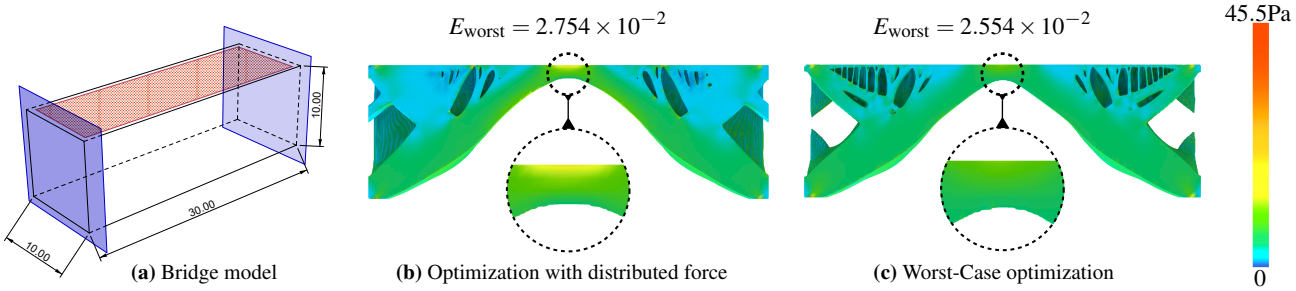
DOI: 10.1111/cgf.14698
Pacific Graphics 2022
N. Umetani, E. Vouga, and C. Wojtan
(Guest Editors)

Volume 41 (2022), Number 7

Large-Scale Worst-Case Topology Optimization

Di Zhang[1] Xiaoya Zhai[1†] Xiao-Ming Fu[1] Heming Wang[2] Ligang Liu [1]

[1]University of Science and Technology of China [2] Harrow International School Hong Kong



(a) Bridge model **(b)** Optimization with distributed force **(c)** Worst-Case optimization

Figure 1: *Large-scale worst-case topology optimization. (a) The Bridge model, i.e., a cuboid mesh with* $512 \times 170 \times 170$ *elements (15M voxel elements). Its both sides are fixed and the load region is the top surface. (b) The optimization result from a force acting uniformly on the top surface. (c) The optimized structure from the worst-case topology optimization. The color encodes the stress from a force that leads to the maximum compliance* E_{worst}*. Our method allows the material to be distributed in the middle of the bridge to strengthen and improve the stiffness (*2.554×10^{-2} *vs.* 2.754×10^{-2}*). It takes about 10 minutes for (b) and 50 minutes for (c) using a desktop computer.*

Abstract

We propose a novel topology optimization method to efficiently minimize the maximum compliance for a high-resolution model bearing uncertain external loads. Central to this approach is a modified power method that can quickly compute the maximum eigenvalue to evaluate the worst-case compliance, enabling our method to be suitable for large-scale topology optimization. After obtaining the worst-case compliance, we use the adjoint variable method to perform the sensitivity analysis for updating the density variables. By iteratively computing the worst-case compliance, performing the sensitivity analysis, and updating the density variables, our algorithm achieves the optimized models with high efficiency. The capability and feasibility of our approach are demonstrated over various large-scale models. Typically, for a model of size $512 \times 170 \times 170$ *and 69934 loading nodes, our method took about 50 minutes on a desktop computer with an NVIDIA GTX 1080Ti graphics card with 11 GB memory.*

Keywords: Worst-case topology optimization, Displacement-oriented problem, Modified inverse power method

1. Introduction

Topology optimization is a fundamental method for structure design and has been used in a wide variety of applications. To solve for the best topology or layout, a number of prior numerical optimization methods provided the foundation, such as density-based methods [Ben89], boundary-based methods [SW00], and topological derivative [SZ99]. Especially, density-based methods are the most widely used based on a binary (0-1) integer optimization.

In practical applications, there are two important factors: (1) handling high-resolution models containing at least millions of voxels [MLR*21, BP01, WDW15] and (2) optimizing the worst case [PRZ17, ZPZ13]. Performing large-scale topology optimization can accurately predict the physical shape properties and adapt to high-resolution low-cost 3D printing devices. The commonly uncertain load conditions in the real world lead to changing compliance, so the maximum compliance should be minimized to enhance the stiffness of the optimized model in the worst case. Thus, our focus is to optimize the maximum compliance for a large-scale model under load uncertainty on a desktop computer.

To solve the worst-case topology optimization problem, the existing methods transform it into a maximum eigenvalue minimization problem [Ped00, CC08, TNKK11] or a semi-definite programming problem [BTN97, HTK15, THK15]. However, the computational cost is extremely high when applying these methods to high-resolution models. Many methods have been proposed to

† Corresponding author: xiaoya93@mail.ustc.edu.cn (Xiaoya Zhai)

achieve high efficiency for the large-scale topology optimization problem [MLR*21], but their goal is not to minimize the worst-case compliance. To the best of our knowledge, previous methods cannot efficiently solve the large-scale worst-case topology optimization under load uncertainty.

Achieving high efficiency for the large-scale worst-case topology optimization is challenging as the maximum eigenvalue minimization problem or the semi-definite programming problem is nonlinear and large-scale. Taking the maximum eigenvalue minimization problem as example, the most time-consuming step is to compute the maximum eigenvalue of a matrix, which changes in each iteration of the optimization [TNKK11]. In practice, the traditional power method can be directly used. However, to obtain the changing matrix, the number of linear systems, whose left side is the stiffness matrix, to be solved is equal to 3 times the number of the loading nodes. Since many loading nodes are often used for the high-resolution model, the time overhead is high despite using a multigrid solver.

In this paper, we propose an efficient method for large-scale worst-case topology optimization problems with uncertain loads. Our method is also to optimize the maximum eigenvalue. The success to improve efficiency is a modified power method for computing the maximum eigenvalue. As the aforementioned changing matrix is the multiplication of three matrices, where the middle is the inverse of the stiffness matrix, we can perform only matrix-vector multiplication in each iteration of the power method, thereby avoiding the explicit construction of the changing matrix. Then, there is only one linear system left to solve. To further reduce the computational cost of the power method, an inexact solution, e.g., performing one V-cycle in the multigrid solver, to the linear system is used. Although strict conditions are needed to ensure that our modified power method outputs the maximum eigenvalue in theory, we find in practice that it always generates the same results as the power method using the exact solution of the linear system.

With the computed maximum eigenvalue, the Optimality Criteria method [BS98] is used to solve the topology optimization problem. Specifically, the adjoint variable method is used to perform the sensitivity analysis for updating the density variables. Consequently, our method can efficiently optimize the worst-case compliance for a large-scale model with uncertain external loads. Code for this paper is at `https://github.com/lavenklau/robtop`.

2. Related Work

Topology optimization Searching for optimal material distribution of structures has been an attractive subject for researchers in recent decades. The Solid Isotropic Material with Penalization (SIMP) method was first introduced in [Ben89] to solve mechanical topology optimization problems. They consider structural optimization problems under specified forces, which is different from our objective. Besides, evolutionary structural optimization (ESO) [XS93] is also one well-known method, which was widely used in engineering. The above two methods are both density-based, i.e., optimizing densities defined on the voxels after discretization. Other methods also use implicit or explicit scalar functions to represent the solid part in the design domain, such as Level Set method [SW00, WWG03], Moving Morphable Component (MMC) [GZZ14], and Triply Periodic Minimal Surfaces (TPMS) based methods [HWL*20, YRL*20]. Usually, these scalar functions are analogs of the signed distance functions. In addition, there are some works related to stress constrained topology optimization [LNB*10, ZCW21], but most of these problems are for fixed load conditions.

Robust topology optimization There are mainly two categories of methods to solve robust topology optimization problems: stochastic [LSS12, LSD*16, CLSK20] and deterministic approaches [HTK15] (also called worst-case approaches). The worst case is where the load produces the largest compliance. [ZPZ13] analyzed the worst case of the structures, but did not perform topology optimization. Later, [PRZ17] proposed a shape optimization to detect fragile locations and strengthen them. Optimizing lightweight structure under force location uncertainty was studied [UMK17]. [SZB18] estimated the loads leading to the weakest structure and then minimize the failure potential. The method of [Ped00] transforms the min-max problem into a maximum eigenvalue problem and applies it to a 2D case. [CC08] and [TNKK11] gave a practical formulation for the derivative of the maximum eigenvalue with respect to the density and minimized the maximum eigenvalue. In general, the evaluation of the maximum eigenvalue is very time-consuming. [BTN97] transformed the optimization problem into a linear, semi-definite, convex program and proved to have polynomial-time complexity. So [OFKK99] and [HTK15] used this method to deal with the small-scale problem. [THK15] further transformed the worst-case compliance design into a non-linear semi-definite problem; but, for a high-resolution model with a large load area, the time and memory costs are still very high.

Large-scale topology optimization For the sake of accuracy, the structure's resolution would reach more than one million or higher. Large-scale topology optimization problems usually appear in industrial applications that need high-performance computing (see the survey in [MLR*21]). There are several typical ways, such as re-analysis [AKS08], multi-grid solvers [WDW15], model reduction [XLCB15], parallel computing [BP01], narrow-band solvers [LHZ*18, LLL*21], etc. There are only a few algorithms for large-scale and robust topology optimization problems [WDW15, MFHP16]. However, these methods heavily rely on computer hardware devices that are difficult to transplant to ordinary desktop computers.

3. Method

3.1. Problem and formulation

Inputs Given a model that is voxelized into a high-resolution domain Ω, we predefine load regions bearing uncertain external loads (Figure 2 (a)) and certain boundary conditions. Each voxel element e is equipped with a density function ρ_e that is parameterized into "0-1". We stack all ρ_e in the column format to form a density vector $\boldsymbol{\rho}$ as design variables. The external load on each node i is denoted as $\mathbf{f}_i$ forming a column vector denoted as $\mathbf{f}$. We assume that the magnitude of the uncertain load $\mathbf{f}$ is bounded, similar to [TNKK11].

Uncertain load conditions Various constraints on $\mathbf{f}_i$ exist as the load $\mathbf{f}$ varies in value and direction. Here, we consider that $\mathbf{f}_i$ is only one of the following three:

- There is no load on the node i, i.e., $\mathbf{f}_i = \mathbf{0}$.

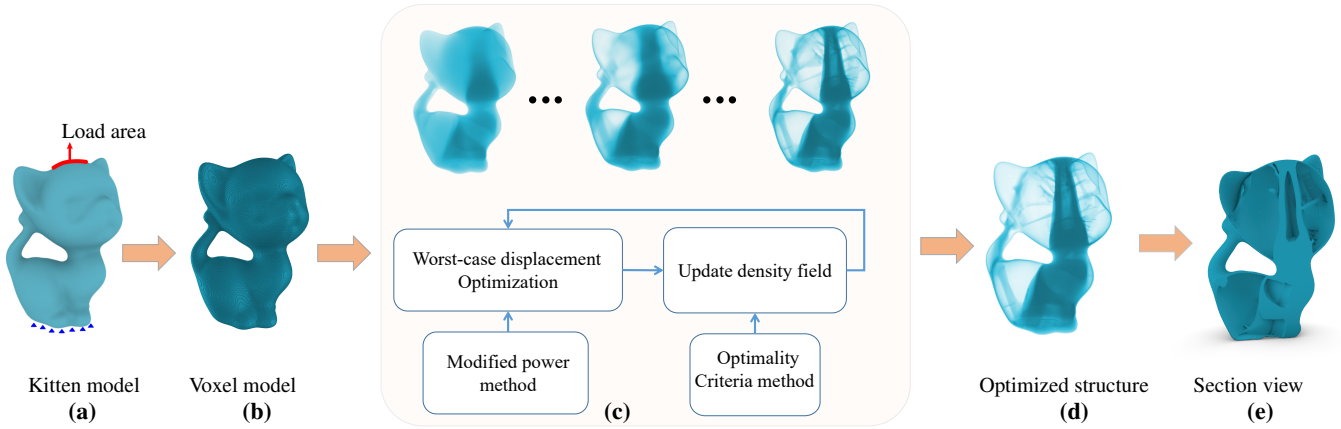


Figure 2: *Pipeline of our algorithm. Given an input model with fixed boundaries and load area (a), we first voxelize it into* $334 \times 510 \times 302$ *elements (b) and then iteratively solve the worst-case displacement field and update the density field (c). We show the final optimized density field in (d) and one section of the resulting structure in (e).*

- The load direction of $\mathbf{f}_i$ is a specified direction $\mathbf{d}_i$. Then, we have $\mathbf{d}_i^1 \cdot \mathbf{f}_i = 0$ and $\mathbf{d}_i^2 \cdot \mathbf{f}_i = 0$, where $\mathbf{d}_i^1$ and $\mathbf{d}_i^2$ are two unit vectors perpendicular to $\mathbf{d}_i$. In general, $\mathbf{d}_i$ is the normal of the surface node since the model often bears pressure. We determine $\mathbf{d}_i$ as the normal at the point computed by projecting the node onto the input surface.
- There is no constraint for $\mathbf{f}_i$.

When no boundary region of Ω is fixed, we further constrain $\mathbf{f}$ to be balanced: $R^\top \mathbf{f} = \mathbf{0}$, where $R^\top \mathbf{f}$ computes the total force and torque, which are zero for a balanced load. The method for computing R is shown in [ZPZ13, SBR*15]. We assemble all these constraints into a linear equation $N\mathbf{f} = \mathbf{0}$.

Computing E_{worst} As the load $\mathbf{f}$ varies, the compliance changes. Consequently, the maximum compliance (denoted as E_{worst}) under load-uncertainty should be minimized. Given a fixed $\boldsymbol{\rho}$, E_{worst} is the objective function of the following optimization problem:

$$\begin{aligned} \max_{\mathbf{f}} \quad & \mathbf{f}^\top K^{-1} \mathbf{f}, \\ \text{s.t.} \quad & \|\mathbf{f}\|_2 = 1, \\ & N\mathbf{f} = \mathbf{0}. \end{aligned} \tag{1}$$

Here, $\|\mathbf{f}\|_2 = 1$ indicates that the load $\mathbf{f}$ has a bounded magnitude and K is the stiffness matrix. Based on the SIMP method [Ben89], $K = \sum_i \rho_i^p K_i$, where K_i is the element stiffness matrix of i^{th} element and p is a constant (usually set to 3). According to the theory of elastic mechanics and finite element method, the balance equation is $K\mathbf{u} = \mathbf{f}$, where $\mathbf{u}$ is the discretized displacement vector.

Reformulating E_{worst} Let $F = \ker N$ include all uncertain loads satisfying $N\mathbf{f} = \mathbf{0}$ and H be a set of standard orthogonal bases of F. Each colume of H is a normalized uncertain load. Then, any $\mathbf{f} \in F$ can be represented as $H\boldsymbol{\phi}$, where $\boldsymbol{\phi}$ is a vector. Consequently, we reformulate (1) as follows:

$$\begin{aligned} \max_{\boldsymbol{\phi}} \quad & \boldsymbol{\phi}^\top (H^\top K^{-1} H) \boldsymbol{\phi}, \\ \text{s.t.} \quad & \|\boldsymbol{\phi}\|_2 = 1. \end{aligned} \tag{2}$$

$\boldsymbol{\phi}$ becomes the new optimization variable. This formulation is equivalent to finding the largest eigenvalue of $H^\top K^{-1} H$ [TNKK11].

Formulation Our goal is to perform a worst-case topology optimization of Ω with uncertain load conditions and bounded volume constraints. The optimization problem is formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{\rho}} \quad & E_{\text{worst}}, \\ \text{s.t.} \quad & \text{Volume}(\boldsymbol{\rho}) \le \varepsilon_V, \\ & \rho_e \in [\rho_{\min}, 1], \forall e, \end{aligned} \tag{3}$$

where ε_V is the specified volume fraction threshold and $\rho_{\min}$ is the minimum density (0.001 in the experiments).

Optimization overview The Optimality Criteria method [BS98] is used to solve (3). In each iteration, the following three steps are performed in turns:

1. Compute E_{worst} and the corresponding load vector $\mathbf{f}_{\text{worst}}$ and displacement vector $\mathbf{u}_{\text{worst}}$ by solving (2) (Section 3.2).
2. Perform the sensitivity analysis, i.e., compute the gradient $\frac{\partial E_{\text{worst}}}{\partial \rho_i}$ (Section 3.3).
3. Update the density $\boldsymbol{\rho}$ using the derivatives $\frac{\partial E_{\text{worst}}}{\partial \rho_i}$ (Section 3.4).

3.2. Modified power method

Revisiting [TNKK11] To solve (2), the method of [TNKK11] first determines H, then treats $H^\top K^{-1} H$ as a whole matrix (denoted as A), and finally computes the maximum eigenvalue of A. Although the detailed method for finding the maximum eigenvalue is not discussed by [TNKK11], the power method can be directly used:

1. Given an arbitrary vector $\boldsymbol{\phi}^{[0]}$ with $\|\boldsymbol{\phi}^{[0]}\|_2 = 1$. Set the iteration number $i = 0$.
2. Compute $\boldsymbol{\phi}^{[i+1]} \leftarrow A\boldsymbol{\phi}^{[i]}$.
3. Normalize $\boldsymbol{\phi}^{[i+1]} \leftarrow \boldsymbol{\phi}^{[i+1]} / \|\boldsymbol{\phi}^{[i+1]}\|_2$.
4. If $\|\boldsymbol{\phi}^{[i+1]} - \boldsymbol{\phi}^{[i]}\|_2 < \varepsilon$ (ε is a small positive threshold), output $\boldsymbol{\phi}^{[i+1]}$ as the eigenvector; otherwise, $i \leftarrow i + 1$ and go to Step 2.

The most time-consuming step in [TNKK11] is the construction of A as many linear systems need to be solved for computing $K^{-1}H$ despite the use of a multigrid solver. Specifically, the number of linear systems is equal to the number of the columns in H, which is usually great in the large-scale topology optimization problem. However, H is a dense matrix when there are no boundary conditions. As shown in Table 1, H and A are not capable of being stored in a desktop computer if the number of load nodes exceeds 10^5.

Our modification To overcome the explicit construction of H and A, we introduce a variable $\mathbf{y}^{[i]} \triangleq H\boldsymbol{\phi}^{[i]}$. Then, $\boldsymbol{\phi}^{[i+1]} \leftarrow A\boldsymbol{\phi}^{[i]}$ becomes $\mathbf{y}^{[i+1]} = (HH^\top)K^{-1}\mathbf{y}^{[i]}$, where we multiply H on both sides. Since $P \triangleq HH^\top = I - N^\top N$, we omit the computation and storage of H. To further simplify computation, the following proposition is proposed.

PROPOSITION 1. *PK^{-1} and $H^\top K^{-1}H$ have the same eigenvalues.*

The proof is provided in Appendix A. Then, we use the power method to compute the maximum eigenvalue of PK^{-1}:

1. Set $i = 0$ and generate a random normalized vector $\mathbf{y}^{[0]}$.
2. Compute $\mathbf{x}^{[i+1]} \leftarrow K^{-1}\mathbf{y}^{[i]}$.
3. Update $\mathbf{y}^{[i+1]} \leftarrow P\mathbf{x}^{[i+1]}$ and then normalize $\mathbf{y}^{[i+1]}$.
4. If $\|\mathbf{y}^{[i+1]} - \mathbf{y}^{[i]}\|_2 < \varepsilon$, output $\mathbf{y}^{[i+1]}$ as $\mathbf{f}_{\text{worst}}$ and $\mathbf{x}^{[i+1]}$ as $\mathbf{u}_{\text{worst}}$; otherwise, set $i = i+1$ and return to step 2.

Multigrid solver In each iteration of our modified method, a linear system $K\mathbf{x} = \mathbf{y}^{[i]}$ needs to be solved to obtain $\mathbf{x}^{[i]}$. Fortunately, the memory-efficient geometric multigrid solver [WDW15] can be utilized to solve this linear system. The main idea of multigrid is to solve a coarse problem by a global correction of the fine grid solution approximation from time to time to accelerate the convergence. The fine grid is first obtained by voxelizing the model via CUDA Voxelizer [Bae17]; then, we construct the coarse grid by quadrupling or doubling the length of elements until it covers at least one element on the previous level. The coarse grid operators are obtained by Galerkin-based coarsening with trilinear interpolation operator and its transpose as the restriction operator. We use V-cycles with one pre- and one post-smoothing eight-color Gauss-Seidel step. This solver is efficient on both memory and time. However, since the power method usually converges after dozens of iterations, we still need to solve many linear systems causing high computational cost.

Acceleration Our key idea for efficiency improvement is to solve $K\mathbf{x} = \mathbf{y}^{[i]}$ inaccurately. To obtain the inaccurate solution, we perform one V-cycle in the multigrid solver in practice:

$$\mathbf{x}^{[i+1]} = \mathbf{x}^{[i]} + V^{-1}(\mathbf{r}^{[i]}), \tag{4}$$

where $\mathbf{r}^{[i]} = \mathbf{y}^{[i]} - K\mathbf{x}^{[i]}$ and $V^{-1}(\cdot)$ indicates the one V-cycle operator. We use $\mathbf{x}^{[i+1]}$ as an approximation of $K^{-1}\mathbf{y}^{[i]}$ to update $\mathbf{y}^{[i+1]} \leftarrow P\mathbf{x}^{[i+1]}$. Besides, $\mathbf{x}^{[i+1]}$ is reserved as the starting point of the V-cycle operator in the next iteration. The pseudocode of our modified power method is shown in Alg. 1. We terminate the algorithm when $\|\mathbf{y}^{[i+1]} - \mathbf{y}^{[i]}\|_2 < \varepsilon$ and the residual $r_{\text{rel}} = \|\mathbf{r}^{[i]}\|_2 < \varepsilon_r$ ($\varepsilon_r = 0.01$ in the experiments).

Although the number of iterations of the power method increases due to the inaccurate solutions, the running time of each iteration decreases significantly, resulting in a more efficient algorithm (Figure 3). Besides, we find in practice that our accelerated algorithm always converges to the same maximum eigenvalue as the method using accurate solutions for $K\mathbf{x} = \mathbf{y}^{[i]}$ (see Figure 3).

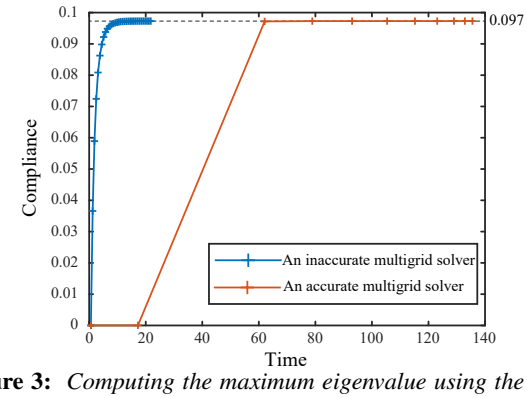


Figure 3: *Computing the maximum eigenvalue using the power method with the accurate or inaccurate solvers for $K\mathbf{x} = \mathbf{y}^{[i]}$. The Kitten model in Figure 2 is used. We plot the compliance vs. running time in seconds.*

ALGORITHM 1: Modified power method

Input: The matrices K, P
Output: $\mathbf{u}_{\text{worst}}$ and $\mathbf{f}_{\text{worst}}$
Generate random normalized $\mathbf{y}^{[0]}$;
$\mathbf{x}^{[0]} \leftarrow \mathbf{0}$; $i \leftarrow 0$; $r_{\text{rel}} \leftarrow 1$;
while $\|\mathbf{y}^{[i]} - \mathbf{y}^{[i-1]}\|_2 \geq \varepsilon$ *or* $r_{rel} > \varepsilon_r$ **do**
  $\mathbf{r}^{[i]} \leftarrow \mathbf{y}^{[i]} - K\mathbf{x}^{[i]}$;
  $r_{\text{rel}} \leftarrow \|\mathbf{r}^{[i]}\|_2$; // r_{rel} is the norm of the residual vector $\mathbf{r}^{[i]}$.
  $\mathbf{x}^{[i+1]} \leftarrow \mathbf{x}^{[i]} + V^{-1}(\mathbf{r}^{[i]})$;
  $\mathbf{y}^{[i+1]} \leftarrow P\mathbf{x}^{[i+1]}$;
  $\mathbf{y}^{[i+1]} \leftarrow \mathbf{y}^{[i+1]} / \|\mathbf{y}^{[i+1]}\|_2$;
  $i \leftarrow i+1$;
end
Output $\mathbf{u}_{\text{worst}} \leftarrow \mathbf{x}^{[i]}$ and $\mathbf{f}_{\text{worst}} \leftarrow \mathbf{y}^{[i]}$;

Theoretical analysis Since we only use the approximation of $K^{-1}\mathbf{y}^{[i]}$, the output may not be the maximum eigenvalue, although it was not found in experiments. Hence, the following theoretical analysis is provided. We denote the eigenvalues of PK^{-1} as $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n \geq 0$ and the eigenvectors as $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$, which form the orthogonal bases of $\mathbb{R}^n$. Then, $\mathbf{y}^{[i]}$ can be represented as $\mathbf{y}^{[i]} = \sum_j \beta_j^{[i]}\mathbf{v}_j$. We introduce $\mathbf{e}^{[i]} = P(\mathbf{x}^{[i+1]} - K^{-1}\mathbf{y}^{[i]})$, that is represented by $\mathbf{e}^{[i]} = \sum_j \varepsilon_j^{[i]}\mathbf{v}_j$. We propose a proposition as follows:

PROPOSITION 2. *If $\mathbf{y}^{[i]}$ converges to $\mathbf{y}$ using our modified power method, and there exist $N \in \mathbb{N}^+$ and $C > 0$ such that $|\beta_j^{[N]}/\beta_1^{[N]}| \leq C$ and $\delta^N = \max_j \sup_{k \geq N} |\varepsilon_j^{[k]}| \leq \frac{C(\lambda_1 - \lambda_j)}{1+C} \frac{1}{\sqrt{1+C^2(n-1)}}$ for $j = 2, \ldots, n$, then $\mathbf{y}$ is the largest eigenvector of PK^{-1}.*

The proof is provided in Appendix B. The conditions in Prop. 2 may be strict and the numerical experiments demonstrate the effectiveness and convergence of our method (Figures 4 and 11).

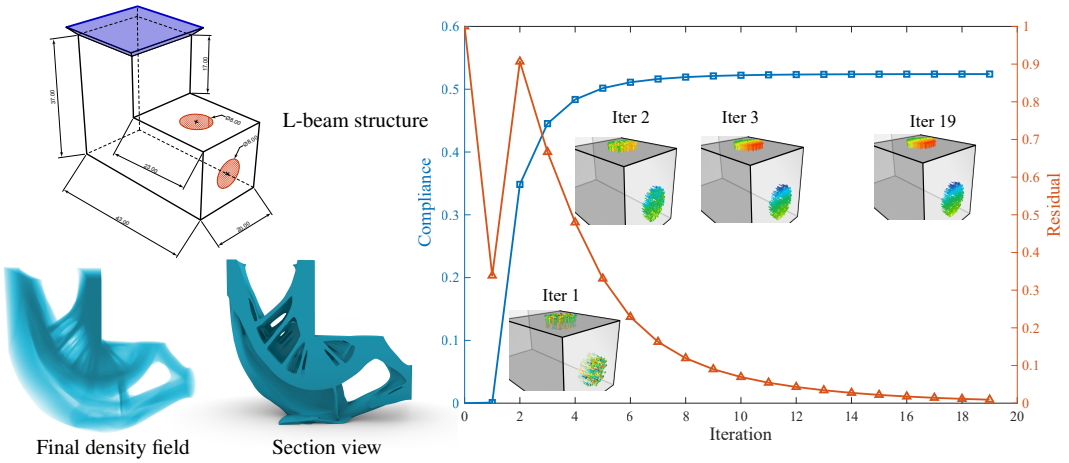


Figure 4: *The input is an L-beam model voxelized by* $431 \times 200 \times 370$ *elements. The loads are applied on the red region and the top surface is fixed. Given a density field, we first run Alg. 1 and then plot the compliance vs. the number of iterations (blue line) and the residual* r_{rel} *vs. the number of iterations (orange line). As the iteration proceeds, the compliance converges, and the residual decreases to be less than the threshold* $\varepsilon_r = 0.01$*. The nodal forces at each iteration of Alg. 1 are shown in a colormap. A color closer to red indicates a larger magnitude, and vice versa. The result shows that the worst case occurs when the nodal forces are the largest at the upper surface and the smaller at the side.*

Other large-scale eigenvalue solvers The goal of Alg. 1 is to find the maximum eigenvalue and the corresponding eigenvector of the matrix PK^{-1}. Several other solvers for computing large-scale eigenvalue problems are also available to solve this problem, such as ARPACK [GS97]. The algorithm behind ARPACK is the Implicitly Restarted Arnoldi Method (IRAM) [Leh01], which searches for the eigenvector in the Krylov subspace whose dimension is larger than the number of wanted eigenvectors. This Krylov subspace is spanned by the Arnoldi vectors and updated in every loop of IRAM, each of these vectors involves a matrix-vector multiplication which entails solving the stiffness equation by our multigrid solver. Thus, it leads to high computation costs due to the larger dimension of the subspace and loop times. Besides, these Arnoldi vectors use much memory in large-scale problems.

3.3. Sensitivity analysis

Since $\|\mathbf{f}_{\text{worst}}\|_2 = \|K\mathbf{u}_{\text{worst}}\|_2 = 1$ and $N\mathbf{f}_{\text{worst}} = NK\mathbf{u}_{\text{worst}} = \mathbf{0}$, we introduce two adjoint variables μ and Λ to evaluate $\frac{\partial E_{\text{worst}}}{\partial \rho_i}$:

$$\frac{\partial E_{\text{worst}}}{\partial \rho_i} = \frac{\partial}{\partial \rho_i}\mathbf{u}_{\text{worst}}^\top K\mathbf{u}_{\text{worst}}, \tag{5}$$

$$= \frac{\partial}{\partial \rho_i}\left[\mathbf{u}_{\text{worst}}^\top K\mathbf{u}_{\text{worst}} - \mu\left(\mathbf{u}_{\text{worst}}^\top K^\top K\mathbf{u}_{\text{worst}} - 1\right) - \Lambda NK\mathbf{u}_{\text{worst}}\right],$$

$$= \left(2\mathbf{u}_{\text{worst}}^\top K - 2\mu\mathbf{u}_{\text{worst}}^\top K^\top K - \Lambda^\top NK\right)\frac{\partial \mathbf{u}_{\text{worst}}}{\partial \rho_i} + \mathbf{u}_{\text{worst}}^\top \frac{\partial K}{\partial \rho_i}\mathbf{u}_{\text{worst}} - 2\mu\mathbf{u}_{\text{worst}}^\top \frac{\partial K^\top}{\partial \rho_i}K\mathbf{u}_{\text{worst}} - \Lambda^\top N\frac{\partial K}{\partial \rho_i}\mathbf{u}_{\text{worst}}.$$

We choose proper μ and Λ to avoid computing $\partial \mathbf{u}_{\text{worst}}/\partial \rho_i$, i.e., satisfy the following equation:

$$2K\mathbf{u}_{\text{worst}} - 2\mu K^\top K\mathbf{u}_{\text{worst}} - KN^\top \Lambda = \mathbf{0}. \tag{6}$$

By multiplying $\mathbf{u}_{\text{worst}}^\top$ on the both sides of (6) and using the conditions $NK\mathbf{u}_{\text{worst}} = \mathbf{0}$ and $\|K\mathbf{u}_{\text{worst}}\|_2^2 = 1$, we get

$$\mu = \mathbf{u}_{\text{worst}}^\top K\mathbf{u}_{\text{worst}},$$
$$N^\top \Lambda = 2(\mathbf{u}_{\text{worst}} - K\mathbf{u}_{\text{worst}}).$$

After putting the computed μ and $N^\top \Lambda$ into (5), we have

$$\frac{\partial E_{\text{worst}}}{\partial \rho_i} = -\mathbf{u}_{\text{worst}}^\top \frac{\partial K}{\partial \rho_i}\mathbf{u}_{\text{worst}}. \tag{7}$$

3.4. Algorithm details

Updating densities We follow the classical SIMP model to update densities. To make sure the solution exists and avoid checkerboard patterns, we filter the sensitivities, i.e., $\frac{\partial E_{\text{worst}}}{\partial \rho_i}$ as follows [Sig94]:

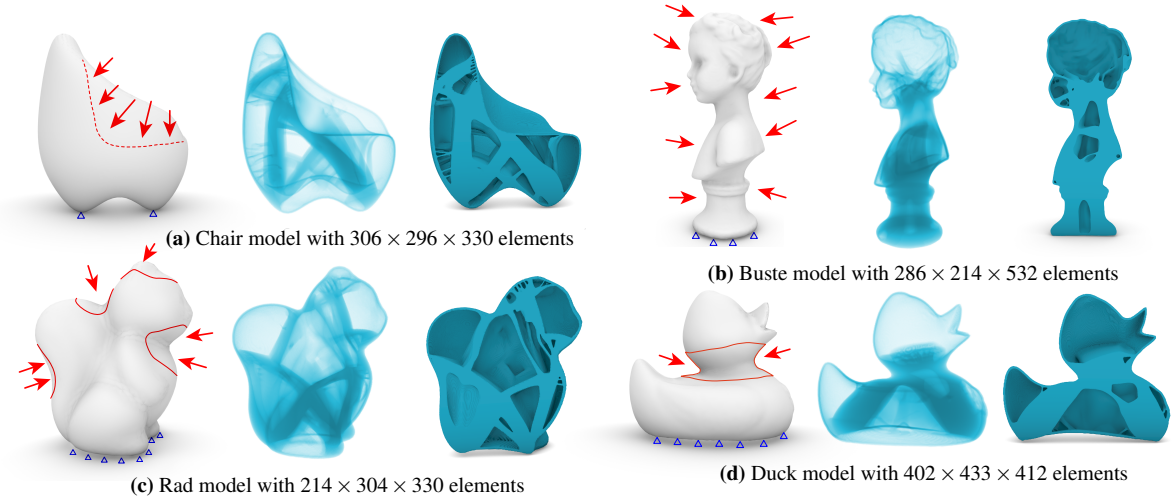
$$g_i = \frac{\sum_j w(\|x_i - x_j\|_2/r)\rho_j \frac{\partial E}{\partial \rho_j}}{\rho_i \sum_j w(\|x_i - x_j\|_2/r)},$$

where x_i is the center of i^{th} voxel, r is a positive radius, and $w(x)$ is a locally supported function [WDW15]:

$$w(x) = \begin{cases} 1 - 6x^2 + 8x^3 - 3x^4, x < 1, \\ 0, x \geq 1. \end{cases}$$

If $\|x_i - x_j\|_2 \geq r$, $w(\|x_i - x_j\|_2/r) = 0$ indicating that j^{th} voxel has no effect on i^{th} voxel in the filtering process. With the filtered sensitivities, we update the densities by following [BS98]:

$$\rho_i \leftarrow \text{Clamp}\left(\text{Clamp}\left(\left(\frac{g_i}{\bar{g}}\right)^\eta \rho_i, \rho_i - \Delta\rho, \rho_i + \Delta\rho\right), \rho_{\min}, 1.0\right), \tag{8}$$



(a) Chair model with $306 \times 296 \times 330$ elements

(b) Buste model with $286 \times 214 \times 532$ elements

(c) Rad model with $214 \times 304 \times 330$ elements

(d) Duck model with $402 \times 433 \times 412$ elements

Figure 5: *Optimization results of different models. For each model, we show the design domain and boundary conditions (left), the optimized material distribution (middle), and the corresponding section view (right).*

where the threshold $\Delta\rho$ is to limit the density change in one iteration, $\rho_{\min}$ is a lower-bound of density to avoid numerical instability, η is a damping factor (typically 0.5), and $\bar{g}$ is a multiplier determined by the binary search in each iteration to make the volume constraint satisfied after all densities are updated. (8) is the well-known Optimal Criteria method. When the algorithm converges, the KKT condition of the problem (3) is satisfied. Besides, a step limits $\Delta\rho$ is introduced to stabilize the optimization.

Stopping criterion We represent the computed worst-case compliance in the i^{th} iteration as c_i. When the following condition is satisfied

$$((c_i + c_{i-1}) - (c_{i-2} + c_{i-3}))\,/\,(c_{i-2} + c_{i-3}) < \varepsilon_c,$$

for continuous n_c times, we regard the iteration converged. We set $\varepsilon_c = 5 \times 10^{-3}$ and $n_c = 2$ in all the results shown in this paper. Also, when solving $\mathbf{u}_{\text{worst}}$, the parameter ε is set to 10^{-4}, and ε_r is set to 10^{-2} in Alg. 1.

Shell The shell structure is introduced into the optimization of the interior structure, which ensures the characteristics of the model and the stability of optimization process. When updating the density, elements in the shell structures are set as solid. The shell thickness is user-defined in our algorithm.

4. Experiments

We have computed the optimal structures for various 3D objects in Figure 5. Models are selected from different categories consisting of a wide range of typical features and realistic boundary conditions. All experiments are executed on a desktop PC with a 3.6 GHz Intel Core i9-9900K, 32GB of memory, and an NVIDIA GTX 1080Ti graphics card with 11 GB graphics card RAM size. Table 1 summarizes the statistics of the time and memory cost under different parameters (volume fraction, resolutions, *et al.*). In summary, our algorithm handles 24 million elements while the maximum memory is less than 8Gb, and the time is controlled within 3 hours. In practice, the consumed time increases with model detail (e.g., Figures 2 and 5 (b)) and resolution.

4.1. External forces

The force direction constraint in Section 3.1 restricts the worst-case load along the direction of $\mathbf{n}_i$ or $-\mathbf{n}_i$, which is bidirectional. However, objects usually bear pressure, and tensile on the objects almost never happens (Figure 6). When the force is unidirectional, the worst case should be solved by the following problem:

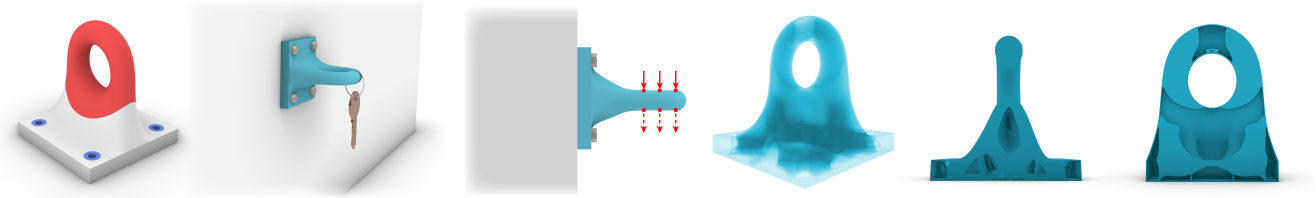
$$\begin{aligned} \max_{\mathbf{f}} \quad & \mathbf{f}^\top K^{-1}\mathbf{f}, \\ \text{s.t.} \quad & \|\mathbf{f}\|_2 = 1, \\ & N\mathbf{f} = \mathbf{0}, \\ & \mathbf{V}^\top \mathbf{f} \geq 0, \end{aligned} \tag{9}$$

where each column of $\mathbf{V}$ is a direction of the force, e.g., the inward normal. Here, we assume columns of $\mathbf{V}$ are a set of the orthogonal vectors in $\ker N$, i.e., $\mathbf{V}^\top\mathbf{V} = I$ and $N\mathbf{V} = \mathbf{0}$. Since optimization problem (9) cannot be reformulated as a maximum eigenvalue problem, it is harder to solve. We give the following proposition:

PROPOSITION 3. *The maximum compliances of the optimization model (1) and (9) are denoted as c_1 and c_2, respectively. Then, we have $\frac{1}{2}c_1 \leq c_2 \leq c_1$.*

The proof of Prop. 3 is given in Appendix C. Therefore, our algorithm can deal with unidirectional forces (Figure 6 (b)).

It also should be noted that our algorithm can handle the worst-case problem under force from any direction. The external forces are user-defined according to different situations. Figure 7 shows the optimized structures of the foot model. In the presence of friction force, the force in the worst-case problem is not in the normal direction. The final structure B is more stable for dealing with friction force conditions over structure A.



(a) Design model and boundary conditions **(b)** Optimized structure and its section views

Figure 6: *The hook model is voxelized into* $406 \times 406 \times 433$ *elements. The hook has external forces at the ring and is fixed at four points on the bottom. When an object is suspended from the hook, the force is only along the direction* $-\mathbf{n}_i$*. Thus, the force is unidirectional at each node.*

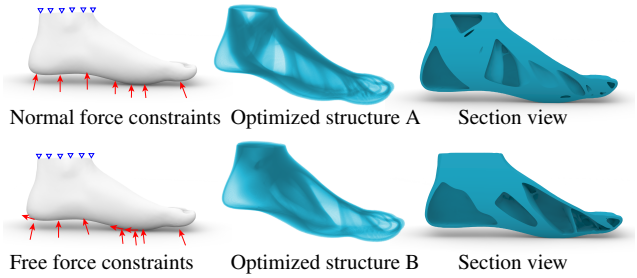


Figure 7: *Comparisons between the normal force constraints and the free force constraints on the Foot model containing* $146 \times 432 \times 180$ *voxel elements.*

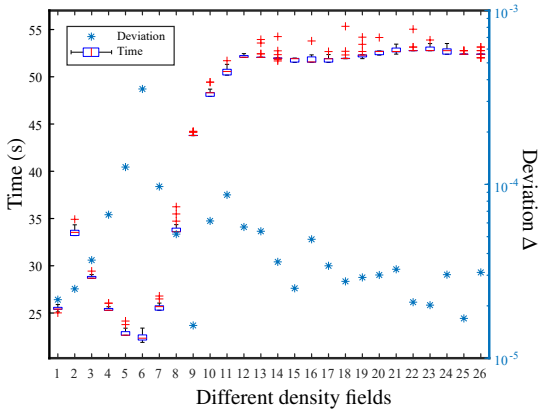


Figure 8: *Our modified power method with different random normalized force* $\mathbf{y}^{[0]}$ *on the Duck model in Figure 5. The x-axis indicates the density fields of different iterations during the optimization. For each density field, we show time statistics by the box-plot.*

4.2. Initial forces

The random normalized force $\mathbf{y}^{[0]}$ is taken as the input of the modified power method (Alg. 1). In Figure 8, given a density field, we test 20 random normalized forces $\mathbf{y}^{[0]}$. To measure the stability of Alg. 1, we define a load deviation metric as follows:

$$\Delta = \max_{1 \le j \le 20} \|\mathbf{f}^j_{\text{worst}} - \bar{\mathbf{f}}\| / \|\bar{\mathbf{f}}\|, \tag{10}$$

where $\bar{\mathbf{f}} = \frac{1}{20} \sum_{i=1}^{20} \mathbf{f}^i_{\text{worst}}$ is the average worst-case load. Besides, 26 different density fields are tested. Judging from Figure 8, we find that our algorithm converges to almost the same load as each deviation metric is less than 10^{-3} and the variation of the time cost caused by the random initial forces is small.

4.3. Without fixed boundaries

In daily life, any movable object may bear loads from any direction on its entire surface. In the work of [WDW15], they consider touch/grasp interaction at arbitrary locations via uniformly distributed forces and set the bottom zone of the object as fixed boundary. Since the bottom zone may not be fixed either, this manually enforced fixed boundary and its location affects the final results. One way to solve this problem is to remove the fixed boundary, yet it makes the stiffness matrix singular, and generally, there is no solution of the FEM equation in such a case. To overcome this problem, we can project the uniformly distributed force to balanced load space [ZPZ13]. Then, the FEM equation will have the solution, and the general topology optimization method will work. However, it ignores the fragile load mode of the object. Instead, our algorithm iteratively optimizes the worst-case load; we think it is more suitable to perform topology optimization via our method.

We remove the fixed boundary on the Foot model in Figure 7 and then optimized it. Final results are shown in Figure 10. Compared with the results in the fixed-boundary setting, the material from the non-fixed boundary case tends to distribute around the surface to resist random forces. Figure 9 shows the results of the bunny model with no fixed boundary at different volume fractions. External forces are applied on the whole bunny surface. The optimized results show that the material was preferentially distributed in thin parts such as ears, followed by head.

4.4. Comparisons

To verify the accuracy of our method, we compare the worst-case load $\mathbf{f}_{\text{our}}$ computed by our method and the load $\mathbf{f}_{\text{cmp}}$ computed by direct eigenvalue decomposition of $H^\top K^{-1} H$. We define an error as follows:

$$err = \frac{\|\mathbf{f}_{\text{cmp}} - \mathbf{f}_{\text{our}}\|_2}{\|\mathbf{f}_{\text{cmp}}\|_2}.$$

A cuboid model in Figure 11 is used. The maximum error is less than 10^{-2}, indicating that $\mathbf{f}_{\text{our}}$ approaches $\mathbf{f}_{\text{cmp}}$. With the proof of Prop. 2 and the numerical experiments, our algorithm is confirmed to be feasible and convergent in practice.

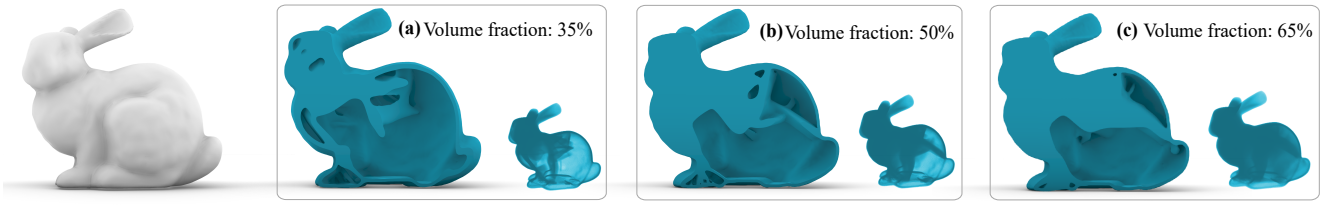


Figure 9: *Different volume fractions for the Bunny model with* $288 \times 284 \times 222$ *elements. The whole surface is taken as the load area, and no fixed boundary is given.*

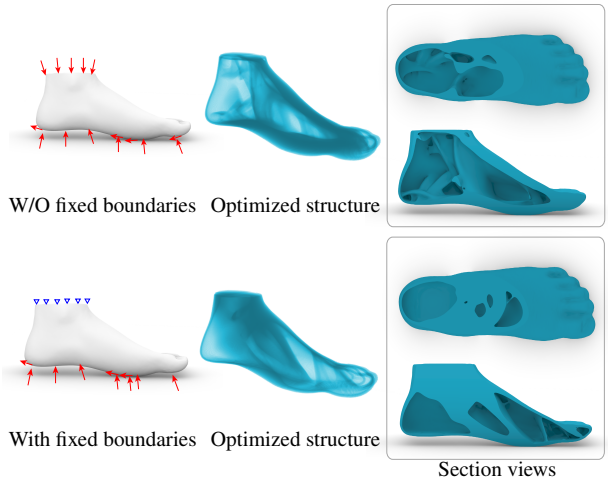


Figure 10: *Large-scale worst-case topology optimization without fixed boundaries (left) and with fixed boundaries (right) on the Foot model in Figure 7.*

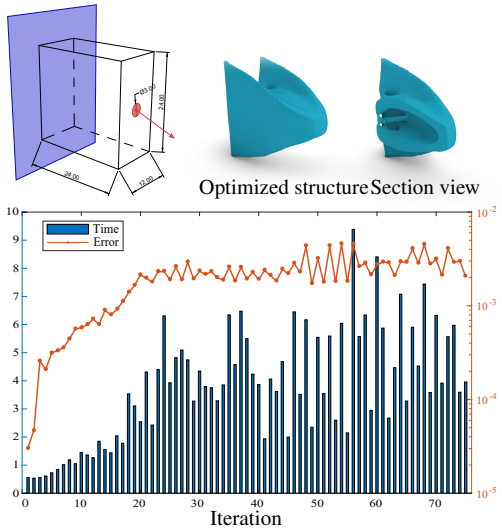
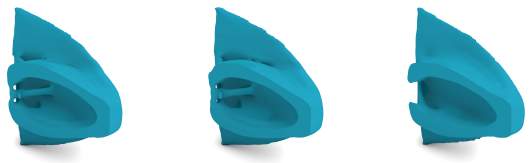


Figure 11: *Validating the accuracy of our method using the Cuboid model with* $100 \times 50 \times 100$ *elements. In each iteration of our topology optimization algorithm, we compute* $\mathbf{f}_{our}$ *and* $\mathbf{f}_{cmp}$ *for determining the error. We plot the error vs. the number of iterations (orange curve). The time (in seconds) for computing* $\mathbf{f}_{our}$ *using our modified power method is shown using bars.*



PK^{-1}+inaccurate MG	PK^{-1}+accurate MG	$H^\top K^{-1} H$ + accurate MG
$1.253 \times 10^{-3} / 1.214 \times 10^{-3}$	$1.283 \times 10^{-3} / 1.234 \times 10^{-3}$	$1.064 \times 10^{-3} / 1.036 \times 10^{-3}$

Figure 12: *Final results with different methods using the Cuboid model in Figure 11. The values in the second row are the largest two eigenvalues of the aggregated systems of the final results.*

Our goal is to explore the optimal solution of the worst-case problem formulated in (1). We summarize three methods to solve this problem in Section 3.2. The essential problem is to solve the maximum eigenvalue of $H^\top K^{-1} H$. The first method is to evaluate the matrix $H^\top K^{-1} H$ explicitly. The second method is to modify $H^\top K^{-1} H$ into PK^{-1} equivalently. Both two method compute the maximum eigenvalue with an accurate multigrid solver. The third methods is our innovative approach. Taking the Cuboid in Figure 11 as an example, the optimized structures by the above three methods are shown in Figure 12 which are very similar. The values in the second row are the largest two eigenvalues of the three methods. The computational efficiency of these three algorithms is tested and shown in Figure 13. The difference between the first and second method is to modify $H^\top K^{-1} H$ into PK^{-1}. However, 99.36% of the time is saved. Our method is further faster than the second method. The time reduction is about 65%. Our calculation time is 0.23% of that of the first method. Thus, our algorithm significantly improves the computational efficiency while ensuring the accuracy of the calculation.

We change the area of the load region for the Cuboid in Figure 11 with a fixed resolution and a given density field. Then, the time costs of the three algorithms for computing the maximum eigenvalue are compared in Figure 14. As the load area becomes larger, the time consumption of the first method increases rapidly, while our time consumption changes slightly. Our modified power method takes the least time and shows particular advantages in dealing with large areas of load regions.

Model	$\varepsilon_V(\%)$	#Elements	#Load nodes	r	$\Delta\rho$	Memory (MB)	Time (m)	t_{apm} (m) / Speedup	t_{agg} (m) / Speedup
Figure 1	35	14.8×10^6	69934	2	0.06	4614	50	136 / 2.72	-
Figure 2	35	16.7×10^6	8730	3	0.06	5253	112	652 / 5.82	-
Figure 4	40	24.0×10^6	10094	2	0.02	7416	63	247 / 3.92	-
Figure 5 (a)	35	11.5×10^6	48413	2	0.06	3623	83	324 / 3.90	-
Figure 5 (b)	50	9.3×10^6	262772	4	0.06	3222	162	769 / 4.75	-
Figure 5 (c)	50	8.8×10^6	35840	2	0.06	2806	56	123 / 2.20	-
Figure 5 (d)	50	10.6×10^6	44863	3	0.06	3356	28	49 / 1.75	-
Figure 6	65	18.9×10^6	159873	2	0.06	5976	35	53 / 1.51	-
Figure 7 - Top	50	4.5×10^6	55434	4	0.02	1462	26	113 / 4.35	-
Figure 7 - Bottom	50	4.5×10^6	55434	4	0.02	1462	72	348 / 4.83	-
Figure 10 - Top	50	4.5×10^6	67699	4	0.02	2097	89	212 / 2.38	-
Figure 9 (a)	35	4.8×10^6	168790	4	0.02	2254	54	166 / 3.07	-
Figure 9 (b)	50	4.8×10^6	168790	4	0.02	2254	33	102 / 3.09	-
Figure 9 (c)	65	4.8×10^6	168790	4	0.02	2254	24	69 / 2.88	-
Figure 11	20	0.5×10^6	121	2	0.06	166	11	12 / 1.09	75 / 6.82

Table 1: *Optimization statistics. $\varepsilon_V(\%)$ is the volume fraction. #Elements means the number of FEM elements. #Load nodes is the number of the load nodes. r is the filter radius and is measured in the unit of element length. $\Delta\rho$ is the limitation of the change in density. Memory is the graphics card memory cost. t_{apm} and t_{agg} are the time cost of PK^{-1} + an accurate multgrid solver and $H^\top K^{-1}H$ + an accurate mutlgrid solver. "-" means the optimization did not finish in two weeks due to high computation or memory cost.*

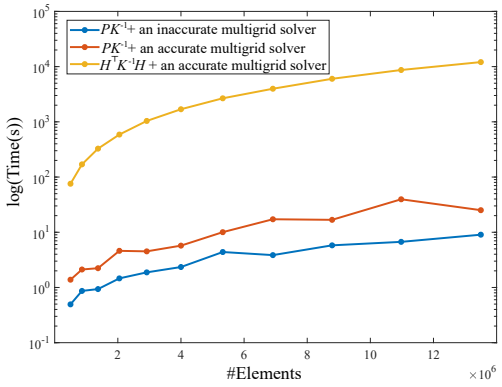


Figure 13: *Comparison of the time taken by three methods for computing the maximum eigenvalue of $H^\top K^{-1}H$. The abscissa represents the resolution, i.e., the number of elements (#Elements), and the ordinate represents the logarithmic computational time in seconds. For one resolution, the three methods run on the same model with a given density field and the same load region.*

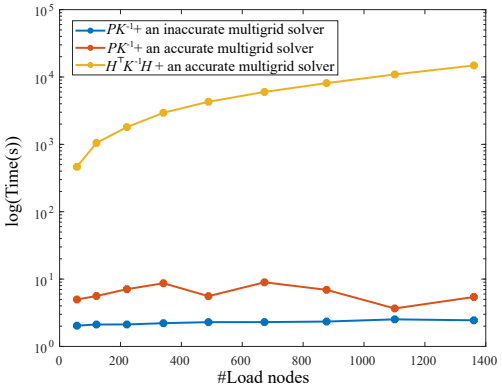


Figure 14: *Comparison of the time taken by three methods for computing the maximum eigenvalue of $H^\top K^{-1}H$. The abscissa represents the number of load nodes, and the ordinate represents the logarithmic computational time in seconds.*

5. Conclusion

We propose an approach for large-scale worst-case topology optimization. Given a 3D model with specified load regions and boundary conditions, our method optimizes a structure that minimizes the maximum compliance produced by the normalized external loads under a certain volume fraction. By transforming the worst-case problem into a maximum eigenvalue problem and replacing $H^\top K^{-1}H$ with PK^{-1}, we solve the maximum eigenvalue of PK^{-1} with an inaccurate multigrid solver. The accuracy of the proposed algorithm is verified by numerical tests, and it is much faster than existing algorithms.

Limitation and future work There are several limitations. First, convergence could slow down when the eigenvalues are very close to each other by the modified power method. This means that we need to spend more time to achieve the final convergence by Alg. 1. More efficient algorithms need to be developed, such as the combination of narrow band strategy. Second, manufacturing constraints [ZFS*02] are not taken into consideration in this work, such as the minimum length scale constraints [ZLWS15], self-supporting constraints [BTX20], etc. For more practical requirements, the fabrication of the optimized structures by our method is also very meaningful. In order to meet the needs of manufacturing, we will put manufacturing constraints into the optimization process in future work. Third, extending our algorithm using other domain representations (e.g., implicit or explicit scalar functions) would be an intriguing direction for future research.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive suggestions and comments. This work is supported by the Provincial Natural Science Foundation of Anhui (2208085QA01), the Fundamental Research Funds for the Central Universities (WK0010000075), the National Natural Science Foundation of China (62025207), and the Major Project of Science and Technology of Anhui Province (202203a05020050).

References

[AKS08] AMIR O., KIRSCH U., SHEINMAN I.: Efficient non-linear reanalysis of skeletal structures using combined approximations. *International journal for numerical methods in engineering 73*, 9 (2008), 1328–1346. 2

[Bae17] BAERT J.: Cuda voxelizer: A gpu-accelerated mesh voxelizer. https://github.com/Forceflow/cuda_voxelizer, 2017. 4

[Ben89] BENDSØE M. P.: Optimal shape design as a material distribution problem. *Structural optimization 1*, 4 (1989), 193–202. 1, 2, 3

[BP01] BORRVALL T., PETERSSON J.: Large-scale topology optimization in 3d using parallel computing. *Computer methods in applied mechanics and engineering 190*, 46-47 (2001), 6201–6229. 1, 2

[BS98] BENDSØE M., SIGMUND O.: *Optimization of structural topology, shape, and materials*. 1998. 2, 3, 5

[BSS08] BRENNER S. C., SCOTT L. R., SCOTT L. R.: *The mathematical theory of finite element methods*, vol. 3. Springer, 2008. 11

[BTN97] BEN-TAL A., NEMIROVSKI A.: Robust truss topology design via semidefinite programming. *SIAM journal on optimization 7*, 4 (1997), 991–1016. 1, 2

[BTX20] BI M., TRAN P., XIE Y. M.: Topology optimization of 3d continuum structures under geometric self-supporting constraint. *Additive Manufacturing 36* (2020), 101422. 9

[CC08] CHERKAEV E., CHERKAEV A.: Minimax optimization problem of structural design. *Computers & Structures 86*, 13-14 (2008), 1426–1435. 1, 2

[CLSK20] CUI Q., LANGLOIS T., SEN P., KIM T.: Fast and robust stochastic structural optimization. In *Computer Graphics Forum* (2020), vol. 39, Wiley Online Library, pp. 385–397. 2

[GS97] GOMES F. M., SORENSEN D. C.: Arpack. 5

[GZZ14] GUO X., ZHANG W., ZHONG W.: Doing Topology Optimization Explicitly and Geometrically—A New Moving Morphable Components Based Framework. *Journal of Applied Mechanics 81*, 8 (2014). 2

[HTK15] HOLMBERG E., THORE C.-J., KLARBRING A.: Worst-case topology optimization of self-weight loaded structures using semi-definite programming. *Structural and Multidisciplinary Optimization 52*, 5 (2015), 915–928. 1, 2

[HWL*20] HU J., WANG S., LI B., LI F., LUO Z., LIU L.: Efficient representation and optimization for tpms-based porous structures. *IEEE Transactions on Visualization and Computer Graphics* (2020). 2

[Leh01] LEHOUCQ R. B.: Implicitly restarted arnoldi methods and subspace iteration. *SIAM Journal on Matrix Analysis and Applications 23*, 2 (2001), 551–562. 5

[LHZ*18] LIU H., HU Y., ZHU B., MATUSIK W., SIFAKIS E.: Narrow-band topology optimization on a sparsely populated grid. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–14. 2

[LLL*21] LI Y., LI X., LI M., ZHU Y., ZHU B., JIANG C.: Lagrangian–eulerian multidensity topology optimization with the material point method. *International Journal for Numerical Methods in Engineering 122*, 14 (2021), 3400–3424. 2

[LNB*10] LE C., NORATO J., BRUNS T., HA C., TORTORELLI D.: Stress-based topology optimization for continua. *Structural and Multidisciplinary Optimization 41*, 4 (2010), 605–620. 2

[LSD*16] LANGLOIS T., SHAMIR A., DROR D., MATUSIK W., LEVIN D. I.: Stochastic structural analysis for context-aware design and fabrication. *ACM Transactions on Graphics (TOG) 35*, 6 (2016), 1–13. 2

[LSS12] LAZAROV B. S., SCHEVENELS M., SIGMUND O.: Topology optimization considering material and geometric uncertainties using stochastic collocation methods. *Structural and Multidisciplinary optimization 46*, 4 (2012), 597–612. 2

[MFHP16] MARTÍNEZ-FRUTOS J., HERRERO-PÉREZ D.: Large-scale robust topology optimization using multi-gpu systems. *Computer Methods in Applied Mechanics and Engineering 311* (2016), 393–414. 2

[MLR*21] MUKHERJEE S., LU D., RAGHAVAN B., BREITKOPF P., DUTTA S., XIAO M., ZHANG W.: Accelerating large-scale topology optimization: State-of-the-art and challenges. *Archives of Computational Methods in Engineering 28*, 7 (2021), 4549–4571. 1, 2

[OFKK99] OHSAKI M., FUJISAWA K., KATOH N., KANNO Y.: Semi-definite programming for topology optimization of trusses under multiple eigenvalue constraints. *Computer Methods in Applied Mechanics and Engineering 180*, 1-2 (1999), 203–217. 2

[Ped00] PEDERSEN N. L.: Maximization of eigenvalues using topology optimization. *Structural and multidisciplinary optimization 20*, 1 (2000), 2–11. 1, 2

[PRZ17] PANETTA J., RAHIMIAN A., ZORIN D.: Worst-case stress relief for microstructures. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–16. 1, 2

[SBR*15] SCHUMACHER C., BICKEL B., RYS J., MARSCHNER S., DARAIO C., GROSS M.: Microstructures to control elasticity in 3d printing. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 1–13. 3

[Sig94] SIGMUND O.: *Design of Material Structures Using Topology Optimization*. PhD thesis, 01 1994. 5

[SW00] SETHIAN J. A., WIEGMANN A.: Structural boundary design via level set and immersed interface methods. *Journal of computational physics 163*, 2 (2000), 489–528. 1, 2

[SZ99] SOKOLOWSKI J., ZOCHOWSKI A.: Topological derivatives for elliptic problems. *Inverse problems 15*, 1 (1999), 123. 1

[SZB18] SCHUMACHER C., ZEHNDER J., BÄCHER M.: Set-in-stone: worst-case optimization of structures weak in tension. *ACM Transactions on Graphics (TOG) 37*, 6 (2018), 1–13. 2

[THK15] THORE C.-J., HOLMBERG E., KLARBRING A.: Large-scale robust topology optimization under load-uncertainty. In *11th World Congress on Structural and Multidisciplinary Optimization* (2015). 1, 2

[TNKK11] TAKEZAWA A., NII S., KITAMURA M., KOGISO N.: Topology optimization for worst load conditions based on the eigenvalue analysis of an aggregated linear system. *Computer Methods in Applied Mechanics and Engineering 200*, 25-28 (2011), 2268–2281. 1, 2, 3, 4

[UMK17] ULU E., MCCANN J., KARA L. B.: Lightweight structure design under force location uncertainty. *ACM Transactions on Graphics (TOG) 36*, 4 (2017), 1–13. 2

[WDW15] WU J., DICK C., WESTERMANN R.: A system for high-resolution topology optimization. *IEEE transactions on visualization and computer graphics 22*, 3 (2015), 1195–1208. 1, 2, 4, 5, 7

[WWG03] WANG M. Y., WANG X., GUO D.: A level set method for structural topology optimization. *Computer methods in applied mechanics and engineering 192*, 1-2 (2003), 227–246. 2

[XLCB15] XU H., LI Y., CHEN Y., BARBIČ J.: Interactive material design using model reduction. *ACM Transactions on Graphics (TOG) 34*, 2 (2015), 1–14. 2

[XS93] XIE Y. M., STEVEN G. P.: A simple evolutionary procedure for structural optimization. *Computers & structures 49*, 5 (1993), 885–896. 2

[YRL*20] YAN X., RAO C., LU L., SHARF A., ZHAO H., CHEN B.: Strong 3d printing by tpms injection. *IEEE Transactions on Visualization and Computer Graphics 26*, 10 (2020), 3037–3050. 2

[ZCW21] ZHAI X., CHEN F., WU J.: Alternating optimization of design and stress for stress-constrained topology optimization. *Structural and Multidisciplinary Optimization 64*, 4 (2021), 2323–2342. 2

[ZFS*02] ZHOU M., FLEURY R., SHYY Y.-K., THOMAS H., BRENNAN J.: Progress in topology optimization with manufacturing constraints. In *9th AIAA/ISSMO Symposium on multidisciplinary analysis and optimization* (2002), p. 5614. 9

[ZLWS15] ZHOU M., LAZAROV B. S., WANG F., SIGMUND O.: Minimum length scale in topology optimization by geometric constraints. *Computer Methods in Applied Mechanics and Engineering 293* (2015), 266–282. 9

[ZPZ13] ZHOU Q., PANETTA J., ZORIN D.: Worst-case structural analysis. *ACM Trans. Graph. 32*, 4 (2013), 137–1. 1, 2, 3, 7

Appendix A: Proof of Prop. 1

Proof. Suppose λ is an eigenvalue of $H^\top K^{-1}H$ and the corresponding eigenvector is ϕ, then

$$\phi = \lambda H^\top K^{-1} H\phi. \tag{11}$$

Multiplying H on both sides of Eq.(11) and denote $y = H\phi$, we have

$$y = \lambda HH^\top K^{-1} y = \lambda PK^{-1} y, \tag{12}$$

which means λ is an eigenvalue of PK^{-1}, and the corresponding eigenvector is y .

Suppose μ is an eigenvalue of PK^{-1} and the corresponding eigenvector is z, then

$$z = \mu PK^{-1} z. \tag{13}$$

Since P is a projection and $P^2 = P$, we get

$$Pz = \mu P^2 K^{-1} z = \mu PK^{-1} z = z. \tag{14}$$

So Eq.(13) can be rewritten as

$$z = \mu PK^{-1} Pz = \mu HH^\top K^{-1} HH^\top z. \tag{15}$$

Multiplying $H^\top$ on both sides of Eq.(15) leads to

$$H^\top z = \mu H^\top HH^\top K^{-1} HH^\top z = \mu H^\top K^{-1} HH^\top z. \tag{16}$$

which shows that μ is an eigenvalue of $H^\top K^{-1}H$ and the corresponding eigenvector is $H^\top z$. □

Appendix B: Proof of Prop. 2

To prove Prop. 2, we first propose the following proposition:

PROPOSITION 4. *Assuming that* $\mathbf{y}^i$ *converges to* $\mathbf{y}$*, then* $\mathbf{y}$ *is an eigenvector of* $PK^{-1}P$.

Proof. In Alg. 1, the iteration in line 8 can be rewritten as:

$$\mathbf{x}^{[i+1]} = \mathbf{x}^{[i]} + V^{-1}(\mathbf{y}^{[i]} - K\mathbf{x}^{[i]}). \tag{17}$$

Note that V^{-1} is a linear operator, so we get:

$$\mathbf{x}^{[i+1]} = (I - V^{-1}K)\mathbf{x}^{[i]} + V^{-1}\mathbf{y}^{[i]}. \tag{18}$$

According to the convergence theorem of the multigrid method [BSS08], for a fixed linear system $K\mathbf{u} = \mathbf{f}$, we have the following relation:

$$\|\mathbf{u}^* - \mathbf{u}^{[i+1]}\| \le \theta \|\mathbf{u}^* - \mathbf{u}^{[i]}\|, \quad \theta < 1, \tag{19}$$

where $\mathbf{u}^*$ is the true solution of the linear system, and $\mathbf{u}^{[i+1]} = \mathbf{u}^{[i]} + V^{-1}(\mathbf{f} - K\mathbf{u}^{[i]})$. Then we get

$$\|(I - V^{-1}K)(\mathbf{u}^* - \mathbf{u}^{[i]})\| \le \theta \|\mathbf{u}^* - \mathbf{u}^{[i]}\|, \tag{20}$$

which implies that $\|I - V^{-1}K\| \le \theta < 1$ since $\mathbf{u}^{[i]}$ is arbitrary. We denote $\mathbf{x} = K^{-1}\mathbf{y}$, and subtract $\mathbf{x}$ from both sides of Eq.(18). Then we have

$$\begin{aligned} \mathbf{x}^{[i+1]} - \mathbf{x} &= (I - V^{-1}K)\mathbf{x}^{[i]} - \mathbf{x} + V^{-1}\mathbf{y}^{[i]} \\ &= (I - V^{-1}K)(\mathbf{x}^{[i]} - \mathbf{x}) + V^{-1}(\mathbf{y}^{[i]} - \mathbf{y}). \end{aligned} \tag{21}$$

From Eq.(21), we get the following inequality

$$\|\mathbf{x}^{[i+1]} - \mathbf{x}\| \le \theta \|\mathbf{x}^{[i]} - \mathbf{x}\| + \eta \|\mathbf{y}^{[i]} - \mathbf{y}\|, \tag{22}$$

where $\eta = \|V^{-1}\|$. We iteratively expand right hand side of (22) and get

$$\begin{aligned} \|\mathbf{x}^{[k+1]} - \mathbf{x}\| &\le \theta \|\mathbf{x}^{[k]} - \mathbf{x}\| + \eta \|\mathbf{y}^{[k]} - \mathbf{y}\| \\ &\le \theta^2 \|\mathbf{x}^{[k-1]} - \mathbf{x}\| + \theta\eta \|\mathbf{y}^{[k-1]} - \mathbf{y}\| + \eta \|\mathbf{y}^{[k]} - \mathbf{y}\| \\ &\le \cdots \\ &\le \theta^k \|\mathbf{x}^0 - \mathbf{x}\| + \eta \sum_{i=0}^{k} \theta^i \|\mathbf{y}^{[k-i]} - \mathbf{y}.\| \end{aligned} \tag{23}$$

The first item in the last inequality converges to 0. For the second item, we divide the sum into two parts

$$\begin{aligned} \sum_{i=0}^{k} \theta^i \|\mathbf{y}^{[k-i]} - \mathbf{y}\| &= \sum_{i=0}^{\lfloor k/2 \rfloor} \theta^i \|\mathbf{y}^{[k-i]} - \mathbf{y}\| + \sum_{\lfloor k/2 \rfloor + 1}^{k} \theta^i \|\mathbf{y}^{[k-i]} - \mathbf{y}\| \\ &\le \sup_{i \ge \lceil k/2 \rceil} \|\mathbf{y}^{[i]} - \mathbf{y}\| \sum_{i=0}^{\lfloor k/2 \rfloor} \theta^i + M \sum_{\lceil k/2 \rceil}^{k} \theta^i \\ &\le \frac{1 - \theta^{\lfloor k/2 \rfloor + 1}}{1 - \theta} \sup_{i \ge \lceil k/2 \rceil} \|\mathbf{y}^{[i]} - \mathbf{y}\| + M\theta^{\lceil k/2 \rceil} \\ &\le \frac{1}{1 - \theta} \sup_{i \ge \lceil k/2 \rceil} \|\mathbf{y}^{[i]} - \mathbf{y}\| + M\theta^{\lceil k/2 \rceil}, \end{aligned} \tag{24}$$

where $M = 2\sup_i \|\mathbf{y}^{[i]}\|$. Since $\mathbf{y}^{[i]}$ converges to $\mathbf{y}$, we have

$$\begin{aligned} &M < +\infty, \\ &\sup_{i \ge \lceil k/2 \rceil} \|\mathbf{y}^{[i]} - \mathbf{y}\| \to 0. \end{aligned} \tag{25}$$

So $\sum_{i=0}^{k} \theta^i \|\mathbf{y}^{[k-i]} - \mathbf{y}\| \to 0$, which implies that $\mathbf{x}^{[k]}$ converges to $\mathbf{x}$ from (23).

As V^{-1} is non-singular, Eq.(17) shows that $\mathbf{r}^{[i]} = \mathbf{y}^{[i]} - K\mathbf{x}^{[i]} \to 0$. Accordingly, we have:

$$\mathbf{y} = P\mathbf{x}/\|P\mathbf{x}\|, \tag{26}$$

$$\mathbf{y} = K\mathbf{x}. \tag{27}$$

We substitute $\mathbf{x} = K^{-1}\mathbf{y}$ into Eq.(26), then

$$\mathbf{y} = \frac{1}{\|P\mathbf{x}\|}PK^{-1}\mathbf{y} = \frac{1}{\|P\mathbf{x}\|}PK^{-1}P\mathbf{y}, \tag{28}$$

where we use the condition that P is a projection and $P^2 = P$.

To ensure that $\mathbf{y}$ is the largest eigenvector, we need more assumptions. First, we give some definitions. We define

$$\mathbf{E}^{[i]} = \mathbf{y}^{[i]} - K\mathbf{x}^{[i+1]}. \tag{29}$$

Since $\mathbf{y}^{[i]}$ and $\mathbf{x}^{[i]}$ converge to $\mathbf{y}$ and $\mathbf{x}$ respectively, we have $\mathbf{E}^{[i]} \to \mathbf{0}$ from Eq.(27). Then $\mathbf{x}^{[i+1]} = K^{-1}\mathbf{y}^{[i]} - K^{-1}\mathbf{E}^{[i]}$, we have

$$c\mathbf{y}^{[i+1]} = PK^{-1}P\mathbf{y}^{[i]} + \mathbf{e}^{[i]}, \tag{30}$$

where c is a constant and $\mathbf{e}^{[i]} = -PK^{-1}\mathbf{E}^{[i]} \to \mathbf{0}$. Again, we used the condition that $P\mathbf{y}^{[i]} = \mathbf{y}^{[i]}$.

Suppose the eigenvalues of $PK^{-1}P$ are $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_n \geq 0$, and the corresponding eigenvectors are $\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_n$, which are the orthonormal basis of $\mathbb{R}^n$. We represent $\mathbf{y}^{[i]}$ and $\mathbf{e}^{[i]}$ under such basis as

$$\mathbf{y}^{[i]} = \sum_j \beta_j^{[i]}\mathbf{v}_j, \tag{31}$$

$$\mathbf{e}^{[i]} = \sum_j \varepsilon_j^{[i]}\mathbf{v}_j, \tag{32}$$

where $\varepsilon_j^{[i]} \to 0$. We define $\delta_j^N = \sup_{k\geq N}|\varepsilon_j^{[k]}|$ and $\delta^N = \max_j \delta_j^N$. □

We can give the proof of Prop.2 now:

Proof. We first prove by mathematical induction that $|\beta_j^{[k]}/\beta_1^{[k]}| \leq C$ for $\forall k \geq N$. When $k = N$, the result is provided in the proposition. Suppose $|\beta_j^{[m]}/\beta_1^{[m]}| \leq C$ for some $m \geq N$; When $k = m+1$, we substitute Eq.(31) into Eq.(30) and get

$$c\mathbf{y}^{[m+1]} = \sum_j (\lambda_j\beta_j^{[m]} + \varepsilon_j^{[m]})\mathbf{v}_j = c\sum_j \beta_j^{[m+1]}\mathbf{v}_j. \tag{33}$$

For $j = 2, ..., n$, we have

$$\left|\frac{\beta_j^{[m+1]}}{\beta_1^{[m+1]}}\right| = \left|\frac{\lambda_j\beta_j^{[m]} + \varepsilon_j^{[m]}}{\lambda_1\beta_1^{[m]} + \varepsilon_1^{[m]}}\right| \leq \frac{C\lambda_j|\beta_1^{[m]}| + \delta^N}{\lambda_1|\beta_1^{[m]}| - \delta^N}, \tag{34}$$

where the last inequality holds since δ^N is a decreasing sequence. Note that

$$\frac{C\lambda_j|\beta_1^{[m]}| + \delta^N}{\lambda_1|\beta_1^{[m]}| - \delta^N} \leq C \iff \delta^N \leq \frac{C(\lambda_1 - \lambda_j)|\beta_1^{[m]}|}{1+C}. \tag{35}$$

Since $|\beta_j^{[m]}| \leq C|\beta_1^{[m]}|$ and $\|\mathbf{y}^{[i]}\| = 1$, we have

$$1 = \sum_j \left(\beta_j^{[m]}\right)^2 \leq \left(1 + C^2(n-1)\right)\left(\beta_1^{[m]}\right)^2. \tag{36}$$

Hence, $\beta_1^{[m]} \geq \frac{1}{\sqrt{1+C^2(n-1)}}$. So the inequality

$$\delta^N \leq \frac{C(\lambda_1 - \lambda_j)}{1+C}\frac{1}{\sqrt{1+C^2(n-1)}} \leq \frac{C(\lambda_1 - \lambda_j)|\beta_1^{[m]}|}{1+C}, \tag{37}$$

holds. From (35), we know that

$$\left|\frac{\beta_j^{[m+1]}}{\beta_1^{[m+1]}}\right| \leq C. \tag{38}$$

Then by mathematical induction, $|\beta_j^{[k]}/\beta_1^{[k]}| \leq C$ holds for $\forall k \geq N$. Thus $\mathbf{y}$ must be the largest eigenvector since $\beta_1^{[k]} \nrightarrow 0$ as $k \to \infty$ from (36). The propostion is proved.

Since $P = HH^\top$, the eigenvectors of $H^\top K^{-1}H$ are the same of $PK^{-1}P$ except by multplying matrix $H^\top$. □

Appendix C: Proof of Prop. 3

Proof. The feasible set of (1) covers the feasible set of (9), thus $c_2 \leq c_1$. We need to prove $\frac{1}{2}c_1 \leq c_2$. For simplicity, we denote $\phi_2(s)$ as the maximal objective value of the following problem

$$\begin{aligned} \max_{\mathbf{f}} \quad & \mathbf{f}^\top K^{-1}\mathbf{f}, \\ \text{s.t.} \quad & \|\mathbf{f}\|_2 \leq s, \\ & N\mathbf{f} = \mathbf{0}, \\ & \mathbf{V}^\top\mathbf{f} \geq 0. \end{aligned} \tag{39}$$

Hence, $c_2 = \phi_2(1)$. We expand the columns of $\mathbf{V}$ to a set of orthonormal basis of $\mathbb{R}^n$ as $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m, \mathbf{w}_{m+1}, \mathbf{w}_{m+2}, \cdots \mathbf{w}_n\}$.

Suppose $\mathbf{f}^*$ is the solution of (1), we can represent it as

$$\mathbf{f}^* = \sum_{i\in\mathscr{I}} \alpha_i\mathbf{v}_i + \sum_{i\in\mathscr{E}} \alpha_i\mathbf{v}_i + \sum_{k\in\mathscr{W}} \gamma_k\mathbf{w}_k, \tag{40}$$

where $\alpha_i = \mathbf{v}_i^\top\mathbf{f}^*$ and $\gamma_k = \mathbf{w}_k^\top\mathbf{f}^*$, $\mathscr{I}$ and $\mathscr{E}$ are the index sets which are defined as

$$\begin{aligned} \mathscr{I} &= \left\{i \in \mathbb{N}^+ : i \leq m,\ \alpha_i < 0\right\}, \\ \mathscr{E} &= \left\{i \in \mathbb{N}^+ : i \leq m,\ \alpha_i \geq 0\right\}. \end{aligned} \tag{41}$$

Then, we define

$$\begin{aligned} \mathbf{f}_+ &= \sum_{i\in\mathscr{E}} \alpha_i\mathbf{v}_i + \sum_{k\in\mathscr{W}} \gamma_k\mathbf{w}_k, \\ \mathbf{f}_- &= \sum_{i\in\mathscr{I}} \alpha_i\mathbf{v}_i. \end{aligned} \tag{42}$$

and get the following relation

$$1 = \|\mathbf{f}^*\|_2^2 = \|\mathbf{f}_+\|_2^2 + \|\mathbf{f}_-\|_2^2. \tag{43}$$

Since $c(\mathbf{f}) = \mathbf{f}^\top K^{-1}\mathbf{f}$ is a convex function of $\mathbf{f}$, we obtain

$$\begin{aligned} c_1 = \mathbf{f}^{*\top}K^{-1}\mathbf{f}^* &\leq \frac{1}{2}\left(4\mathbf{f}_+^\top K^{-1}\mathbf{f}_+ + 4\mathbf{f}_-^\top K^{-1}\mathbf{f}_-\right) \\ &\leq \frac{1}{2}\left(4\phi_2(\|\mathbf{f}_+\|_2) + 4\phi_2(\|\mathbf{f}_-\|_2)\right) \\ &= 2\left(\|\mathbf{f}_+\|_2^2 + \|\mathbf{f}_-\|_2^2\right)\phi_2(1) \\ &= 2\phi_2(1) \\ &= 2c_2. \end{aligned} \tag{44}$$

where the second inequality in (44) holds since $-\mathbf{f}_-$ satisfies the constrain in (39) when $s = \|\mathbf{f}_-\|$, and the third equality holds because $\phi_2(s) = s^2\phi_2(1)$. □