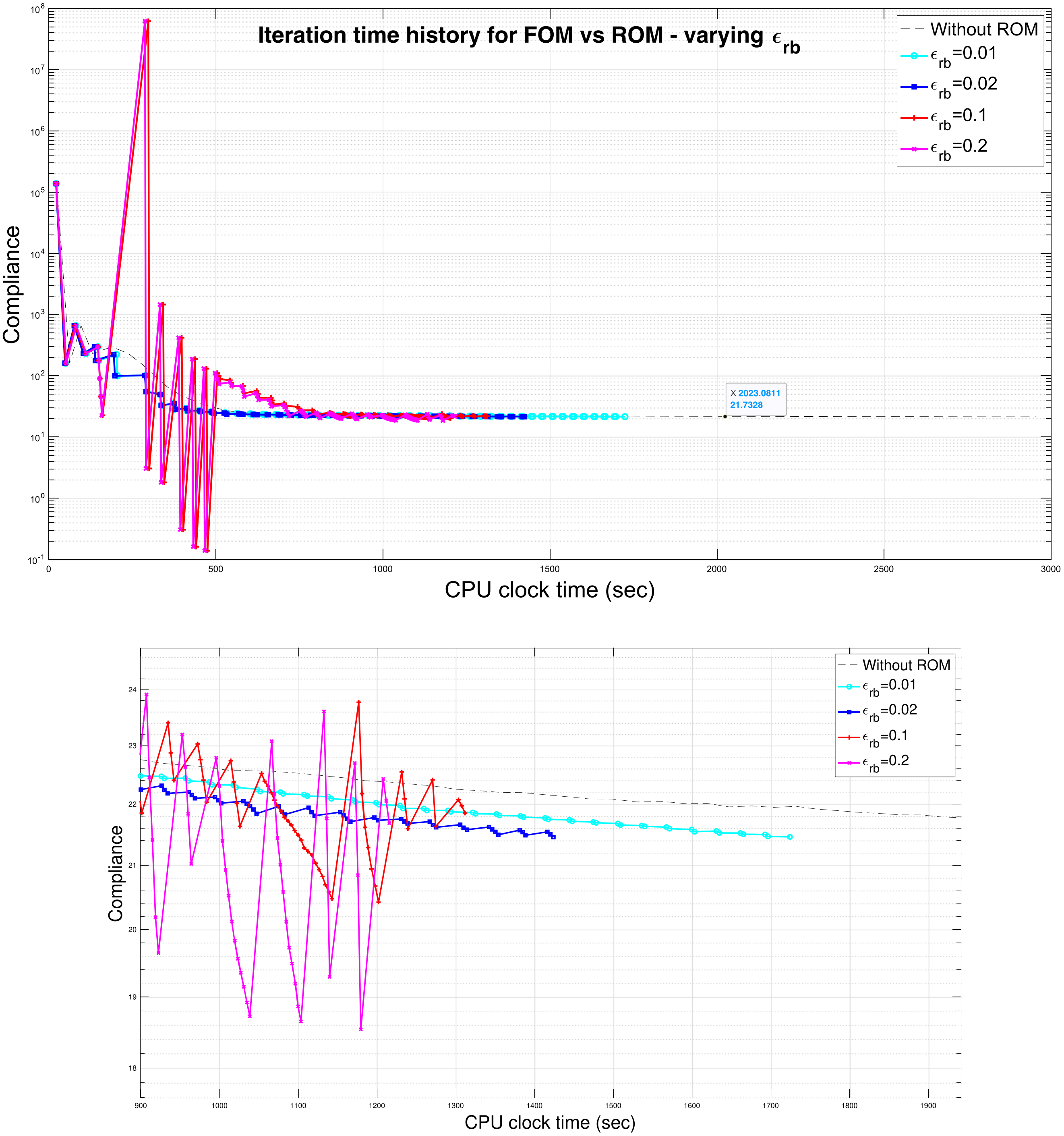

Fig. 10 132×33×88 grid—semilog plot comparison of PCA computational effort for different ϵ_{rb} (no truncation) and $N_b = 5$ (below) zoomed in to the marked region

as to compare the number of calls to the ROM with calls to the full-field solution, as earlier. The threshold is fixed at $\epsilon_{\text{rb}} = 0.1$. The results are shown in Fig. 7a, b for two different grid resolutions and Fig. 8.

These results are summarized in Table 2. It is interesting that there is no monotonic relationship between N_b and the number of full-field calls, and 10 modes being the ideal basis size for this particular problem.

Table 3 Performance comparison for various ϵ_{rb} over 100 iterations for the first 3D test case (using a 132×33×88 grid)

ϵ_{rb}	Calls to FEM	Calls to ROM	Break point	CPU time (sec)	Compliance
0.01	48	52	62	2506.5	20.156
0.02	32	68	28	1931.0	20.098
0.05	29	71	24	1815.0	20.158
0.1	24	76	12	2018.9	20.422

Figures 9 and 10 show the influence of varying the error threshold ϵ_{rb} from 0.01 to 0.2 on the performance of the ROM-coupled topology optimization, when the number of snapshots/modes N_b is fixed at 5. There is an “expectedly” monotonic trend in the number of full-field calls with reducing ϵ_{rb}.

The above results are summarized in Table 3.

While calls to the ROM/full-field model are a crucial performance indicator, it is important to distinguish between a reduction in full-field calls and a reduction in CPU time. If a full-field call is followed by a single ROM call before we require another full-field call, we have gained nothing from the ROM. The CPU time reduction is therefore the final litmus test for the ROM.

Summing up, the error threshold determines the position of the “break/transition point” where the optimizer makes more calls to the ROM compared to the full-field FE solution, since raising ϵ_{rb} increases the admissibility of the ROM solution U_{rb}, thus increasing the number of calls to the ROM while reducing the calls to the full-field FEM. However, there is a tradeoff since increasing the threshold beyond a certain point reduces the precision of the solution, thus potentially reducing the performance of the procedure. For this particular problem, 0.05 appears to be a reasonably good choice.

One would expect increasing N_b to improve the ROM but this is not necessarily the case. By increasing N_b, we increase the amount of information in the ROM but also the number of less relevant modes, leading to a loss of efficacy. The number of modes to be retained for this particular problem appears to be around 10 where both computational efficiency and precision are both simultaneously maximized. Too few (or too many) modes retained will reduce the performance of the ROM, at least for this case.

3.3 3D case 2: MBB beam

We next consider another classical 3D benchmark topology optimization test case: the original Messerschmitt–Bolkow–Blohm/MBB problem in 3D. The boundary conditions of the beam are given in Fig. 11. Just like in the previous test case, we study the effect of N_b, ϵ_{rb}, and grid resolution (for scalability). The elastic parameters are the same as before, i.e., Young’s moduli (maximum and minimum), Poisson’s ratio. v_{frac} is chosen as 0.1, the penalization = 3, and the density filter radius is 0.5.

In addition, three different maximum allowable volume fractions v_{frac} have been considered: 0.1, 0.2, and 0.3.

As in the previous test case, we will focus on the influence of different v_{frac} $((\sum_1^N v_e)/V)$, ROM error threshold ϵ_{rb}, and the number of snapshots N_b used to construct the basis Φ, as well as the scalability with grid resolution.

3.3.1 Performance and scalability of ROM

Three different grid resolutions (in voxels) were considered in this work: a fairly coarse 12×12×72 grid, a finer 24×24×144 grid, and (c) very fine grid 48×48×288.

The volume fraction $v_{\text{frac}} = 0.1$ here.

Figure 12 shows the optimized topologies generated by the TopOpt–PCA algorithm, which are, as expected, visually indistinguishable from those obtained without using the ROM. The results are shown in Fig. 13.

3.3.2 Performance of ROM with varying N_b and ϵ_{rb}

In the first part, we have used all the snapshots without truncation of the basis, and varied N_b (number of modes/snapshots) from 2 to 20, and compared the number of calls to the ROM with calls to the full-field solution, with the threshold $\epsilon_{\text{rb}} = 0.1$ (fixed). In the second part, we show the influence of varying the error threshold ϵ_{rb} from 0.02 to 0.1 on the performance of the ROM-coupled topology optimization routine. The number of snapshots/modes N_b here is fixed at 8.

The results are shown in Figs. 14, 15, and 16. with a summary given in Tables 4 and 5.

From the above results, it is clear that N_b and ϵ_{rb} are vital parameters, which are unfortunately problem and grid resolution dependent.

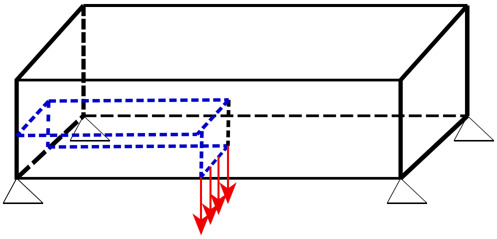

Fig. 11 Boundary conditions for the second test case: 3D MBB beam problem

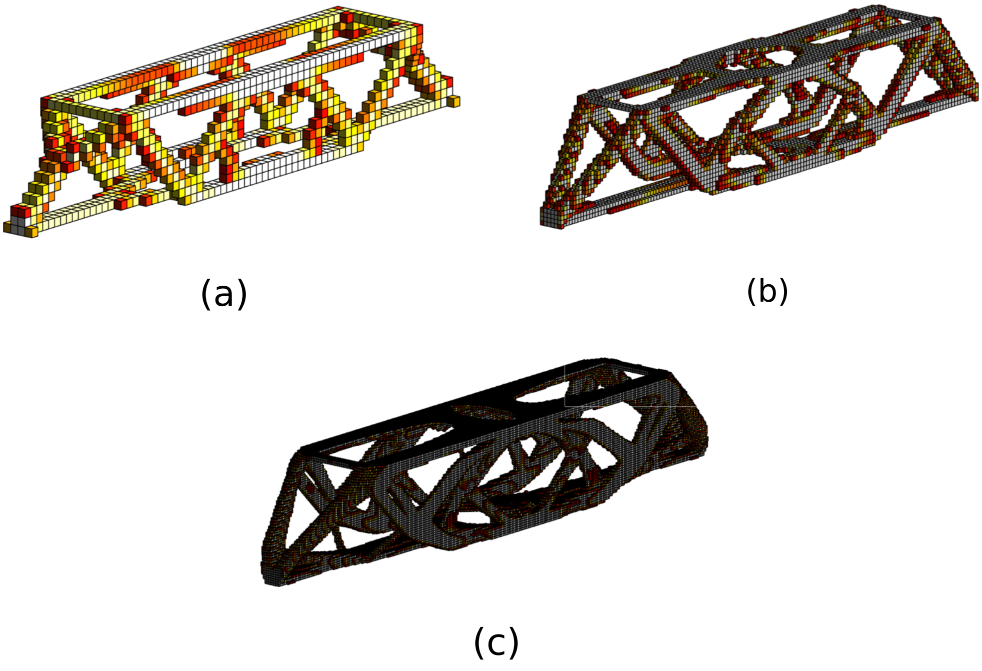

Fig. 12 Comparing optimized 3D topologies for the MBB beam problem, using three different grids with increasing resolution **a** coarse 12×12×72 grid, **b** finer 24×24×144 grid), and **c** very fine grid (48×48×288) obtained using PCA

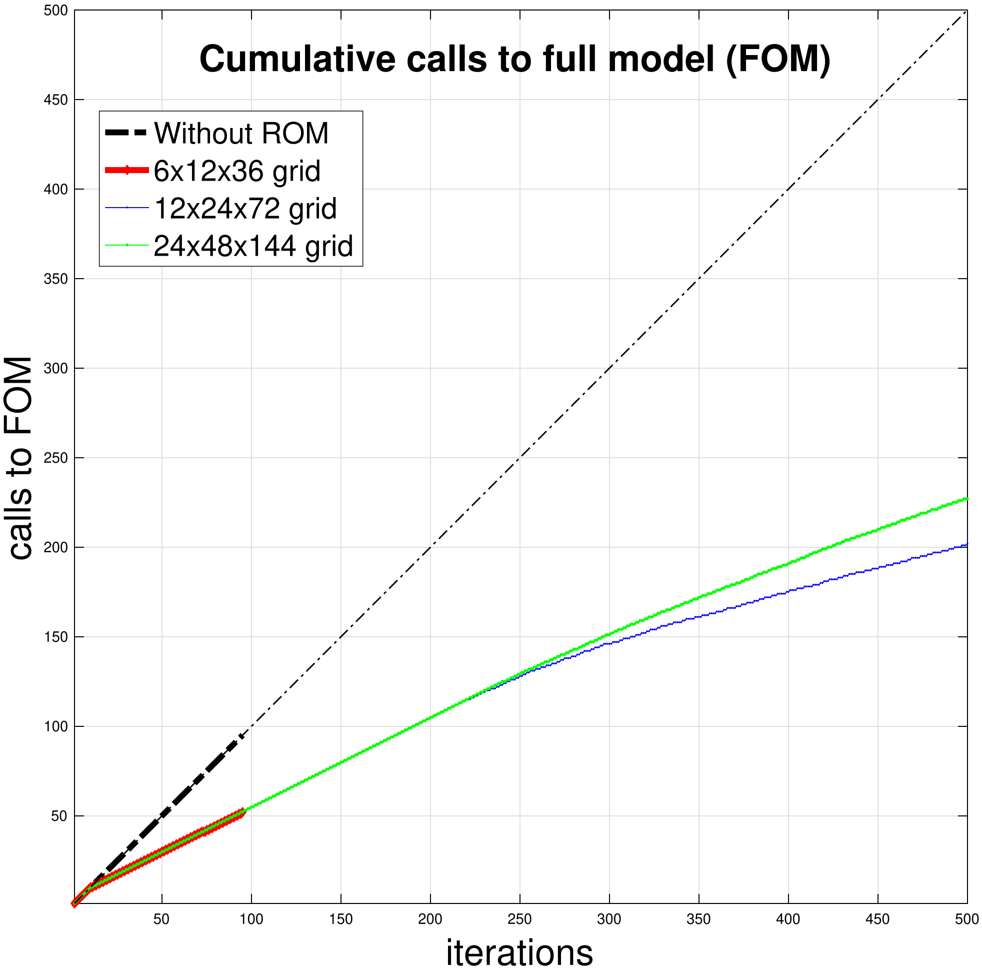

Fig. 13 Comparing number of function calls to FE solver vs. PCA using $Nb = 4$ modes (10 total snapshots) and $\epsilon_{\mathrm{rb}} = 0.1$ for three grids with increasing resolution)

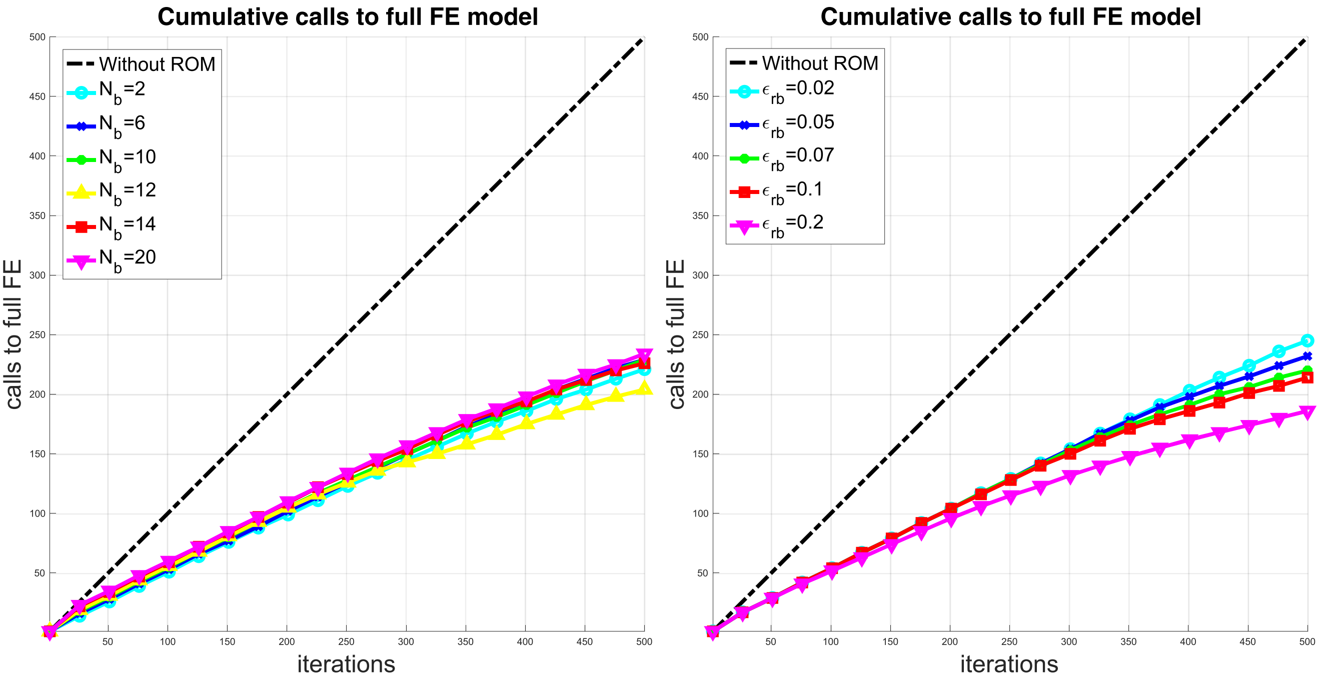

Fig. 14 Performance of PCA varying **a** N_b (without truncation and $\epsilon_{\mathrm{rb}} = 0.01$) and **b** ϵ_{rb} (using $N_b = 8$ modes) on a 24×24×144 grid

3.3.3 Effect of material volume fraction

Finally, we consider three different $v_{\mathrm{frac}} = 0.1, 0.2$, and 0.3 in order to study the evolution of the computational savings with increasing material volume fraction. The results are shown in Fig. 17 below:

The corresponding optimized topologies are shown below in Fig. 18: It is interesting to note that the material volume fraction has a striking influence on the ROM performance. As we increase material volume fraction, the proportion of calls to the ROM increases. In Bendsoe and Sigmund (2004), it is noted that for low v_{frac} (i.e., below 10%), the convergence of the topology optimization routine becomes more tedious due to oscillations. The benefit of using the ROM is in being able to avoid unnecessary full-field calculations by extracting the most relevant modes (of the density map).

3.4 Discussion

The PCA algorithm significantly enhances the performance of the topology optimization routine with a significant reduction in computational effort and CPU time in all test cases investigated. We note that the improvement in performance scales up with the grid resolution. It is also clear that there is an improvement in the reduction in computational effort as we increase the volume fraction—though this may simply be because the higher volume fraction problem would be expected to converge faster.

A conceivably less obvious advantage of the “on-the-fly” ROM, applied to the displacement vector, with constant monitoring for precision using the full-field model as a stand-by, very likely allows for a basis size (N_b)-independence of the optimized density map. It stands to reason that if ϵ_{rb} were inflated to an unreasonable level, we would lose this benefit.

4 Perspectives: extension of approach to non self-adjoint problems

We have, in this paper, focused on developing an ROM approach for self-adjoint problems, with a primary focus on the popular compliance minimization. Consider now a typical compliant mechanism design problem (shown in Fig. 19) The input end A is subjected to a horizontal concentrated load $F_{\mathrm{in}} = 100$ towards the right. Our objective is to maximize the displacement u_{out} of output point B.

Here, we consider the simplest possible type of compliant mechanism in which the displacement (U_{out}) is prescribed at a given node or set of nodes using the sparse vector $\tilde{L}$.

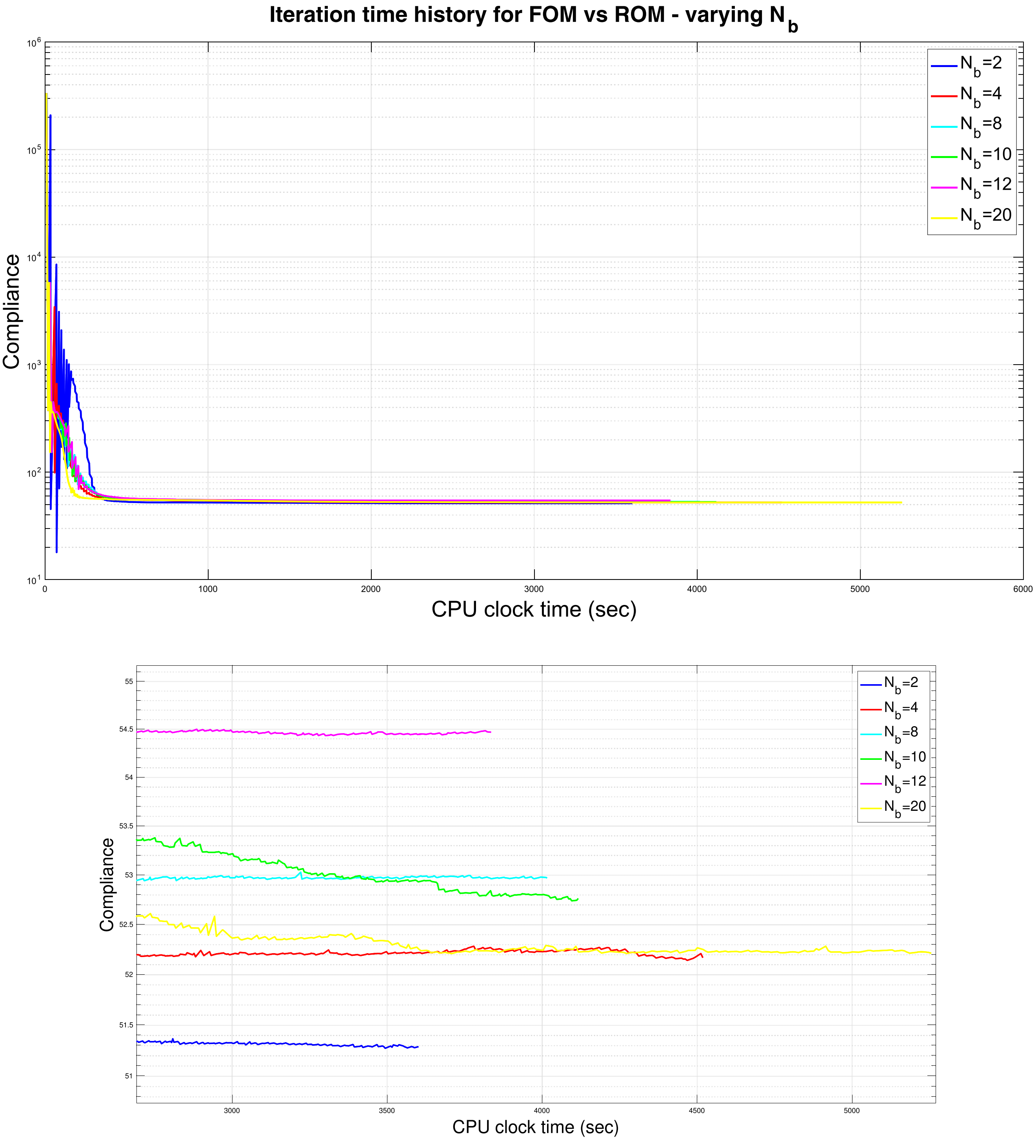

Fig. 15 24×24×144 grid—semilog plot comparison of PCA computational effort for different N_b (no truncation) and $\epsilon_{\text{rb}} = 0.01$

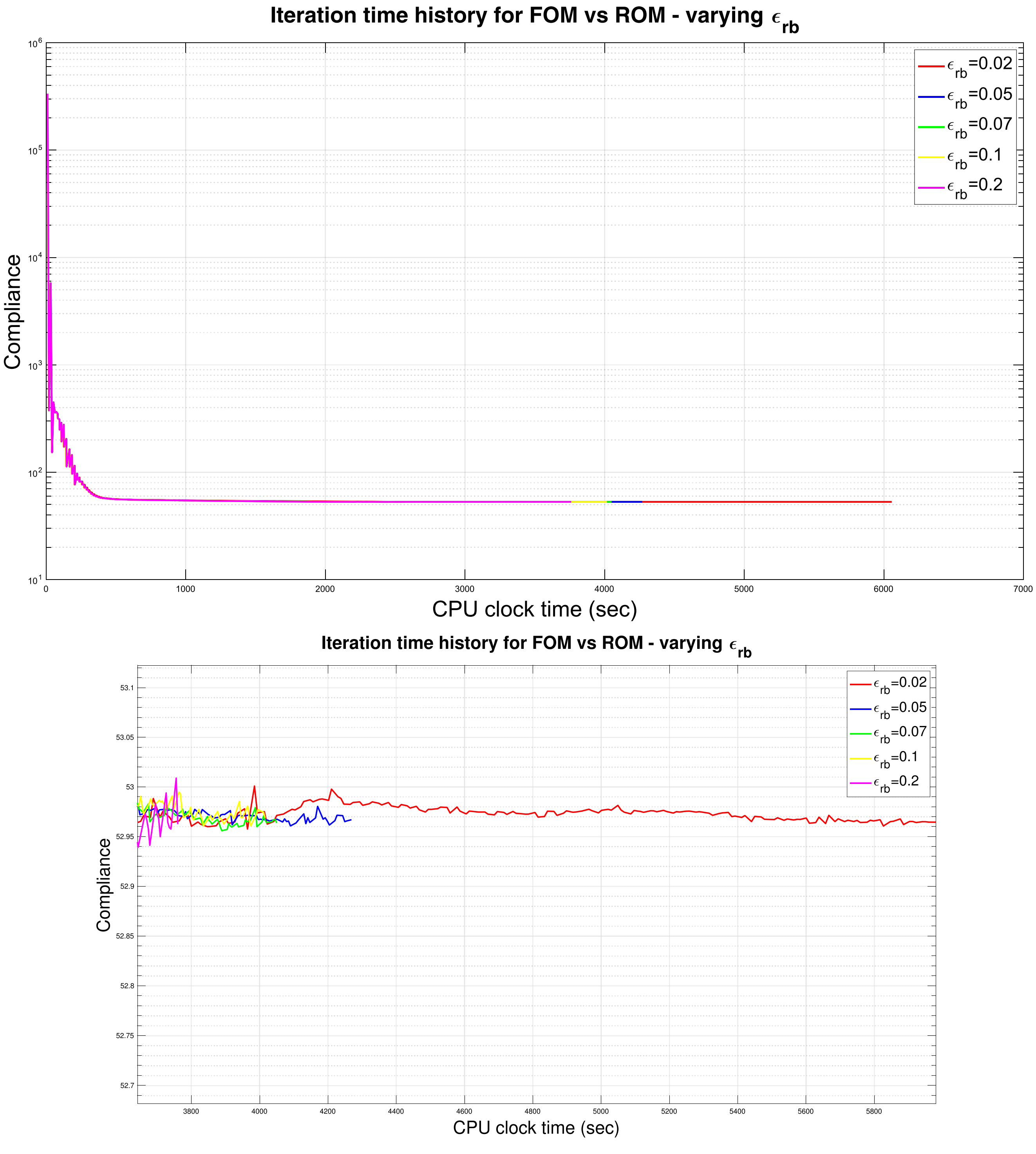

Fig. 16 24×24×144 grid—semilog plot comparison of PCA computational effort for different ϵ_{rb} and $N_b = 8$ modes

Table 4 Performance comparison for various N_b for 500 iterations for the second 3D test case (using a 24×24×144 grid)

N_b (modes)	Calls to FEM	Calls to ROM	Break point	CPU time (sec)	Compliance
2	279	221	144	3591.6	51.286
4	285	215	254	4509.6	52.166
6	271	229	236	4002.0	52.963
8	286	214	292	4006.2	52.969
10	271	229	292	4107.3	52.761
14	277	223	272	4317.7	52.130
18	274	226	356	4348.7	52.186
20	266	234	374	5246.6	52.211

Table 5 Performance comparison for various ϵ_{rb} for 500 iterations for the second 3D test case (using a 24×24×144 grid)

ϵ_{rb}	Calls to FEM	Calls to ROM	Break point	CPU time (sec)	Compliance
0.02	245	255	434	6047.5	52.969
0.05	232	268	378	4258.3	52.967
0.07	220	280	336	4040.7	52.964
0.1	214	276	292	4006.2	52.969
0.2	186	314	124	3750.4	52.962

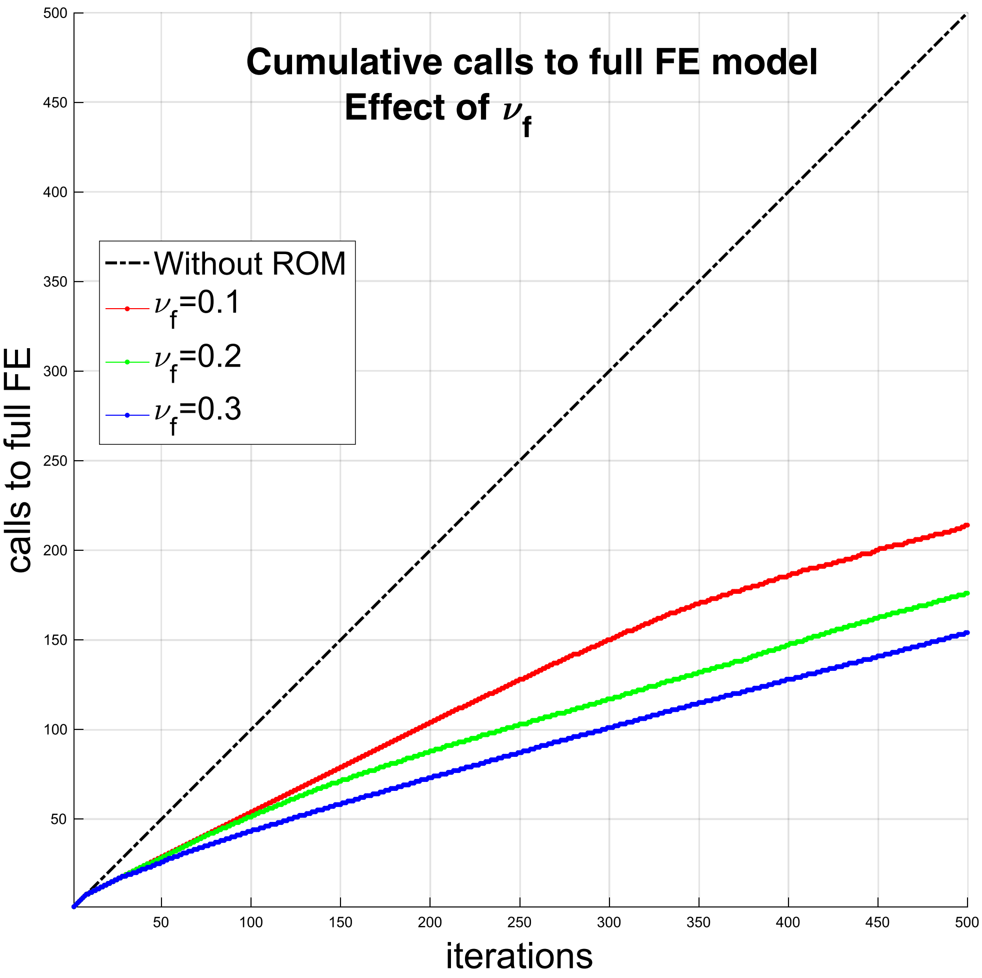

Fig. 17 ROM (PCA using $N_b = 8$ modes without truncation and $\epsilon_{\mathrm{rb}} = 0.1$) performance for three different volume fractions (v_{frac}) 0.1, 0.2 and 0.3 on a 24×24×144 grid

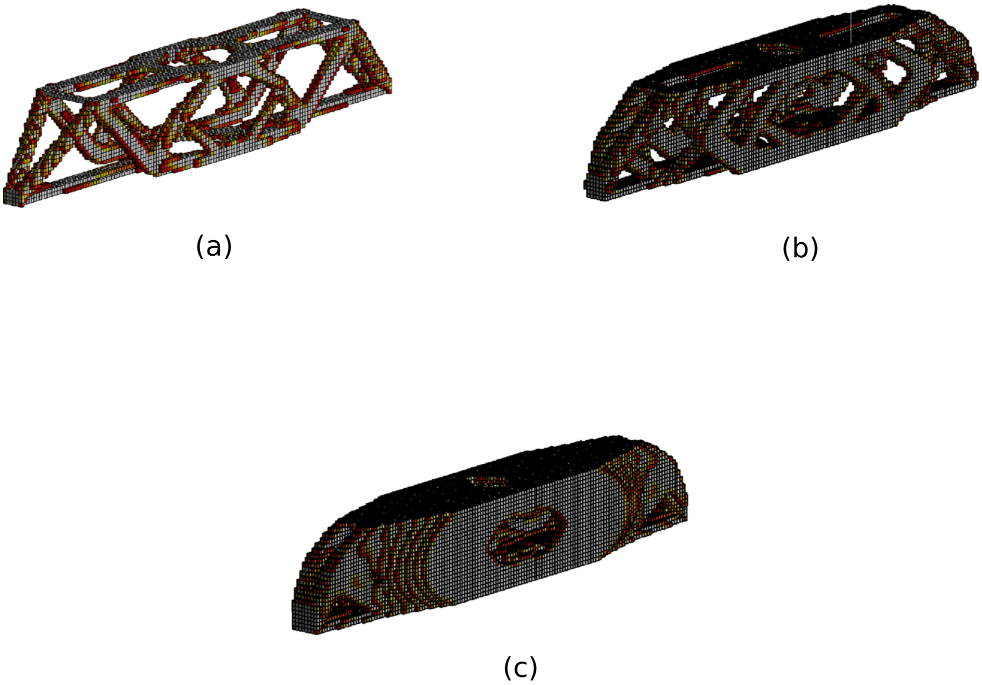

Fig. 18 3D topologies for the three volume fractions **a** 0.1, **b** 0.2, and **c** 0.3 on a 24×24×144 grid

The optimization problem may then be posed as follows:

$$\begin{aligned} &\max_{\rho} U_{\text{out}}(\rho) = \tilde{L}^T U = \tilde{L}^T(\Phi\alpha + \bar{u}) \\ &\qquad \text{such that } \Phi^T K(\Phi\alpha + \bar{u}) = F \\ &\qquad\qquad \sum_{e=1}^{N} v_e \rho_e = v_{\text{frac}} V < V \\ &\qquad \rho_e \in [0, 1], \quad e = 1, \ldots, N \end{aligned} \tag{25}$$

Following Sections 2.2 and 2.3:

$$\begin{aligned} \mathcal{L}(\rho, \mu, \lambda) &= \tilde{L}^T K(\Phi\alpha + \bar{u}) - \mu^T \Phi^T [K(\Phi\alpha + \bar{u}) - F] \\ &\quad - \sum_{i=1}^{N_b} \lambda_i^T (K_i U_i - F) \end{aligned} \tag{26}$$

Using the same reasoning in Section 2.3, for the on-the-fly updation strategy, the basis Φ is not a continuously evolving function of ρ_e, we state that $\frac{\partial \Phi}{\partial \rho}$, $\frac{\partial \bar{u}}{\partial \rho}$ as well as the last two terms in the derivative vanish giving the following:

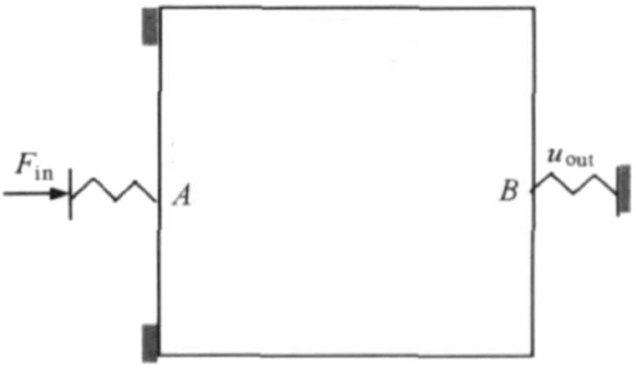

Fig. 19 Displacement–inverter topology optimization problem

$$\frac{\partial \mathcal{L}}{\partial \rho_e} = (\tilde{L}^T \Phi - \mu^T \Phi^T K \Phi) \frac{\partial \alpha}{\partial \rho_e} - \mu^T \Phi^T \frac{\partial K}{\partial \rho_e} U_{\text{rb}} \tag{27}$$

We choose μ such that:

$$(\Phi^T K \Phi)\mu = K_{\text{rb}}\mu = \Phi^T \tilde{L} \tag{28}$$

where K_{rb} is the reduced stiffness matrix from (10), allowing for inexpensive inversion, this giving us the simple expression for the reduced sensitivity:

$$\frac{\partial \mathcal{L}}{\partial \rho_e} = -\mu^T \Phi^T \frac{\partial K}{\partial \rho_e} U_{\text{rb}} \tag{29}$$

The solution of (28) is then used to calculate the reduced sensitivity from (29). But the system in (28) has reduced dimensionality compared to (3), indicating that we now have a single reduced-order system with two load cases to solve.

We now apply both the above modified "on-the-fly" POD ROM as well as the Gram–Schmidt orthogonalization (Gogu 2015) and investigate the influence of the type of ROM, N_b and ϵ_{rb} on the results obtained for a displacement inverter. v_{frac} is set as 0.3 and the MMA algorithm is used for the optimization. Material elastic modulus is 1, the minimum (void) elastic modulus is 10^{-9}, and Poisson ratio is 0.3. The SIMP penalty factor is 3 and the filter radius is 1.5 (using sensitivity filtering). Optimization terminates when the maximum elemental density variation < 0.1% or 400 iterations have been completed.

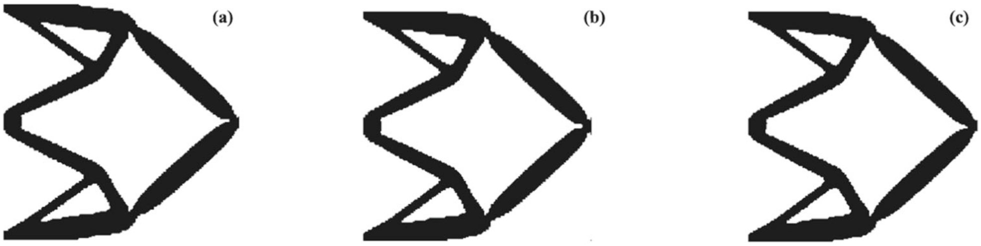

Fig. 20 Optimal topologies obtained **a** without ROM, b G-S, and **c** POD

In the design domain shown, the upper and lower ends on the left are simply supported, middle nodes of the left and right boundaries are input (load) end and output end (displacement) respectively. The structure is discretized by 100×100 square elements of unit volume. Linear springs simulate the structural stiffness of the input end and output end ($k_{\text{in}} = k_{\text{out}} = 1$). Figure 20 shows the optimal topologies of the reference model, POD and Gram–Schmidt orthogonalization (simply referred to as G-S) ROMs for $N_b = 5, \epsilon_{\text{rb}} = 0.01$.

The optimal topology obtained using the POD model is almost the same as that of G-S model as well as the reference model, by visual inspection, satisfying the property of vertical symmetry and the requirements of mechanical properties as well as actual processing and manufacturing, indicating that the proposed method can meet the requirements of high-accuracy design.

Table 6 compares the results of the two ROMs (POD and G-S) by varying ϵ_{rb} and N_b. From Table 6, we see that for the same ROM parameters (N_b and ϵ_{rb}), there are significantly more calls to the POD ROM than the G-S, particularly for smaller values of ϵ_{rb}, not to mention the ROM is used far more frequently than the full-field solution. We again note that CPU time is not necessarily proportional to the number of full FE calls, since oversampling could potentially increase the cost of updating the reduced basis, and any reduction in full FE calls can no longer make up for the time gap.The top speed-up for the POD ROM is 1.47 (corresponding to 1.37 for the G-S), and time-saving is about 32% (against 27% for the G-S) and for low N_b and high ϵ_{rb}, the optimization efficiency is higher.

5 Conclusions

In this paper, we have presented an approach for efficient large-scale topology optimization based on coupling of topology optimization with reduced-order modeling by principal components analysis, using on-the-fly construction of the reduced basis with a database of previously calculated solutions of the FE equations.

Topology optimization coupled with on-the-fly PCA calculated basis is seen to significantly outperform the classical approach. It is important to note that we avoid storage of the "temporary" stiffness matrices and basis vectors during the "basis-changing" iterations, which means

Table 6 Performance comparison of G-S ROM and POD ROM (with reference)

Method	ϵ_{rb}	N_b	Calls to ROM	Calls to full FE	CPU time time (s)	Speedup	Relative error (c)
Reference	–	–	0	400	132.43	1	0
G-S	0.01	5	337	63	96.48	1.37	0.35
		10	330	70	106.18	1.25	0.23
		40	315	85	116.77	1.13	0.20
	0.001	5	228	172	105.03	1.26	0.12
		10	156	244	120.78	1.10	0.05
		40	181	219	127.49	1.04	0.11
PCA	0.01	5	351	49	90.3	1.47	0.27
		10	346	54	103.63	1.28	0.09
		40	326	74	115.59	1.15	0.06
	0.001	5	279	121	102.73	1.29	0.18
		10	263	137	108.74	1.22	0.07
		40	259	141	134.57	0.98	0.02

that the storage requirement is significantly reduced compared to previous methods. The PCA approach showed a significant reduction in computational effort over the traditional full-field solution approach. The improvement in performance scales well with the size of the problem.

While we have focused on the compliance minimization problem, the current method should be applicable to virtually any self-adjoint topology optimization problem, regardless of the particular physics involved.

Another obvious area of immediate work is using high-performance computing and non-intrusive asynchronous PCA to obtain additional improvement in the computational time and effort needed.

Finally, a formal extension of the approach to general non self-adjoint problems is a key area of future research.

Acknowledgments The first and third author formally acknowledge the contribution of their Masters student Mr. Jin Wentao, Huawei inc. to this work. The third and fourth author express their sincere gratitude to Mr. Tabrej Alam, Masters student at NIT Silchar for his help in testing the codes in its initial stages.

Funding information This multi-national research study was supported by the National Natural Science Foundation of China (Grant No. 11620101002 and Grant No. 11972166) and the Fundamental Research Funds for the Central Universities (Grant No. 310201911cx029).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results The source codes in this work are an evolution of the 88-line Matlab code, according to the proposed methodology, along with the definition of test cases, which allow to reproduce the numerical results presented in this paper. These codes could be made available on request by emailing the corresponding author.

References

Aage N, Lazarov BS (2013) Parallel framework for topology optimization using the method of moving asymptotes. Struct Multidiscip Optim 47(4):493–505. https://doi.org/10.1007/s00158-012-0869-2

Aage N, Andreassen E, Lazarov BS (2015) Topology optimization using PETSc: an easy-to-use, fully parallel, open source topology optimization framework. Struct Multidiscip Optim 51(3):565–572. https://doi.org/10.1007/s00158-014-1157-0

Aage N, Andreassen E, Lazarov B, Sigmund O (2017) Giga-voxel computational morphogenesis for structural design. Nature 550(7674):84–86. https://doi.org/10.1038/nature23911

Alaimo G, Auricchio F, Bianchini I, Lanzarone E (2018) Applying functional principal components to structural topology optimization. Int J Numer Methods Eng 115(2):189–208. https://doi.org/10.1002/nme.5801

Allaire G, Jouve F, Toader A-M (2004) Structural optimization using sensitivity analysis and a level-set method. J Comput Phys 194(1):363–393. https://doi.org/10.1016/j.jcp.2003.09.032

Amir O, Bendsoe MP, Sigmund O (2009) Approximate reanalysis in topology optimization. Int J Numer Methods Eng 78(12):1474–1491. https://doi.org/10.1002/nme.2536

Amir O, Stolpe M, Sigmund O (2010) Efficient use of iterative solvers in nested topology optimization. Struct Multidisc Optim 42(1):55–72. https://doi.org/10.1007/s00158-009-0463-4

Amir O, Sigmund O, Lazarov BS, Schevenels M (2012) Efficient reanalysis techniques for robust topology optimization. Comput Methods Appl Mech Eng 245-246:217–231. https://doi.org/10.1016/j.cma.2012.07.008

Amsallem D, Zahr M, Choi Y, Farhat C (2015) Design optimization using hyper-reduced-order models. Struct Multidiscip Optim 51(4):919–940. https://doi.org/10.1007/s00158-014-1183-y

Bendsoe MP, Sigmund O (2004) Topology optimization: theory, methods and applications. Springer

Bendsoe M (1989) Optimal shape design as a material distribution problem. Struct Optim 1 (193). https://doi.org/10.1007/BF01650949

Berkooz G, Holmes P, Lumley JL (1993) The proper orthogonal decomposition in the analysis of turbulent flows. Annu Rev Fluid Mech 25(1):539–575. https://doi.org/10.1146/annurev.fl.25.010193.002543

Chinesta F, Ladeveze P, Cueto E (2011) A short review on model order reduction based on proper generalized decomposition. Arch Comput Methods Eng 18(4):395. https://doi.org/10.1007/s11831-011-9064-7

Choi Y, Oxberry G, White D, Kirchdoerfer T (2019) Accelerating design optimization using reduced order models. arXiv:1909.11320

Deaton JD, Grandhi RV (2014) A survey of structural and multidisciplinary continuum topology optimization: post 2000. Struct Multidiscip Optim 49(1):1–38. https://doi.org/10.1007/s00158-013-0956-z

Dulong J-L, Druesne F, Villon P (2007) A model reduction approach for real-time part deformation with nonlinear mechanical behavior. Int J Interact Des Manuf (IJIDeM) 1(4):229. https://doi.org/10.1007/s12008-007-0028-y

Dutta S, Ghosh S, Inamdar MM (2018) Optimisation of tensile membrane structures under uncertain wind loads using PCE and kriging based metamodels. Struct Multidiscip Optim 57(3):1149–1161. https://doi.org/10.1007/s00158-017-1802-5

Ferro N, Micheletti S, Perotto S (2019) Pod-assisted strategies for structural topology optimization. Computers & Mathematics with Applications. https://doi.org/10.1016/j.camwa.2019.01.010

Gogu C (2015) Improving the efficiency of large scale topology optimization through on-the-fly reduced order model construction. Int J Numer Methods Eng 101(4):281–304. https://doi.org/10.1002/nme.4797

He JJ, Jiang JS (2012) New method of dynamical reanalysis for large modification of structural topology based on reduced model. In: Manufacturing science and materials engineering, vol. 443 of advanced materials research. Trans Tech Publications, pp 628–631, https://doi.org/10.4028/www.scientific.net/AMR.443-444.628

Hoang K, Kerfriden P, Bordas S (2016) A fast, certified and ‘tuning free’ two-field reduced basis method for the metamodelling of affinely-parametrised elasticity problems. Comput Methods Appl Mech Eng 298:121–158. https://doi.org/10.1016/j.cma.2015.08.016

Kirsch U, Bogomolni M (2004) Procedures for approximate eigenproblem reanalysis of structures. Int J Numer Methods Eng 60(12):1969–1986. https://doi.org/10.1002/nme.1032

Kirsch U, Papalambros P (2001) Structural reanalysis for topological modifications – a unified approach. Struct Multidiscip Optim 21(5):333–344. https://doi.org/10.1007/s001580100112

Madra A, Breitkopf P, Raghavan B, Trochu F (2018) Diffuse manifold learning of the geometry of woven reinforcements in composites. Comptes Rendus Mécanique 346(7):532–538. https://doi.org/10.1016/j.crme.2018.04.008

Mahdavi A, Balaji R, Frecker M, Mockensturm EM (2006) Topology optimization of 2D continua for minimum compliance using parallel computing. Struct Multidiscip Optim 32(2):121–132. https://doi.org/10.1007/s00158-006-0006-1

Meng L, Breitkopf P, Quilliec GL, Raghavan B, Villon P (2018) Nonlinear shape-manifold learning approach: concepts, tools and applications. Arch Comput Methods Eng 25(1):1–21. https://doi.org/10.1007/s11831-016-9189-9

Meng L, Breitkopf P, Raghavan B, Mauvoisin G, Bartier O, Hernot X (2019a) On the study of mystical materials identified by indentation on power law and voce hardening solids. Int J Mater Form 12(4):587–602. https://doi.org/10.1007/s12289-018-1436-1

Meng L, Zhang W, Quan D, Shi G, Tang L, Hou Y, Breitkopf P, Zhu J, Gao T (2019b) From topology optimization design to additive manufacturing: today's success and tomorrow's roadmap. Archives of Computational Methods in Engineering. https://doi.org/10.1007/s11831-019-09331-1

Norato JA, Bendsøe MP, Haber RB, Tortorelli DA (2007) A topological derivative method for topology optimization. Struct Multidiscip Optim 33(4):375–386. https://doi.org/10.1007/s00158-007-0094-6

Pearson K (1901) LIII. On lines and planes of closest fit to systems of points in space. The London Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2(11):559–572. https://doi.org/10.1080/14786440109462720

Raghavan B, Breitkopf P (2013) Asynchronous evolutionary shape optimization based on high-quality surrogates: application to an air-conditioning duct. Eng Comput 29(4):467–476. https://doi.org/10.1007/s00366-012-0263-0

Raghavan B, Breitkopf P, Tourbier Y, Villon P (2013a) Towards a space reduction approach for efficient structural shape optimization. Struct Multidiscip Optim 48(5):987–1000. https://doi.org/10.1007/s00158-013-0942-5

Raghavan B, Hamdaoui M, Xiao M, Breitkopf P, Villon P (2013b) A bi-level meta-modeling approach for structural optimization using modified pod bases and diffuse approximation. Comput Struct 127:19–28. https://doi.org/10.1016/j.compstruc.2012.06.008

Ryckelynck D, Chinesta F, Cueto E, Ammar A (2006) On thea priori model reduction: overview and recent developments. Arch Comput Methods Eng 13(1):91–128. https://doi.org/10.1007/BF02905932

Saxena A, Ananthasuresh G (2000) On an optimal property of compliant topologies. Struct Multidiscip Optim 19(1):36–49. https://doi.org/10.1007/s001580050084

Senne TA, Gomes FAM, Santos SA (2019) On the approximate reanalysis technique in topology optimization. Optim Eng 20(1):251–275. https://doi.org/10.1007/s11081-018-9408-3

Sigmund O (2001) A 99 line topology optimization code written in matlab. Struct Multidiscip Optim 21(2):120–127. https://doi.org/10.1007/s001580050176

Sun Y, Zhao X, Yu Y, Zheng S (2018) An efficient reanalysis method for topological optimization of vibrating continuum structures for simple and multiple eigenfrequencies. Math Probl Eng 2018:1–10

Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. Int J Numer Methods Eng 24(2):359–373. https://doi.org/10.1002/nme.1620240207

Svanberg K (2002) A class of globally convergent optimization methods based on conservative convex separable approximations. SIAM J Optim 12(2):555–573. https://doi.org/10.1137/S1052623499362822

Tatebe O (1993) The multigrid preconditioned conjugate gradient method. Langley Research Center, The Sixth Copper Mountain Conference on Multigrid Methods, Part 2; pp. 621–634

Wang S, Sturler Ed, Paulino GH (2007) Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. Int J Numer Methods Eng 69(12):2441–2468. https://doi.org/10.1002/nme.1798

Xia L, Breitkopf P (2014) Concurrent topology optimization design of material and structure within FE2 nonlinear multiscale analysis framework. Comput Methods Appl Mech Eng 278:524–542. https://doi.org/10.1016/j.cma.2014.05.022

Xia L, Breitkopf P (2017) Recent advances on topology optimization of multiscale nonlinear structures. Arch Comput Methods Eng 24(2):227–249. https://doi.org/10.1007/s11831-016-9170-7

Xia L, Da D, Yvonnet J (2018) Topology optimization for maximizing the fracture resistance of quasi-brittle composites. Comput Methods Appl Mech Eng 332:234–254. https://doi.org/10.1016/j.cma.2017.12.021

Xiao M, Breitkopf P, Coelho RF, Knopf-Lenoir C, Sidorkiewicz M, Villon P (2009) Model reduction by CPOD and Kriging. Struct Multidiscip Optim 41(4):555–574. https://doi.org/10.1007/s00158-009-0434-9

Xiao M, Zhang G, Breitkopf P, Villon P, Zhang W (2018) Extended co-Kriging interpolation method based on multi-fidelity data. Appl Math Comput 323:120–131. https://doi.org/10.1016/j.amc.2017.10.055

Yin L, Yang W (2001) Optimality criteria method for topology optimization under multiple constraints. Comput Struct 79(20):1839–1850. https://doi.org/10.1016/S0045-7949(01)00126-2

Yoon GH (2010) Structural topology optimization for frequency response problem using model reduction schemes. Comput Methods Appl Mech Eng 199(25):1744–1763. https://doi.org/10.1016/j.cma.2010.02.002

Zheng S, Zhao X, Yu Y, Sun Y (2017) The approximate reanalysis method for topology optimization under harmonic force excitations with multiple frequencies. Struct Multidiscip Optim 56(5):1185–1196. https://doi.org/10.1007/s00158-017-1714-4

Zhou Y, Zhang W, Zhu J (2019) Concurrent shape and topology optimization involving design-dependent pressure loads using implicit b-spline curves. Int J Numer Methods Eng 118(9):495–518. https://doi.org/10.1002/nme.6022