MOX-Report No. 57/2018

POD-assisted strategies for structural topology optimization

Ferro, N.; Micheletti, S.; Perotto, S.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it http://mox.polimi.it

POD-assisted strategies for structural topology optimization

Nicola Ferro[#], Stefano Micheletti[#], Simona Perotto[#]

November 6, 2018

[#] MOX– Modellistica e Calcolo Scientifico
Dipartimento di Matematica, Politecnico di Milano
Piazza L. da Vinci 32, I-20133 Milano, Italy
{nicola.ferro,stefano.micheletti,simona.perotto}@polimi.it

Abstract

We propose a new numerical tool for structural optimization design. To cut down the computational burden typical of the Solid Isotropic Material with Penalization method, we apply Proper Orthogonal Decomposition on SIMP snapshots computed on a fixed grid to construct a rough structure (predictor) which becomes the input of a SIMP procedure performed on an anisotropic adapted mesh (corrector). The benefit of the proposed design tool is to deliver smooth and sharp layouts which require a contained computational effort before moving to the 3D printing production phase.

1 Introduction

Topology optimization methods are nowadays popular thanks to recent developments in 3D and rapid prototyping printing techniques [1]. Topology optimization can be demanding in terms of computational resources, especially when complex structures are designed. Due to this issue, several mathematical methods are commonly employed to reduce the complexity of the problem at hand. The purpose of such methods is to find a trade-off between accuracy and efficiency, by devising procedures characterized by a reduced computational burden without waiving the quality of the final manufactured product.

Our interest is in structural optimization among the several fields of application. Different choices are viable to reach the above trade-off. For instance, in [2, 3] we resort to a customized computational mesh with a contained number of elements, providing smooth and sharp structures almost ready to be printed via additive layer manufacturing. The main idea of this work is to employ a different computational tool to reach the same goal, namely a model reduction procedure based on the Proper Orthogonal Decomposition (POD) [4, 5]. POD is

a standard technique to deal with parametric problems, widely employed in engineering applications [6, 7, 8, 9, 10, 11]. POD exploits an offline/online paradigm, when one first samples the parameter space to collect a certain number of high-fidelity scenarios which are successively employed to extract an informative reduced basis for the space of the parametric solutions; then, this basis is used to recover a new scenario with respect to the ones sampled. In general, the dimension of the reduced basis is considerably lower compared with the dimension of the original problem.

As far as we are aware of, few papers address the employment of POD in a structural optimization context. We cite, for instance, [12] where POD is applied to multi-objective shape optimization, [13] combining POD with homogeneization techniques with a view to a multiscale model, and [14] where a frequency response problem is tackled by combining a standard topology optimization method with POD at an algebraic level.

Structural optimization can be pursued by means of different strategies, ranging from size, to shape and topology optimization [15]. In this work, we focus on topology optimization, namely the design tool seeking an optimal material distribution in an initial domain, for assigned loads and boundary conditions, under some constraints (see [15, 16, 17, 18, 19]). Typical optimality criteria are represented by minimum volume, minimum compliance (or maximum stiffness), maximum fundamental frequency in the dynamic case, while constraints can be maximum allowed displacements and stresses, or a given fraction of the initial volume.
Density-based methods are among the widely employed in the engineering community, which offer an alternative to level-set methods [20, 21], topological derivative procedures [22], phase field techniques [23, 24], evolutionary approaches [25], homogenization [16, 26], performance-based optimization [27].

Here we focus on the minimization of the compliance for a fixed volume fraction, by resorting to the density-based SIMP (Solid Isotropic Material with Penalization) method [15, 16, 19]. In practice, it consists of solving a minimization problem for an auxiliary density variable, identifying the material/void distribution, constrained by the linear elasticity equation. In particular, POD is properly combined with SIMP, by generating a reduced basis for the density only. Extra care will be taken to deal with parameters involving the design constraints. A straightforward combination of POD with SIMP leads to a numerical design process which is efficient but lacks accuracy in some circumstances, in particular when the collected snapshots are excessively either sharp or smooth in correspondence with the material/void boundary. This first merging between POD and SIMP is improved according to a predictor-corrector approach, where the POD prediction is used as an initial guess for the corrector standard SIMP method, further enriched with anisotropic mesh adaptation [2, 28]. We refer to this new method as to PC-SIMPOD. It turns out that PC-SIMPOD is robuster than the basic POD approach, providing the desired trade-off between fast simulations and reliable structures, essentially ready-to-print thanks to mesh adaptation.

The paper is organized as follows. In Section 2, the basic POD approach for topology optimization is introduced and numerically checked on some benchmark configurations, for different choices of the paramenters. Section 3 proposes the PC-SIMPOD method, showing the improvements of this new procedure with respect to the basic POD one. In Section 4, PC-SIMPOD is extended to a multi-parameter setting with a view to practical engineering problems. Conclusions are drawn in the last section with perspectives for the future.

2 POD for topology optimization

This section focuses on a first attempt to contain the computational cost of SIMP algorithm, by resorting to POD. The assessment in Section 2.3 show that we can achieve a considerable gain in terms of computational time even though the predicted layouts are not as performing as expected, in terms of compliance. An additional weak point turns out to be the proper tuning of the procedure used to yield the snapshots for the offline phase.

2.1 The topology optimization technique

Structural topology optimization is a mathematical technique whose goal is to provide an optimized structure fulfilling user-defined requirements. In the most general formulation, it consists in redistributing the material inside an initial design domain in order to satisfy mechanical performances combined with physical constraints. There are several models available in the specialized literature that address the topology optimization problem (see, for instance, [17, 19, 29] for a review on the topic).

Among the several approaches, level set methods and density-based techniques are the most common ones. In both cases, the reference state equation is represented by the linear elasticity problem, suitably incorporating information about the material distribution through an auxiliary function. The level set approach relies on a function χ governed by a time-dependent equation which makes an initial contour propagating towards the optimized final layout [20, 21, 30, 31, 32, 33]. Density-based methods modify the elasticity equation by weighting the Lamé coefficients via a density function, ρ, which identifies the allocation of the material in the structure. In particular, ρ takes values in $[0, 1]$, where $\rho = 0$ means void and $\rho = 1$ material. These methods include phase-field models [24, 34, 23, 35] and the SIMP (Solid Isotropic Material with Penalization) method [36, 15, 37, 38, 39].

In this paper, we focus on the SIMP method. To formalize SIMP method

we start from the density-modified elasticity problem

$$\begin{cases} -\nabla \cdot \sigma_\rho(\mathbf{u}) = \mathbf{0} & \text{in } \Omega \\ \mathbf{u} = \mathbf{0} & \text{on } \Gamma_D \\ \sigma_\rho(\mathbf{u})\mathbf{n} = \mathbf{f} & \text{on } \Gamma_N \\ \sigma_\rho(\mathbf{u})\mathbf{n} = \mathbf{0} & \text{on } \Gamma_F, \end{cases} \tag{1}$$

where $\Omega \subset \mathbb{R}^2$ defines the design domain with boundary $\partial\Omega$; $\mathbf{u} = [u_1, u_2]^T$ is the displacement field; $\sigma_\rho(\mathbf{u}) = \rho^p \big[2\mu\varepsilon(\mathbf{u}) + \lambda I : \varepsilon(\mathbf{u})\big]$ is the penalized stress tensor, with ρ the density function, p the penalization exponent set to 3 [15, 36], $\varepsilon(\mathbf{u}) = \big(\nabla\mathbf{u} + (\nabla\mathbf{u})^T\big)/2$ the strain tensor,

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)}, \qquad \mu = \frac{E}{2(1+\nu)}$$

the Lamé coefficients, with E the Young modulus, and ν the Poisson ratio, I the identity tensor; $\mathbf{f}$ is a traction applied to a portion Γ_N of $\partial\Omega$; $\Gamma_D \subset \partial\Omega$ and $\Gamma_F = \partial\Omega \setminus (\Gamma_N \cup \Gamma_D)$ denote the portion of the domain where the structure is clamped and stress-free, respectively; $\mathbf{n}$ is the unit outward normal vector to $\partial\Omega$.

With a view to the minimization of the structure compliance, $\mathcal{G}(\mathbf{u}) = \int_{\Gamma_N} \mathbf{f} \cdot \mathbf{u}\, d\gamma$, SIMP formulation becomes

find $\rho \in L^\infty(\Omega)$ such that

$$\min_{\rho \in L^\infty(\Omega)} \mathcal{G}(\mathbf{u}(\rho)) \ : \begin{cases} a_\rho(\mathbf{u}(\rho), \mathbf{v}) = \mathcal{G}(\mathbf{v}) \ \forall \mathbf{v} \in U \\ \mathcal{C}(\rho, \mathbf{u}(\rho)) \le 0 \end{cases} \tag{2}$$

with

$$\mathcal{C}(\rho, \mathbf{u}(\rho)) = \begin{cases} \displaystyle\int_\Omega \rho \, d\Omega - \alpha|\Omega| \\ \rho_{\min} - \rho, \\ \rho - 1, \end{cases}$$

and where $\mathbf{u} \in U = \{\mathbf{v} \in [H^1(\Omega)]^2 \ : \ \mathbf{v} = \mathbf{0} \text{ on } \Gamma_D\}$,

$$a_\rho(\mathbf{u}, \mathbf{v}) = \int_\Omega \sigma_\rho(\mathbf{u}) : \varepsilon(\mathbf{v}) \, d\Omega,$$

is the bilinear form associated with (1), $\alpha > 0$ is the maximum volume fraction allowed for the optimized structure, $|\Omega|$ is the measure of the domain, and $\rho_{\min} \in (0, 1)$ is a lower value for the density to ensure the well-posedness of the state equation. Notice that $\rho \mapsto \mathbf{u}(\rho)$ defines the solution operator of the state equation.

The discretization of problem (2) is tackled by using standard finite elements [40], yielding

find $\rho_h \in V_h^r$ such that

$$\min_{\rho_h \in V_h^r} \mathcal{G}(\mathbf{u}_h(\rho_h)) \ : \begin{cases} a_\rho(\mathbf{u}_h(\rho_h), \mathbf{v}_h) = \mathcal{G}(\mathbf{v}_h) \ \forall \mathbf{v}_h \in U_h^s \\ \mathcal{C}(\rho_h, \mathbf{u}_h(\rho_h)) \leq 0 \end{cases} \tag{3}$$

with

$$\mathcal{C}(\rho_h, \mathbf{u}_h(\rho_h)) = \begin{cases} \displaystyle\int_\Omega \rho_h \, d\Omega - \alpha|\Omega| \\ \rho_{\min} - \rho_h, \\ \rho_h - 1, \end{cases} \tag{4}$$

and where $V_h^r \subseteq H^1(\Omega)$ and $U_h^s \subset U$ are the scalar and vector continuous finite element spaces, associated with a triangulation, $\mathcal{T}_h = \{K\}$, of Ω, of degree r and s, respectively and where it is understood that $\mathbf{u}_h(\rho_h) \in U_h^s$.

It is well-known that SIMP suffers from some issues, such as the mesh dependence, the presence of undesired intermediate densities, and checkerboard patterns [15, 41, 42]. In particular, to mitigate the checkerboard effect, it is advisable choosing $r \leq s$ in (3)-(4). As an alternative, we follow the approach proposed in [2, 3], by picking $r = s = 1$. For this particular choice, in the sequel we adopt the simplified notation V_h and U_h.

The minimization is performed using a gradient-like method by properly including the constraints [43].

2.2 The POD method applied to topology optimization

We now aim at performing a structure optimization driven by SIMP at a contained computational cost. For this purpose, we introduce the parametrized version of (3)-(4),

find $\rho_h^\mu \in V_h$ such that

$$\min_{\rho_h^\mu \in V_h} \mathcal{G}(\mathbf{u}_h(\rho_h^\mu)) \ : \begin{cases} a_\rho^\mu(\mathbf{u}_h(\rho_h^\mu), \mathbf{v}_h) = \mathcal{G}^\mu(\mathbf{v}_h) \ \forall \mathbf{v}_h \in U_h \\ \mathcal{C}^\mu(\rho_h^\mu, \mathbf{u}_h(\rho_h^\mu)) \leq 0 \end{cases} \tag{5}$$

with

$$\mathcal{C}^\mu(\rho_h^\mu, \mathbf{u}_h(\rho_h^\mu)) = \begin{cases} \displaystyle\int_\Omega \rho_h^\mu \, d\Omega - \alpha|\Omega| \\ \rho_{\min} - \rho_h^\mu, \\ \rho_h^\mu - 1, \end{cases} \tag{6}$$

with μ a real parameter which may be related to the state equation and/or to the constraint inequality.

Algorithm 1 provides a possible implementation of the computational procedure employed to solve problem (5)-(6), denoted by SIMP_μ.

Algorithm 1 : SIMP_μ

Input : `CTOL, kmax, kmax1, kmax2,` τ_1, τ_2, τ_3, β, $\rho_{\min}$, μ

Set: $\rho_h^0 = 1$, $\mathtt{k} = 0$
while $\mathtt{k} <$ `kmax` **do**
    $\rho_h^{\mathtt{k}+1} = \mathtt{optimize}(\rho_h^{\mathtt{k}}, \mathtt{kmax1}, \mathtt{CTOL}, \rho_{\min}, \mu)$;
    $\rho_h^{\mathtt{k}+1} = \mathtt{filter}(\rho_h^{\mathtt{k}+1}, \tau_1)$;
    `k = k+1`;
endwhile
$\rho_h^{\mathtt{k}+1} = \mathtt{optimize}(\rho_h^{\mathtt{k}}, \mathtt{kmax2}, \mathtt{CTOL}, \rho_{\min}, \mu)$;
$\rho_h^{\mathtt{k}+1} = \mathtt{filter}(\rho_h^{\mathtt{k}+1}, \tau_2)$;
$\rho_h^{\mathtt{k}+1} = \mathtt{sharpening}(\rho_h^{\mathtt{k}+1}, \beta)$;
$\rho_h^{\mathtt{k}+1} = \mathtt{filter}(\rho_h^{\mathtt{k}+1}, \tau_3)$;

This version of SIMP algorithm is a variant of the basic approach, due to the enrichment with both filtering and sharpening. This choice is justified with a view to the offline phase of the POD algorithm. In particular, to perform the filtering, we adopt the Helmholtz-type partial differential problem

$$\begin{cases} -\tau^2 \Delta \rho_f + \rho_f = \rho_h & \text{in } \Omega \\ \tau^2 \nabla \rho_f \cdot \mathbf{n} = 0 & \text{on } \partial\Omega, \end{cases} \tag{7}$$

with τ a real parameter measuring the thickness of the smoothed density, to be properly tuned [44]. In practice, problem (7) is discretized with piecewise linear finite elements.

Concerning the sharpening, we apply the projection step

$$\rho_S = 0.5 \left(1 + \frac{\tanh(\beta(\rho_h - 0.5))}{\tanh(0.5\,\beta)} \right),$$

with β a parameter tuning sharpening features, to emphasize the density gradient, thus yielding a sharper material-void pattern [44, 45, 46]. We observe that sharpening is, in general, applied to a filtered density. The particular alternation of filtering and sharpening adopted in Algorithm 1 will be more precisely justified later on.

Function `optimize` implements a suitable algorithm for constrained minimization. For this purpose, we adopt function `IPOPT` in FreeFem++ [47, 48]. In particular, `CTOL` is a tolerance for the stopping criterion, `kmax1`, `kmax2` set the maximum number of iterations allowed for the optimizer. Routine `IPOPT` requires also the gradient of $\mathcal{G}$ and of $\mathcal{C}$ with respect to the density. For more details about the computation of these gradients, we refer to [2].

According to an offline/online paradigm typical of a POD approach [4, 5], in the offline phase we collect the solutions to the full-size problem by SIMP_μ, called

snapshots, into the response matrix, $\mathcal{S}$, for a sufficiently large set of parameters, $\{\mu_i\}_{i=1}^M$. In particular, we are interested in the output density of Algorithm 1, so that

$$\mathcal{S} = [\boldsymbol{\rho}^h_{\mu_1}, \boldsymbol{\rho}^h_{\mu_2}, ..., \boldsymbol{\rho}^h_{\mu_M}] \in \mathbb{R}^{N\times M},$$

with $\dim(V_h) = N < +\infty$, and where $\boldsymbol{\rho}^h_\mu \in \mathbb{R}^N$ collects the degrees of freedom of ρ^h_μ, solution to (5)-(6), with respect to the basis $\{\varphi_i\}_{i=1}^N$ of V_h. Then, to identify the POD basis we apply the singular value decomposition (SVD) [49] to $\mathcal{S}$,

$$\mathcal{S} = V\Sigma\Phi^T,$$

with $V \in \mathbb{R}^{N\times N}$ and $\Phi \in \mathbb{R}^{M\times M}$ the orthogonal matrices collecting the left and the right singular vectors of $\mathcal{S}$, respectively, while $\Sigma \in \mathbb{R}^{N\times M}$ is the pseudo-diagonal matrix of the singular values of $\mathcal{S}$. The POD basis is thus identified by the first l columns of V, $\{\mathbf{v}_i\}_{i=1}^l$, with $0 < l \leq M$, so that the reduced space $\mathcal{V}^l_{\text{POD}} = \text{span}\{\mathbf{v}_1, \ldots, \mathbf{v}_l\} \subset \mathbb{R}^N$. By exploiting the bijection between V_h and $\mathbb{R}^N$ [40], we define the subspace $V^l_{h,\text{POD}}$ of V_h associated with the subspace $\mathcal{V}^l_{\text{POD}}$ of $\mathbb{R}^N$. As far as the choice of l is concerned, different criteria can be pursued [4, 5]. The role of SVD is to generate a reduced basis which turns out to be particularly effective, by removing the redundancy in the response matrix.

Remark 2.1 *A priori one could build a response matrix also for the displacement. We have decided to work with the density matrix only, to contain the computational cost of the offline phase. Additionally, first numerical checks have highlighted a low accuracy for this double reduction. This topic is currently out of the focus of this paper.*

With a view to the online phase, we introduce matrix $V_l = [\mathbf{v}_1, \ldots, \mathbf{v}_l] \in \mathbb{R}^{N\times l}$ collecting the POD basis vectors, so that, with any vector $\mathbf{w}_l \in \mathbb{R}^l$, we can associate an element $\mathbf{w}_N \in \mathbb{R}^N \cap \mathcal{V}^l_{\text{POD}}$ given by

$$\mathbf{w}_N = V_l\, \mathbf{w}_l. \tag{8}$$

For any $w_h \in V^l_{h,\text{POD}}$, by exploiting the standard expansion in terms of the finite element basis, $\{\varphi_i\}_{i=1}^N$, it follows that, for a suitable $\mathbf{w}_N$ satisfying (8),

$$\begin{aligned} w_h(\mathbf{x}) &= \sum_{i=1}^N w_i\, \varphi_i(\mathbf{x}) = \sum_{i=1}^N \Big(V_l\, \mathbf{w}_l\Big)_i \varphi_i(\mathbf{x}) = \sum_{i=1}^N \left(\sum_{j=1}^l V_{l_{ij}} w_{l_j}\right) \varphi_i(\mathbf{x}) \\ &= \sum_{j=1}^l w_{l_j}\, \tilde{\varphi}_j(\mathbf{x}), \end{aligned}$$

with $\tilde{\varphi}_j(\mathbf{x}) = \sum_{i=1}^N V_{l_{ij}}\, \varphi_i(\mathbf{x})$, i.e., $V^l_{h,\text{POD}} = \text{span}\{\tilde{\varphi}_1, \ldots, \tilde{\varphi}_l\} \subset V_h$, and where we denote by w_i the i-th component of $\mathbf{w}_N$.

We now pick a new value for the parameter μ, say $\mu = \mu^*$, with $\mu^* \neq \mu_i$ with $i = 1 \ldots, M$, and we solve the following $\text{SIMP}_{\mu^*,\text{POD}}$ problem

find $\rho_{h,\text{POD}}^{\mu^*,l} \in V_{h,\text{POD}}^l$ such that

$$\min_{\rho_{h,\text{POD}}^{\mu^*,l} \in V_{h,\text{POD}}^l} \mathcal{G}(\mathbf{u}_h(\rho_{h,\text{POD}}^{\mu^*,l})) \; : \begin{cases} a_\rho^{\mu^*}(\mathbf{u}_h(\rho_{h,\text{POD}}^{\mu^*,l}), \mathbf{v}_h) = \mathcal{G}^{\mu^*}(\mathbf{v}_h) \; \forall \mathbf{v}_h \in U_h \\ \mathcal{C}^{\mu^*}(\rho_{h,\text{POD}}^{\mu^*,l}, \mathbf{u}_h(\rho_{h,\text{POD}}^{\mu^*,l})) \leq 0 \end{cases} \tag{9}$$

instead of the SIMP_{μ^*} in (5)-(6). Solution $\rho_{h,\text{POD}}^{\mu^*,l}$ thus provides an approximation for $\rho_h^{\mu^*}$. The computational benefit expected from this procedure can be ascribed to the fact that (9) involves a constrained minimization problem of dimension l instead of N, in general being $l \ll N$. This implies that few iterations are required to converge. The challenge will be to assess also the reliability of the POD solution.

From an implementative viewpoint, $\text{SIMP}_{\mu^*,\text{POD}}$ is described by the following variant of Algorithm 1:

Algorithm 2 : $\text{SIMP}_{\mu^*,\text{POD}}$

Input : `CTOL`, `kmax`, β, $\rho_{\min}$, μ^*

1: Set: $\rho_h^0 = 1$;
2: $\rho_h^1 = \texttt{optimize}(\rho_h^0, \texttt{kmax}, \texttt{CTOL}, \rho_{\min}, \mu^*)$;
3: $\rho_h^1 = \texttt{sharpening}(\rho_h^1, \beta)$;
4: Set: $\rho_{h,\text{POD}}^{\mu^*,l} = \rho_h^1$;

The whole POD procedure, labeled in the sequel by SIMPOD, is itemized in Algorithm 3.

Algorithm 3 : SIMPOD

Input : `M`, $\{\mu_\texttt{i}\}_{\texttt{i=1}}^{\texttt{M}}$, `l`, μ^*

1: Set: $\mathcal{S} = []$;
2: **for** `i=1:M` **do**
3: $\quad \rho_h^{\mu_\texttt{i}} = \text{SIMP}_{\mu_\texttt{i}}$;
4: $\quad \mathcal{S} = [\mathcal{S}, \boldsymbol{\rho}_h^{\mu_\texttt{i}}]$;
5: **endfor**
6: $[V, \Sigma, \Phi] =$ `svd(`$\mathcal{S}$`)`;
7: `construct space` $V_{h,\text{POD}}^{\texttt{l}}$;
8: $\rho_{h,\text{POD}}^{\mu^*,\texttt{l}}$ `=` $\text{SIMP}_{\mu^*,\text{POD}}$;

We remark that the online phase turns out to be effective if the POD basis does not consist of high-frequency modes. This can be achieved by employing

smooth snapshots with a sufficiently sharp interface between void and material. Actually, the optimizer returns jagged boundaries when no filtering and sharpening are adopted, with associated highly oscillating POD modes. This justifies the tight combination of optimization with filtering and sharpening in SIMP$_\mu$ algorithm.

The effort performed offline allows us to avoid any filtering in the online phase, while keeping only sharpening.

2.3 Numerical results for SIMPOD

We focus on the topology optimization of two different structures, namely a cantilever beam and a bridge. The parameter μ will assume a different meaning, being associated with either the state equation or the inequality constraint.

Concerning the input values for Algorithms 1-2, we refer to Table 1, except for the parameter μ and μ^*, changing through the test-cases.

SIMP$_\mu$	SIMP$_{\mu^*,\text{POD}}$
`CTOL`$= 5 \cdot 10^{-3}$	`CTOL`$= 5 \cdot 10^{-3}$
`kmax`$= 3$	`kmax`$= 300$
`kmax1`$= 50$	$\beta = 7.5$
`kmax2`$= 150$	$\rho_{\min} = 0.01$
$\tau_1 = 0.04$	
$\tau_2 = 0.025$	
$\tau_3 = 0.02$	
$\beta = 5$	
$\rho_{\min} = 0.01$	

Table 1: Input values for SIMP$_\mu$ and SIMP$_{\mu^*,\text{POD}}$ algorithms.

2.3.1 The SIMPOD cantilever beam

We consider the rectangular domain $\Omega = (0,2) \times (0,1) \subset \mathbb{R}^2$, with $\mathbf{x} = [x,y]^T \in \Omega$. The clamped portion of the boundary is $\Gamma_D = \{(x,y) : x = 0, 0 \leq y \leq 1\}$, the traction $\mathbf{f} = [0,-100]^T$ is imposed on $\Gamma_N = \{(x,y) : x = 2, 0.45 \leq y \leq 0.55\}$, while the material constants are $E = 1000$ and $\nu = 1/3$. A triangular structured mesh consisting of 8100 elements discretizes the domain Ω, N being equal to 4186.

We first choose for parameter μ in (5)-(6) the volume fraction, α. The response matrix is assembled with $M = 20$ snapshots, uniformly sampled in the interval $\mathcal{I}_\alpha = [0.2, 0.675]$. For the online phase, we select two new values for the volume fraction, namely, $\mu_1^* = \alpha_1^* = 0.222$ and $\mu_2^* = \alpha_2^* = 0.578$.

Figures 1 and 2, top-left show the full reference solutions, $\rho_h^{0.222}$ and $\rho_h^{0.578}$, respectively, both computed via Algorithm 1 after skipping the last filtering step and with the same inputs as in Table 1. The other panels in the same

figures provide the POD density for the values of l in Tables 2 and 3. The main difference between the two cases is that, while 4 POD modes suffice to detect the topology for the structure in Figure 1, the more complex layout of the cantilever in Figure 2 requires at least 16 modes.

From a quantitative viewpoint, we collect information of interest in Tables 2 and 3, i.e., the CPU time (in seconds[1]), the compliance, $\mathcal{G}$, and the number of iterations demanded by SIMP_{μ^*} and $\text{SIMP}_{\mu^*,\text{POD}}$ to converge, where, for SIMP_{μ^*}, the sum of the iterations involved in the four runs of `optimize` is understood. The values in the two tables exhibit a different trend. For the smaller volume fraction, a few iterations are demanded by the POD procedure, with a consequent reduced CPU time compared with $\text{SIMP}_{\mu_1^*}$ (by a factor 8 in the worst case and 20 for $l = 4$). Nevertheless, the estimated compliance is more than double compared with the reference value, 15.7588. For the larger value of α^*, the predicted compliance is more reliable (with a mismatch of about 14%). Moreover, few iterations suffice to compute the reference solution.

The numerical verification provides a partial justification to the different behaviour in the tables. Actually, the smaller the volume fraction, the larger the number of iterations required by SIMP_{μ^*} to converge, whereas the time demanded by the POD approach essentially scales in the same way for both the configurations.

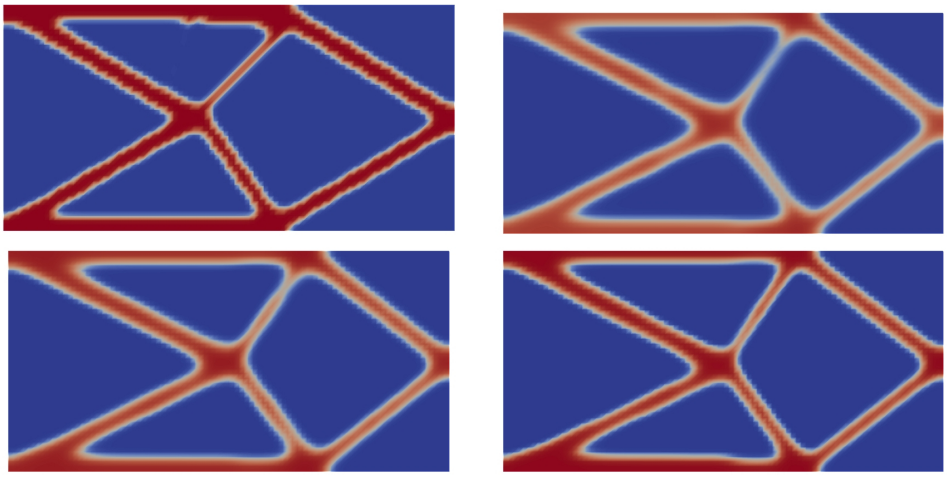

Figure 1: SIMPOD cantilever test case - volume fraction α_1^*: reference solution (top-left); POD solution for $l = 4$ (top-right), $l = 7$ (bottom-left) and $l = 13$ (bottom-right).

We now make a different choice for parameter μ, namely we pick the position, y_f, where the traction is applied. Consequently, the Neumann boundary becomes

[1]The computations have been run on a GenuineIntel Pentium(R) Dual-Core CPU E6300 2.80 GHz 4GB RAM desktop computer.

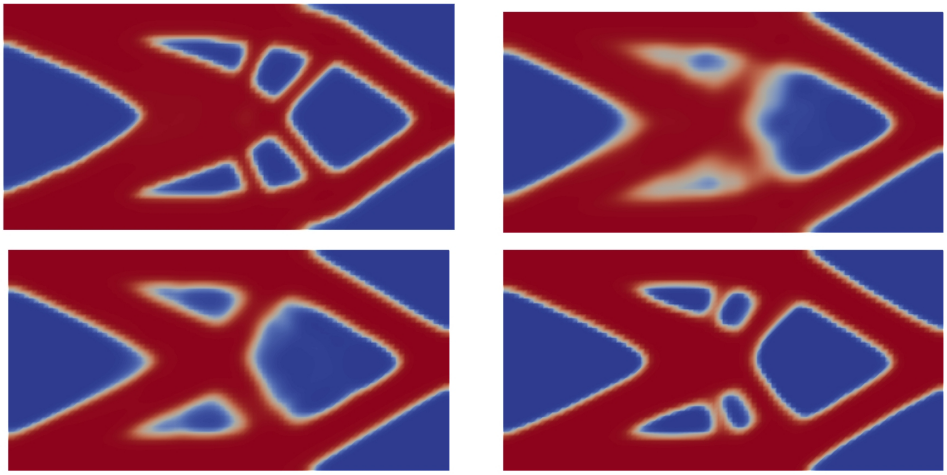

Figure 2: SIMPOD cantilever test case - volume fraction α_2^*: reference solution (top-left); POD solution for $l = 1$ (top-right), $l = 5$ (bottom-left) and $l = 16$ (bottom-right).

	CPU time [s]	$\mathcal{G}$	#iterations
$\text{SIMP}_{\mu_1^*}$	140.71	15.7588	198
$l = 4$	7.27	55.3057	14
$l = 7$	11.14	44.1419	20
$l = 13$	16.58	36.0107	28

Table 2: SIMPOD cantilever test case - volume fraction α_1^*: quantitative data for $\text{SIMP}_{\mu_1^*}$ and $\text{SIMP}_{\mu_1^*,\text{POD}}$ algorithms.

	CPU time [s]	$\mathcal{G}$	#iterations
$\text{SIMP}_{\mu_2^*}$	41.95	4.9978	59
$l = 1$	4.52	6.8983	9
$l = 5$	10.83	6.1095	20
$l = 16$	27.14	5.7209	28

Table 3: SIMPOD cantilever test case - volume fraction α_2^*: quantitative data for $\text{SIMP}_{\mu_2^*}$ and $\text{SIMP}_{\mu_2^*,\text{POD}}$ algorithms.

$\Gamma_N = \{(x, y) : x = 2, |y - y_f| \leq 0.05\}$ and α is now set to 0.5. Matrix $\mathcal{S}$ is built starting from $M = 11$ snapshots, and selecting y_f uniformly in the interval $[0, 1]$. For the online phase, we choose $\mu^* = y_f^* = 0.111$. The reference topology is shown in Figure 3, top-left.

Figure 3, top-right, center and bottom, provides the layout predicted by SIMPOD algorithm for the five values of l in Table 4. We remark that only two modes provide a layout with the same topology as the reference structure,

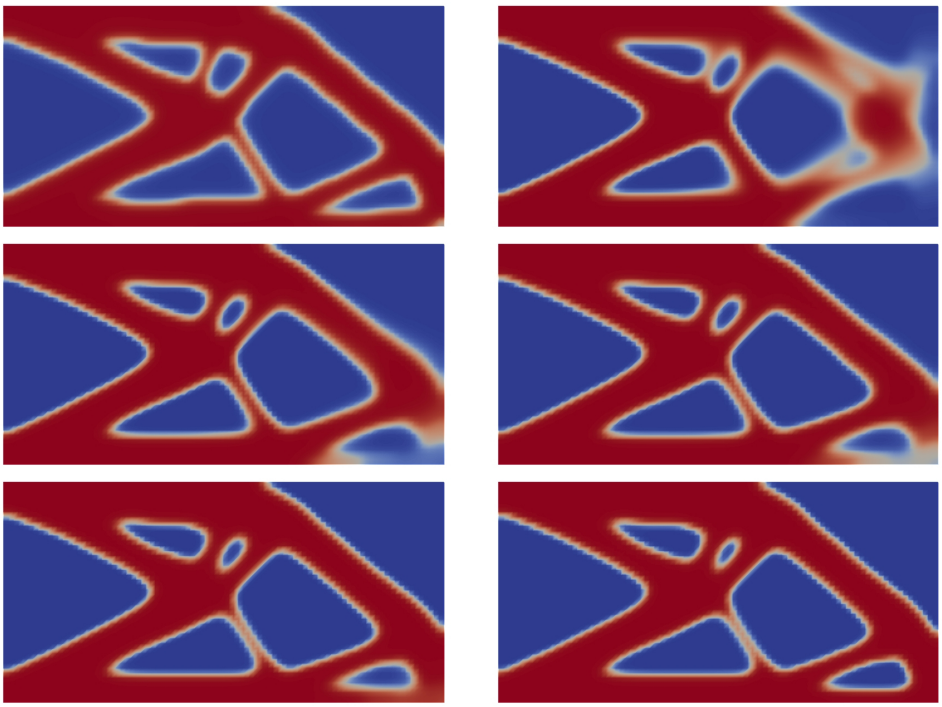

Figure 3: SIMPOD cantilever test case - traction position: reference solution (top-left); POD solution for $l = 1$ (top-right), $l = 2$ (center-left), $l = 3$ (center-right), $l = 5$ (bottom-left) and $l = 8$ (bottom-right).

	CPU time [s]	$\mathcal{G}$	#iterations
SIMP_{μ^*}	56.11	6.5420	79
$l = 1$	3.88	22.0962	8
$l = 2$	5.55	9.5097	11
$l = 3$	14.04	8.4831	27
$l = 5$	10.35	7.6995	18
$l = 8$	16.50	7.5410	16

Table 4: SIMPOD cantilever test case - traction position: quantitative data for SIMP_{μ^*} and $\text{SIMP}_{\mu^*,\text{POD}}$ algorithms.

except for some detail in the bottom-right portion of Ω. Eventually, 8 POD modes furnish an accurate prediction for the expected cantilever, as confirmed also by the quantitative analysis in Table 4. Actually, the reduction of the computational time amounts to a factor of about 3.4, while the discrepancy on the compliance is about 15%.

2.3.2 The SIMPOD bridge

This second test case deals with the optimization of a bridge. The domain is $\Omega = (0,6) \times (0,1)$, discretized with an isotropic triangular mesh consisting of 11260 elements, with $N = 5840$. The traction $\mathbf{f} = [0, -100]^T$ is imposed on the portion $\Gamma_N = \{(x,y) : 2.9 \leq x \leq 3.1, y = 1\}$ of the boundary. On $\Gamma_{D_1} = \{(x,y) : 0 \leq x \leq 0.06, y = 0\}$ we impose that the vertical displacement is null to model a roller, whereas on $\Gamma_{D_2} = \{(x,y) : 5.94 \leq x \leq 6\}$ both the components of the displacements are set to zero. Finally, the material constants are $E = 1000$ and $\nu = 1/3$ as for the cantilever test case.

The parameter adopted for the POD analysis is the volume fraction, α. The offline phase is performed with SIMP$_\mu$ algorithm with the same input values as in Table 1, except for τ_3 which is now set to 0.05, and choosing $M = 20$ values of μ evenly distributed in $\mathcal{I}_\alpha = [0.2, 0.675]$. The parameter selected for the online phase is $\alpha^* = 0.362$.

Figure 4, top-left exhibits the reference solution together with three predictions corresponding to an increasing number of POD modes. At least, 12 modes have to be employed to obtain a somewhat accurate prediction, as shown in Figure 4, top-right and bottom.

Concerning the data in Table 5, the computational saving provided by SIMPOD is about eight times in the worst case, even though the predicted compliance is not so accurate, differing from the reference one of about 19% for the largest value of l ($l = 12$).

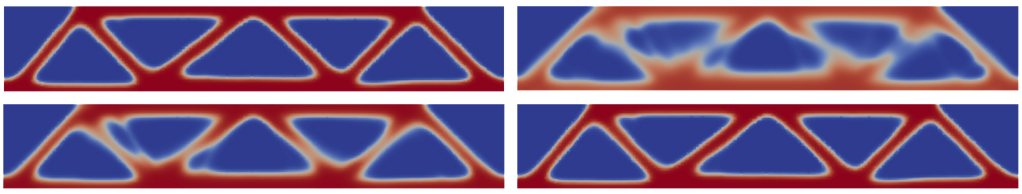

Figure 4: SIMPOD bridge test case - volume fraction: reference solution (top-left); POD solution for $l = 1$ (top-right), $l = 2$ (bottom-left), $l = 12$ (bottom-right).

	CPU time [s]	$\mathcal{G}$	#iterations
SIMP$_{\mu^*}$	227.53	63.6929	273
$l = 1$	5.4	158.1365	8
$l = 2$	10.23	107.4605	14
$l = 12$	28.57	75.7358	35

Table 5: SIMPOD bridge test case - volume fraction: quantitative data for SIMP$_{\mu^*}$ and SIMP$_{\mu^*,\text{POD}}$ algorithms.

3 An enhanced approach

In the previous section, a basic-POD approach for topology optimization has been proposed and verified. As the numerical experiments show, the algorithm succeeds in remarkably reducing the computational time, delivering a reliable result, yet rough and, in general, less performing with respect to the reference solution if the number of POD modes is not sufficiently large. To overcome this limit, it is standard to resort to a larger number of snapshots. This option can be prohibitive in terms of memory usage, since the number of entries in $\mathcal{S}$ linearly depends on the number of snapshots, and the optimal value for both M and l is, in general, not known a priori.

As an alternative, we here propose a new approach, where the output of SIMPOD is used as the initial guess, cheap and rough, for a new run of a topology optimization procedure, in the spirit of a predictor-corrector method, named PC-SIMPOD.

3.1 A predictor-corrector SIMPOD (PC-SIMPOD) technique enriched with mesh adaptation

The scheme here proposed exploits the advantages of SIMPOD (predictor) to quickly obtain an initial guess (more accurate than a dummy choice) for the optimization of the structure, so that the topology optimizer (corrector) can be initialized in a neighbourhood of the solution. The expected result of such a combination is that a few optimization iterations in the correction step suffice to deliver a solution that is competitive with the one directly produced by the SIMP method on a generic initial guess.

In particular, for the correction step, we adopt the adaptive version of SIMP algorithm, namely SIMPATY, proposed in [2, 28], where the employment of a fixed mesh as in Section 2.3 is replaced by a grid which sharply follows the boundaries of the structure to be optimized. As shown in [2, 3], the main benefit of SIMPATY is to yield sharp layouts characterized by smooth material/void boundaries, almost ready for the 3D printing process. This is achievable using an anisotropic mesh adaptation procedure driven by a sound mathematical tool, namely an a posteriori error estimator.

Anisotropic meshes allow us to properly tune the size of mesh elements together with the corresponding shape and orientation. To get all this information, we resort to the setting used in [50, 51, 52, 53], based on the affine transformation $T_K : \hat{K} \to K$, mapping the equilateral reference element $\hat{K}$ inscribed in the unit circle, into a generic element K of $\mathcal{T}_h$,

$$\mathbf{x} = T_K(\hat{\mathbf{x}}) = M_K \hat{\mathbf{x}} + \mathbf{b}_K \quad \forall \hat{\mathbf{x}} \in \hat{K},$$

where $M_K \in \mathbb{R}^{2\times 2}$ deforms and rotates the reference element and $\mathbf{b}_K$ shifts it. The Jacobian M_K can be factorized by successively applying the polar and the

spectral decomposition, so that

$$M_K = (R_K^T \Lambda_K R_K) Z_K,$$

where $Z_K \in \mathbb{R}^{2\times 2}$ is a rotation matrix, $\Lambda_K = \mathrm{diag}(\lambda_{1,K}, \lambda_{2,K}) \in \mathbb{R}^{2\times 2}$ collects the eigenvalues of the symmetric positive definite matrix $R_K^T \Lambda_K R_K$, with $\lambda_{1,K} \geq \lambda_{2,K}$, and $R_K^T = [\mathbf{r}_{1,K}, \mathbf{r}_{2,K}] \in \mathbb{R}^{2\times 2}$ is the orthogonal matrix of the corresponding eigenvectors. Matrices Λ_K and R_K contain the geometric features of the element K, namely the length, $\lambda_{i,K}$, of the semi-axes of the ellipse circumscribed to K, and the directions, $\mathbf{r}_{i,K}$, of such axes, with $i = 1, 2$ (see Figure 5). Triangle K is thus fully identified by the three quantities $\{\lambda_{1,K}, \lambda_{2,K}, \mathbf{r}_{1,K}\}$. The aspect ratio $s_K = \lambda_{1,K}/\lambda_{2,K} \geq 1$ provides a measure of the deformation of the element, with the understanding that high values for s_K are associated with very stretched elements.

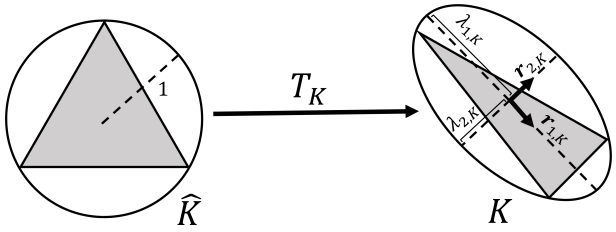

Figure 5: Map T_K from the reference element $\hat{K}$ to the generic one K.

SIMPATY algorithm relies on a posteriori error estimator to generate the anisotropic adapted mesh. We resort to an anisotropic variant of the recovery-based error estimator proposed by O.C. Zienkiewicz and J.Z. Zhu in [54, 55, 56], following [57, 53]. In particular, the estimator η for the $H^1(\Omega)$-seminorm of the density discretization error, $|\rho - \rho_h|_{H^1(\Omega)}$, is given by

$$\eta^2 = \sum_{K\in\mathcal{T}_h} \eta_K^2,$$

with

$$\eta_K^2 = \frac{1}{\lambda_{1,K}\lambda_{2,K}} \sum_{i=1}^{2} \lambda_{i,K}^2 \left(\mathbf{r}_{i,K}^T G_{\Delta_K}(E_\nabla) \mathbf{r}_{i,K}\right) \tag{10}$$

the element contribution, and where $E_\nabla = [P(\nabla\rho_h) - \nabla\rho_h]_{\Delta_K}$ is the recovered error,

$$P(\nabla\rho_h)|_K = \frac{1}{|\Delta_K|} \sum_{T\in\Delta_K} |T|\, \nabla\rho_h|_T$$

is the recovered gradient associated with K, obtained via an area-weighted average of the gradient of ρ_h computed on the patch Δ_K of the elements sharing

at least one vertex with K, $|\Delta_K|$ is the area of Δ_K and $G_{\Delta_K}(\cdot) \in \mathbb{R}^{2\times 2}$ is the symmetric positive semidefinite matrix with entries

$$[G_{\Delta_K}(\mathbf{w})]_{i,j} = \sum_{T\in\Delta_K} \int_T w_i\, w_j\, dT \quad \text{with } i,j = 1,2, \tag{11}$$

for any vector-valued function $\mathbf{w} = (w_1, w_2)^T \in [L^2(\Omega)]^2$.

Estimator η is then turned into a practical information to build the new adapted mesh. This is carried out via a metric-based procedure [58], which aims at minimizing the number of elements to ensure a certain accuracy, $\eta \leq \texttt{TOL}$, in combination with an equidistribution of the error over the triangles. This leads to the prediction of the optimal spacing, $\{\lambda_{i,K}^{\text{opt}}\}_{i=1}^2$, and orientation, $\{\mathbf{r}_{i,K}^{\text{opt}}\}_{i=1}^2$, for each element K of the mesh via an iterative procedure. This can be accomplished following [59], so that

$$\lambda_{1,K}^{\text{opt}} = g_2^{-1/2}\left(\frac{\texttt{TOL}^2}{2\#\mathcal{T}_h\,|\widehat{\Delta}_K|}\right)^{1/2}, \quad \lambda_{2,K}^{\text{opt}} = g_1^{-1/2}\left(\frac{\texttt{TOL}^2}{2\#\mathcal{T}_h\,|\widehat{\Delta}_K|}\right)^{1/2}, \tag{12}$$
$$\mathbf{r}_{1,K}^{\text{opt}} = \mathbf{g}_2, \quad \mathbf{r}_{2,K}^{\text{opt}} = \mathbf{g}_1,$$

with $|\widehat{\Delta}_K| = |\Delta_K|/(\lambda_{1,K}\lambda_{2,K})$, and where $\{g_i, \mathbf{g}_i\}_{i=1,2}$ are the eigenvalue-eigenvector pairs associated with the scaled matrix $\widehat{G}_{\Delta_K}(E_\nabla) = G_{\Delta_K}(E_\nabla)/|\Delta_K|$, with $g_1 \geq g_2 > 0$, $\{\mathbf{g}_i\}_{i=1,2}$ orthonormal vectors, and $\#\mathcal{T}_h$ denotes the cardinality of the mesh elements.

The spacing and the orientation thus predicted, $\{\lambda_{i,K}^{\text{opt}}, \mathbf{r}_{i,K}^{\text{opt}}\}_{i=1}^2$, become the input to a metric-based mesh generator which produces the new anisotropic adapted mesh. The process is repeated in an iterative loop, until some convergence criterion is met.

The PC-SIMPOD with mesh adaptation procedure is itemized in Algorithm 4.

Algorithm 4 : PC-SIMPOD with anisotropic mesh adaptation

Input : $\texttt{MTOL}, \texttt{CTOL}, \texttt{TOL}, \texttt{kmax}, \rho_{\min}, \mathcal{T}_h^0, \mu^*, \texttt{l}$

1: $\texttt{k} = 0$, errM $= 1 + \texttt{MTOL}$;
2: Compute $\rho_{h,POD}^{\mu^*,\texttt{l}}$ with Algorithm 3;
3: Set $\rho_h^0 = \rho_{h,POD}^{\mu^*,\texttt{l}}$;
4: **while** errM $>$ MTOL & k $<$ kmax **do**
5: kmax1=20-3k;
6: $\rho_h^{\texttt{k}+1} = \texttt{optimize}(\rho_h^{\texttt{k}}, \texttt{kmax1}, \texttt{CTOL}, \rho_{\min}, \mu^*)$;
7: $\mathcal{T}_h^{\texttt{k}+1} = \texttt{adapt}(\mathcal{T}_h^{\texttt{k}}, \rho_h^{\texttt{k}+1}, \texttt{TOL})$;
8: errM $= |\#\mathcal{T}_h^{\texttt{k}+1} - \#\mathcal{T}_h^{\texttt{k}}|/\#\mathcal{T}_h^{\texttt{k}}$;
9: $\texttt{k} = \texttt{k} + 1$;
10: **endwhile**

The routine `adapt` in line 7: generates the new adapted mesh identified by the optimal spacing and orientation in (12). Concerning the stopping criterion, we rely on the stagnation of the cardinality of the elements between two consecutive meshes, to within the tolerance `MTOL`. We remark that an interpolation of the density onto the new adapted mesh is understood at each change of the grid before restarting the procedure (see [2] for more details). Moreover, neither filtering nor sharpening are applied in this algorithm, since mesh adaptation automatically provides smooth structures which avoid the employment of these procedures.

3.2 Numerical results for PC-SIMPOD

We show here the improvements led by Algorithm 4 in terms of accuracy on the test-cases in Section 2.3.1 and 2.3.2.

3.2.1 The PC-SIMPOD cantilever beam

We adopt as predictor density the output of SIMPOD algorithm for two different values of l.
The scenario to be recovered is the same as in Figure 3, where the selected parameter coincides with the traction position, namely $\mu^* = 0.111$. In more detail, we start from the two less accurate POD approximations in Figure 3, namely $\rho_h^0 = \rho_{h,POD}^{\mu^*,1}$ and $\rho_h^0 = \rho_{h,POD}^{\mu^*,2}$. Concerning the input parameters for Algorithm 4, we set

$$\texttt{MTOL} = 0.01, \quad \texttt{CTOL} = 10^{-4}, \quad \texttt{TOL} = 0.125, \quad \texttt{kmax} = 4, \quad \rho_{\min} = 0.01,$$

while $\mathcal{T}_h^0$ coincides with the structured mesh of 8100 triangles in Section 2.3.1.

For the sake of comparison, the new reference solution coincides with the output of Algorithm 4, skipping step 2:, and directly setting ρ_h^0 in 3: as the approximation provided by SIMP_{μ^*} (see Figure 6, top).

Figure 6, center-bottom shows the output of PC-SIMPOD for $l = 1$ and $l = 2$, respectively. We remark that the PC-SIMPOD approximation for $l = 1$ is very close to the reference structure even though the initial guess, $\rho_{h,POD}^{\mu^*,1}$, is poor. The final topology is slightly different from the expected one because of the presence of an additional hole in the bottom-right corner, and it also exhibits a bent contour in correspondence with the traction application point.
Starting from a barely richer initial guess is enough to obtain a much more accurate layout, as shown in the bottom panel of the figure. Concerning the computational mesh, it is evident that the anisotropic features allow us to detect the void/material interface in a sharp manner and yield a smooth layout.

Table 6 enriches the quantities in the previous tables with additional information, namely the time, T_C, required by the **while** loop in Algorithm 4, the total number, $\#\texttt{iter}_C$, of iterations of `optimize` in the corrector step, the mesh cardinality, $\#\mathcal{T}_h$, and the maximum aspect ratio, $\max_K s_K$, of the final

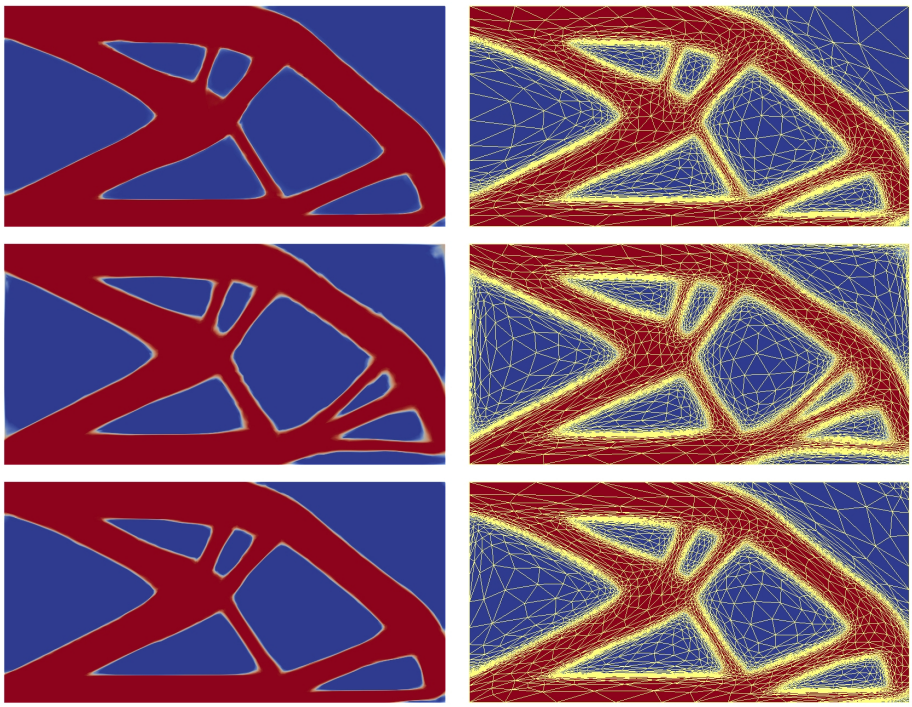

Figure 6: PC-SIMPOD cantilever test case - traction position: density (left) and density superposed to the mesh (right) when $\rho_h^0 = \text{SIMP}_{\mu^*}$ (top), $\rho_h^0 = \rho_{h,POD}^{\mu^*,1}$ (center), $\rho_h^0 = \rho_{h,POD}^{\mu^*,2}$ (bottom).

	CPU time [s]	$\mathcal{G}$	T_C [s]	$\#\texttt{iter}_C$	$\#\mathcal{T}_h$	$\max_K s_K$
SIMP_{μ^*}	113.72	5.1831	59.32	61	5802	564.43
$l = 1$	51.09	6.5215	47.39	45	5920	97.33
$l = 2$	62.36	5.6006	57.06	58	5438	345.58

Table 6: PC-SIMPOD cantilever test case - traction position: quantitative data for PC-SIMPOD for different choices of ρ_h^0.

anisotropic adapted mesh.
The total CPU time demanded by PC-SIMPOD for $l = 1, 2$, is essentially half the time associated with the same procedure starting from $\rho_h^0 = \text{SIMP}_{\mu^*}$. As expected, most of the computational time is spent in the **while** loop. The maximum discrepancy in the compliance is of about +10% with respect to the reference configuration. The adaptive procedure delivers in all the three cases meshes with a similar number of triangles, which are, in general, highly stretched.

3.2.2 The PC-SIMPOD bridge

We aim at reconstructing the structure in Figure 4, top-left starting from the SIMPOD approximation in the bottom-right panel of the same figure (namely, $\mu^* = 0.362$). Algorithm 4 is slightly modified by setting `kmax1`=15 for all iterations, and choosing as input values

$$\texttt{MTOL} = 0.01, \quad \texttt{CTOL} = 10^{-4}, \quad \texttt{TOL} = 0.1, \quad \texttt{kmax} = 12, \quad \rho_{\min} = 0.01,$$

with $\mathcal{T}_h^0$ the same mesh used for SIMPOD.
The reference solution, shown in Figure 7, top, is computed via Algorithm 4 directly setting the initial guess, ρ_h^0, for the density as the approximation delivered by SIMP_{μ^*}.

Figure 7 shows the good matching between the reference and the PC-SIMPOD solutions despite some local differences can be detected. However, these heterogeneities do not affect the mechanical performance of the layout yielded by the predictor. Actually, the compliance of the PC-SIMPOD bridge is thoroughly comparable with the reference value, while the gain in terms of computational time is remarkable (the CPU time is, in practice, halved).

Finally, the anisotropic mesh is highly stretched as confirmed by the values of the maximum stretching factors in Table 7.

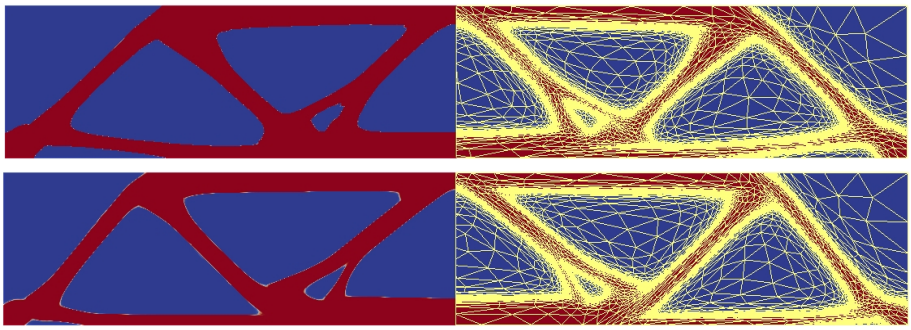

Figure 7: PC-SIMPOD bridge test case - volume fraction: density (left) and density superposed to the mesh (right) when $\rho_h^0 = \text{SIMP}_{\mu^*}$ (top), $\rho_h^0 = \rho_{h,POD}^{\mu^*,12}$ (bottom).

	CPU time [s]	$\mathcal{G}$	T_C [s]	$\#\texttt{iter}_C$	$\#\mathcal{T}_h$	$\max_K s_K$
SIMP_{μ^*}	342.67	29.9937	115.02	62	32480	1540.80
$l = 12$	156.49	30.9364	104.66	58	43308	1310.30

Table 7: PC-SIMPOD bridge test case - volume fraction: quantitative data for PC-SIMPOD for two choices of ρ_h^0.

4 Multi-parameter topology optimization

In Sections 2 and 3, we have focused on the case where structure optimization depends on just one parameter. Nevertheless, realistic configurations involve more parameters, simultaneously. For instance, with reference to test cases tackled in the previous sections, it could be of interest to identify the optimal configuration for a new pair (α, y_f).

From a formal viewpoint, SIMP$_\mu$ formulation can be rewritten as in (5)-(6) simply by replacing the scalar parameter μ by the q-dimensional vector of parameters, $\boldsymbol{\mu} = [\mu^1, \mu^2, ..., \mu^q]^T \in \mathbb{R}^q$. In practice, for each parameter μ^j, we consider S_j different values, $\mu^j_{i_j}$, with $i_j = 1, \ldots, S_j$ and $j = 1, \ldots, q$. SIMP$_\mu$ algorithm is then employed to generate a discrete density $\rho_h^{\boldsymbol{\mu}_\mathbf{I}}$ for each parameter $\boldsymbol{\mu}_\mathbf{I} = [\mu^1_{i_1}, \mu^2_{i_2}, \ldots, \mu^q_{i_q}]^T \in \mathbb{R}^q$ where $\mathbf{I} = [i_1, i_2, \ldots, i_q]^T \in \mathbb{N}^q$, with $i_1 = 1, \ldots, S_1$, $i_2 = 1, \ldots, S_2$, $i_q = 1, \ldots, S_q$. The offline phase thus collects a total of $M = S_1\, S_2 \ldots S_q$ snapshots that have to be properly gathered into a generalization of the standard response matrix usually referred to as atlas, $\mathcal{A}$. With this aim, several approaches can be employed, ranging from an arbitrary organization of the densities $\rho_h^{\boldsymbol{\mu}_\mathbf{I}}$ into a two-dimensional $(N \times M)$ matrix to a q-dimensional array. The first approach is viable if q is small and the standard SVD can be employed to extract the POD basis. In such a case, the ordering of $\rho_h^{\boldsymbol{\mu}_\mathbf{I}}$ in $\mathcal{A}$ is arbitrary. We adopt the index ordering based on the following `for` loops:

```
for i1 = 1:S1
for i2 = 1:S2
...
for iq = 1:Sq
...
end
end
end
```

On the contrary, for large values of q, the approach based on the q-dimensional array turns out to be advisable and a Higher-Order SVD (HOSVD) can be adopted as a more performing procedure to extract the reduced basis [60, 61].

Since we limit the numerical assessment to the case of two parameters, we define $\mathcal{A}$ as

$$\mathcal{A} = \left[\boldsymbol{\rho}_h^{\mu_1^1\mu_1^2}, ..., \boldsymbol{\rho}_h^{\mu_1^1\mu_{S_2}^2}, \boldsymbol{\rho}_h^{\mu_2^1\mu_1^2}, ..., \boldsymbol{\rho}_h^{\mu_2^1\mu_{S_2}^2}, ..., \boldsymbol{\rho}_h^{\mu_{S_1}^1\mu_1^2}, ..., \boldsymbol{\rho}_h^{\mu_{S_1}^1\mu_{S_2}^2}\right],$$

where the generic density $\boldsymbol{\rho}_h^{\mu_{i_1}^1\mu_{i_2}^2}$ is the output of the multi-parameter version, multi-SIMP$_{\boldsymbol{\mu}}$, of Algorithm 1, with μ replaced by the vector $\boldsymbol{\mu} = [\mu^1_{i_1}, \mu^2_{i_2}]^T$, and we apply the standard SVD to $\mathcal{A}$ to extract the POD basis.

The online phase is started by selecting the new multi-parameter, $\boldsymbol{\mu}^* = [\mu^{*,1}, \mu^{*,2}, ..., \mu^{*,q}]^T \in \mathbb{R}^q$, and by resorting to the multi-parameter version, multi-SIMP$_{\boldsymbol{\mu}^*,\text{POD}}$, of SIMP$_{\mu^*,\text{POD}}$.

We will refer to the whole procedure here described as to multi-SIMPOD. In a straightforward way, we can generalize the PC-SIMPOD procedure enriched with anisotropic mesh adaptation to the multi-parameter case, denoting the resulting procedure by multi-PC-SIMPOD.

Goal of the next section is to investigate the performances of both multi-SIMPOD and multi-PC-SIMPOD methods.

4.1 Numerical results for multi-SIMPOD and multi-PC-SIMPOD

We first check the performances of multi-SIMPOD algorithm focusing on the cantilever test case in Section 2.3.1, and by choosing as multi-parameter $\boldsymbol{\mu} = [\mu^1, \mu^2]^T = [\alpha, y_f]^T$. We investigate the sensitivity of the predicted layout to two different atlas consisting of 25 and 50 snapshots. For the first atlas, $\mathcal{A}_1$, we set $S_1 = S_2 = 5$ and

$$\mu^1 \in \{0.3, 0.4, 0.5, 0.6, 0.7\}, \quad \mu^2 \in \{0, 0.25, 0.5, 0.75, 1\}.$$

For the second atlas, $\mathcal{A}_2$, $S_1 = 10$ and $S_2 = 5$ corresponding to the following samplings

$$\mu^1 \in \{0.3, 0.325, 0.35, 0.375, 0.4, 0.45, 0.5, 0.55, 0.575, 0.6\},$$
$$\mu^2 \in \{0, 0.25, 0.5, 0.75, 1\}.$$

The input parameters for SIMP_μ are the same as in Table 1, except for τ_3 and β now set to 0.0286 and 10, respectively, while $\text{SIMP}_{\mu^*,\text{POD}}$ shares all the input values. The computational mesh is the structured one as in Section 2.3.1. The online multi-parameter is $\boldsymbol{\mu}^* = [\mu^{*,1}, \mu^{*,2}]^T = [0.333, 0.444]^T$. As reference structure, we consider the output provided by multi-$\text{SIMP}_{\boldsymbol{\mu}^*}$.

Figure 8 and Table 8 summarize the output of multi-SIMPOD procedure from a qualitative and quantitative viewpoint, respectively. Entries in Table 8 preserve the same meaning as for the previous test cases. At least 5 POD modes have to be adopted to correctly identify the traction area, while almost all of the 25 modes are required to obtain a reliable prediction of the layout.

Similarly to the single-parameter setting, we observe a bad prediction of the mechanical stiffness which is about 140% higher in the multi-SIMPOD case ($l = 22$) with respect to the reference configuration. The computational time reduces, however, by a factor 8.

Due to the poor structural performances of multi-SIMPOD for atlas $\mathcal{A}_1$, we investigate if a finer sampling of the possible scenarios improves the quality of the prediction by resorting to atlas $\mathcal{A}_2$.

In Figure 9, we compare the density distribution provided by multi-SIMPOD for different choices of l with the reference configuration. At least 22 modes are required to identify the final topology, whereas 41 modes deliver a sharp and smooth structure very close to the reference one. Table 9 provides a more quantitative assessment. The performances are still not so satisfactory. Indeed, the

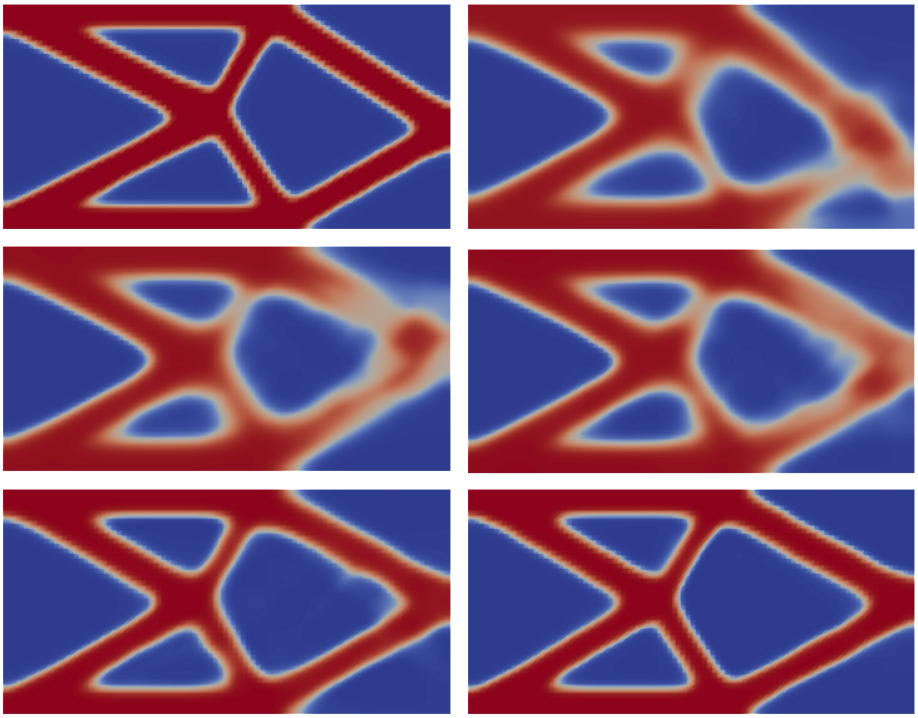

Figure 8: multi-SIMPOD cantilever test case for atlas $\mathcal{A}_1$: reference solution (top-left); POD solution for $l = 3$ (top-right), $l = 4$ (center-left), $l = 5$ (center-right), $l = 13$ (bottom-left) and $l = 22$ (bottom-right).

	CPU time [s]	$\mathcal{G}$	#iterations
multi-SIMP$_{\boldsymbol{\mu}^*}$	172.08	7.4551	249
$l = 3$	8.68	39.4493	15
$l = 4$	19.33	33.7280	33
$l = 5$	13.92	29.4964	23
$l = 13$	26.38	19.9066	39
$l = 22$	21.56	17.9476	34

Table 8: multi-SIMPOD cantilever test case for atlas $\mathcal{A}_1$: quantitative data for multi-SIMP$_{\boldsymbol{\mu}^*}$ and multi-SIMP$_{\boldsymbol{\mu}^*,\mathrm{POD}}$ algorithms.

mismatch between the two configurations in terms of mechanical stiffness is about 100% higher in the multi-SIMPOD case ($l = 41$) with respect to the reference configuration. The gain in terms of computational time is now of a factor about equal to 4. Moreover, it turns out that there are critical configurations where a large number of iterations is demanded, possibly due to the switching of the solution from two different minima of the compliance functional.

Moving from the improvements led by the PC-SIMPOD algorithm in Sec-

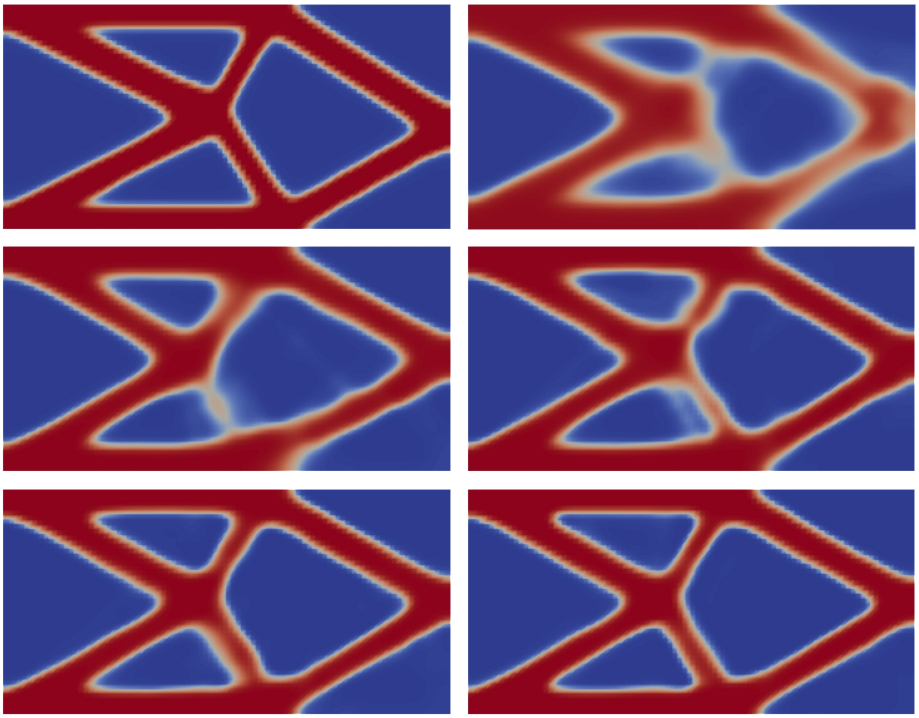

Figure 9: multi-SIMPOD cantilever test case for atlas $\mathcal{A}_2$: reference solution (top-left); POD solution for $l = 4$ (top-right), $l = 17$ (center-left), $l = 22$ (center-right), $l = 31$ (bottom-left) and $l = 41$ (bottom-right).

	CPU time [s]	$\mathcal{G}$	#iterations
multi-SIMP$_{\boldsymbol{\mu}^*}$	172.08	7.4551	249
$l = 4$	10.12	31.1832	18
$l = 17$	70.05	20.2008	109
$l = 22$	49.93	18.6441	75
$l = 31$	38.96	15.9707	52
$l = 41$	39.73	14.6244	46

Table 9: multi-SIMPOD cantilever test case for atlas $\mathcal{A}_2$: quantitative data for multi-SIMP$_{\boldsymbol{\mu}^*}$ and multi-SIMP$_{\boldsymbol{\mu}^*,\mathrm{POD}}$ algorithms.

tion 3.2, we apply the multi-parameter version of such an algorithm to both atlases. Figure 10 compares the reference layout provided by multi-PC-SIMPOD when fed by the output of multi-SIMP$_{\boldsymbol{\mu}^*}$, with the multi-PC-SIMPOD approximation initialized by the structure predicted by multi-SIMPOD algorithm for $l = 4$. There is no striking difference between the two cantilevers, except for a slight discrepancy at the tip, which exhibits a mild bending when starting from the first atlas.

In Table 10, we gather the same quantities as in Table 6 for $l = 4$ and $l = 22$. These two values for l are the only ones shared by Tables 8 and 9. Overall, the four configurations do not yield appreciable differences in terms of compliance and of computational time, for the same l. Also the meshes have a similar number of elements as well as maximum aspect ratio. Nevertheless, since the computational gain provided by the choice $l = 4$ is much higher compared with $l = 22$ (about 3 and 2 times, respectively), we can reasonably assume that the first choice pays off both in terms of accuracy and computational saving.

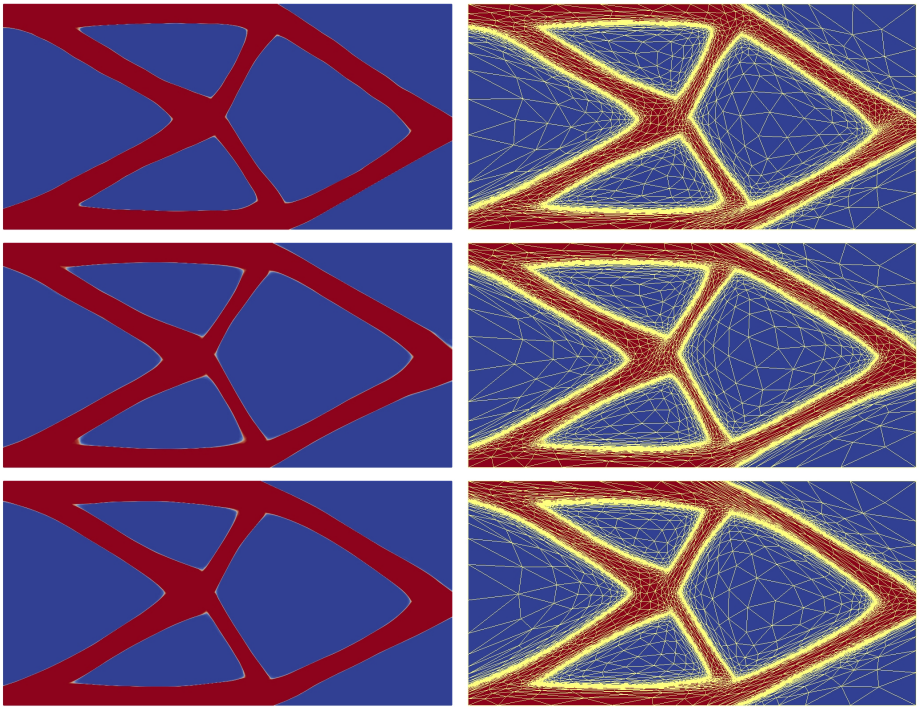

Figure 10: multi-PC-SIMPOD cantilever test case: density (left) and density superposed to the mesh (right) when ρ_h^0 =multi-SIMP$_{\boldsymbol{\mu}^*}$ (top), $\rho_h^0 = \rho_{h,POD}^{\boldsymbol{\mu}^*,4}$ for atlas $\mathcal{A}_1$ (center), $\rho_h^0 = \rho_{h,POD}^{\boldsymbol{\mu}^*,4}$ for atlas $\mathcal{A}_2$ (bottom).

		CPU time [s]	$\mathcal{G}$	T_C [s]	$\#\texttt{iter}_C$	$\#\mathcal{T}_h$	$\max_K s_K$
	multi-SIMP$_{\boldsymbol{\mu}^*}$	249.81	7.3418	78.64	105	12770	921.32
$\mathcal{A}_1$	$l = 4$	76.45	7.5701	60.26	85	11146	342.75
	$l = 22$	103.83	7.1408	82.79	110	9178	539.94
$\mathcal{A}_2$	$l = 4$	74.07	7.4773	65.05	90	12068	458.27
	$l = 22$	124.98	7.4173	78.64	105	12914	349.10

Table 10: multi-PC-SIMPOD cantilever test case: quantitative data for multi-PC-SIMPOD for different choices of ρ_h^0.

5 Conclusions

For the sake of clarity, in Table 11 we provide an overview of the methods considered in this paper, by highlighting some of the associated main features, based on the numerical assessment in Sections 2.3, 3.2, 4.1. Namely, we supply a short description of the methods, classified in offline and/or online (`off/on-line`), computationally efficient and/or reliable (E /R) with respect to the standard SIMP algorithm [15, 16, 19], and based on a fixed or adapted (F /A) mesh.

It turns out that three are the methods outperforming the others, namely, SIMPATY and PC-SIMPOD for the single-parameter case, and multi-PC-SIMPOD for the multi-parameter case. Nevertheless, Table 11 emphasizes qualitative information only. Thus, to compare more deeply SIMPATY with PC-SIMPOD, we refer to Tables 6 and 7, while for SIMPATY versus multi-PC-SIMPOD to Table 10. In general, it turns out that SIMPATY is slightly more reliable as it provides structures with better mechanical properties, whereas PC-SIMPOD and multi-PC-SIMPOD are more efficient, cutting down the CPU time by a factor 2.

On the contrary, the plain application of POD to SIMP on a fixed mesh leads to very efficient simulations which are, however, not so reliable from the mechanical standpoint. Although there is still some room for improvements by a more careful tuning of filtering and sharpening, we are confident that PC-SIMPOD and multi-PC-SIMPOD methods are the ones to be supported as a robust design tool for structural topology optimization. Moreover, we expect that the advantages observed in the two-dimensional case will be magnified by generalizing these methods to a 3D context, which represents the next step of our research.

		`off/on-line`	E / R	F / A
SIMPATY	SIMP+anisotropic mesh adaption [2, 28]	✗ / ✓	✓ / ✓	✗ / ✓
SIMP_{μ}	SIMP+filtering+sharpening	✓ / ✗	✗ / ✓	✓ / ✗
$\text{SIMP}_{\mu^*,\text{POD}}$	SIMP on the reduced space+sharpening	✗ / ✓	✓ / ✗	✓ / ✗
SIMPOD	SIMP_{μ}+$\text{SIMP}_{\mu^*,\text{POD}}$	✓ / ✓	✓ / ✗	✓ / ✗
PC-SIMPOD	predictor: SIMPOD + corrector: SIMPATY	✗ / ✓	✓ / ✓	✗ / ✓
multi-$\text{SIMP}_{\boldsymbol{\mu}}$	multi-parameter SIMP+filtering+sharpening	✓ / ✗	✗ / ✓	✓ / ✗
multi-$\text{SIMP}_{\boldsymbol{\mu}^*,\text{POD}}$	multi-SIMP on the reduced space+sharpening	✗ / ✓	✓ / ✗	✓ / ✗
multi-SIMPOD	multi-SIMP_{μ}+multi-$\text{SIMP}_{\mu^*,\text{POD}}$	✓ / ✓	✓ / ✗	✓ / ✗
multi-PC-SIMPOD	predictor: multi-SIMPOD + corrector: SIMPATY	✗ / ✓	✓ / ✓	✗ / ✓

Table 11: Main features of the methods considered in the paper.

6 Acknowledgments

The authors acknowledge the research project GNCS-INdAM 2018 “Tecniche di Riduzione di Modello per le Applicazioni Mediche”, which partially supported this research.

References

[1] D. Brackett, I. Ashcroft, R. Hague, Topology optimization for additive manufacturing, in: Proceedings of the solid freeform fabrication symposium, Austin, TX, Vol. 1, S, 2011, pp. 348–362.

[2] S. Micheletti, S. Perotto, L. Soli, Topology optimization driven by anisotropic mesh adaptation: towards free-form design, submitted.

[3] N. Ferro, S. Micheletti, S. Perotto, A sequential coupling of shape and topology optimization for structural design, submitted.

[4] M. Kahlbacher, S. Volkwein, Galerkin proper orthogonal decomposition methods for parameter dependent elliptic systems, Discuss. Math. Differ. Incl. Control Optim. 27 (1) (2007) 95–117.

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

[6] M. D. Gunzburger, Perspectives in Flow Control and Optimization, Vol. 5 of Advances in Design and Control, Society for Industrial and Applied Mathematics (SIAM),Philadelphia, PA, 2003.

[7] K. Kunisch, S. Volkwein, Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics, SIAM J. Numer. Anal. 40 (2) (2002) 492–515.

[8] P. LeGresley, J. Alonso, Airfoil design optimization using reduced order models based on proper orthogonal decomposition, AIAA Paper (2000-2545).

[9] S. S. Ravindran, A reduced-order approach for optimal control of fluids using proper orthogonal decomposition, Internat. J. Numer. Methods Fluids 34 (5) (2000) 425–448.

[10] L. Sirovich, Turbulence and the dynamics of coherent structures. I. Coherent structures, Quart. Appl. Math. 45 (3) (1987) 561–571.

[11] K. Willcox, J. Peraire, Balanced model reduction via the proper orthogonal decomposition, AIAA J. 40 (2002) 2323–2330.

[12] B. Raghavan, M. Hamdaoui, M. Xiao, P. Breitkopf, P. Villon, A bi-level meta-modeling approach for structural optimization using modified POD bases and diffuse approximation, Computers & Structures 127 (2013) 19 – 28.

[13] L. Xia, P. Breitkopf, A reduced multiscale model for nonlinear structural topology optimization, Comput. Methods Appl. Mech. Engrg. 280 (2014) 117–134.

[14] G. H. Yoon, Structural topology optimization for frequency response problem using model reduction schemes, Comput. Methods Appl. Mech. Engrg. 199 (25-28) (2010) 1744–1763.

[15] M. P. Bendsøe, O. Sigmund, Topology Optimization: Theory, Methods and Applications, Springer-Verlag, Berlin Heidelberg, 2003.

[16] M. P. Bendsøe, N. Kikuchi, Generating optimal topologies in structural design using a homogeneization method, Comput. Methods Appl. Mech. Eng. 71 (2) (1988) 197–224.

[17] O. Sigmund, K. Maute, Topology optimization approaches, a comparative review, Struct. Multidiscip. Optim. 48 (6) (2013) 1031–1055.

[18] J. D. Deaton, R. V. Grandhi, A survey of structural and multidisciplinary continuum topology optimization: post 2000, Struct. Multidiscip. Optim. 49 (2014) 1–38.

[19] G. I. N. Rozvany, A critical review of established methods of structural topology optimization, Struct. Multidiscip. Optim. 37 (2009) 217–237.

[20] G. Allaire, F. Jouve, A. Toader, Structural optimization using sensitivity analysis and level set-method, J. Comput. Phys. 194 (2004) 363–393.

[21] M. Y. Wang, X. Wang, D. Guo, A level set method for structural topology optimization, Comput. Methods Appl. Mech. Engrg. 192 (1-2) (2003) 227–246.

[22] J. Sokolowski, A. Zochowski, On the topological derivative in shape optimization, SIAM J. Control. Opt. 37 (1999) 1251–1272.

[23] L. Dedè, M. J. Borden, T. J. R. Hughes, Isogeometric analysis for topology optimization with a phase field model, Arch. Comput. Methods Eng. 19 (2012) 427–465.

[24] B. Bourdin, A. Chambolle, Design-dependent loads in topology optimization, ESAIM Control. Optim. Calc. Var. 9 (2003) 19–48.

[25] Y. Xie, G. Steven, A simple evolutionary procedure for structural optimization, Comput. Struct. 49 (1993) 885–896.

[26] G. Allaire, F. Jouve, H. Maillot, Topology optimization for minimum stress design with the homogenization method, Struct. Multidisc. Optim. 28 (2004) 87–98.

[27] Q. Liang, Performance-based Optimization of Structures. Theory and Applications, Spon Press, London, 2005.

[28] S. Micheletti, S. Perotto, L. Soli, Ottimizzazione topologica adattativa per la fabbricazione stratificata additiva, Italian patent application No. 102016000118131, filed on November 22, 2016 (extended as *Adaptive topology optimization for additive layer manufacturing*, International patent application PCT No. PCT/IB2017/057323) (2017).

[29] H. Eschenauer, N. Olhoff, Topology optimization of continuum structures: a review, Appl. Mech. Rev. 54 (4) (2001) 331–390.

[30] M. Burger, S. J. Osher, A survey on level set methods for inverse problems and optimal design, European J. Appl. Math. 16 (2) (2005) 263–301.

[31] V. J. Challis, J. K. Guest, Level set topology optimization of fluids in Stokes flow, Int. J. Numer. Meth. Engng 79 (10) (2009) 1284–1308.

[32] K. Yaji, M. Otomori, T. Yamada, K. Izui, S. Nishiwaki, O. Pironneau, Shape and topology optimization based on the convected level set method, Struct. Multidiscip. Optim. 54 (3) (2016) 659–672.

[33] F. de Gournay, G. Allaire, F. Jouve, Shape and topology optimization of the robust compliance via the level set method, ESAIM Control Optim. Calc. Var. 14 (1) (2008) 43–70.

[34] H. Garcke, C. Hecht, M. Hinze, C. Kahle, Numerical approximation of phase field based shape and topology optimization for fluids, SIAM J. Sci. Comput. 37 (4) (2015) A1846–A1871.

[35] M. Y. Wang, S. Zhou, Phase field: a variational method for structural topology optimization, CMES Comput. Model. Eng. Sci. 6 (6) (2004) 547–566.

[36] S. Amstutz, Connections between topological sensitivity analysis and material interpolation schemes in topology optimization, Struct. Multidiscip. Optim. 43 (6) (2011) 755–765.

[37] M. P. Bendsøe, O. Sigmund, Material interpolation schemes in topology optimization, Arch. Appl. Mech. 69 (9) (1999) 635–654.

[38] M. P. Bendsøe, Optimal shape design as a material distribution problem, Struct. Optimization 1 (4) (1989) 193–202.

[39] M. P. Bendsøe, Optimization of Structural Topology, Shape, and Material, Springer-Verlag, Berlin, 1995.

[40] P. G. Ciarlet, The Finite Element Method for Elliptic Problems, North-Holland Publishing Co., Amsterdam-New York-Oxford, 1978.

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

[42] A. Díaz, O. Sigmund, Checkerboard patterns in layout optimization, Struct. Multidiscip. Optim. 19 (1995) 89–92.

[43] J. Nocedal, S. J. Wright, Numerical Optimization, Springer Series in Operations Research, Springer-Verlag, New York, 1999.

[44] B. S. Lazarov, O. Sigmund, Filters in topology optimization based on Helmholtz-type differential equations, Int. J. Numer. Meth. Engng 86 (6) (2011) 765–781.

[45] J. K. Guest, J. H. Prevost, T. Belytschko, Achieving minimum length scale in topology optimization using nodal design variables and projection functions, Int. J. Numer. Methods Engng 61 (2) (2004) 238–254.

[46] O. Sigmund, Morphology-based black and white filters for topology optimization, Struct. Multidiscip. Optim. 33 (2007) 401–424.

[47] F. Hecht, New development in FreeFem++, J. Numer. Math. 20 (3-4) (2012) 251–265.

[48] A. Wächter, L. T. Biegler, On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming, Math. Program. 106 (1, Ser. A) (2006) 25–57.

[49] G. H. Golub, C. F. Van Loan, Matrix computations, 4th Edition, Johns Hopkins Studies in the Mathematical Sciences, Johns Hopkins University Press, Baltimore, MD, 2013.

[50] L. Formaggia, S. Perotto, New anisotropic a priori error estimates, Numer. Math. 89 (2001) 641–667.

[51] L. Formaggia, S. Micheletti, S. Perotto, Anisotropic mesh adaption with application to CFD problems, in: H. Mang, F. Rammerstorfer, J. Eberhardsteiner (Eds.), Proceedings of WCCM V, Fifth World Congress on Computational Mechanics, 2002, pp. 1481–1493.

[52] L. Dedè, S. Micheletti, S. Perotto, Anisotropic error control for environmental applications, Appl. Numer. Math. 58 (9) (2008) 1320–1339.

[53] S. Micheletti, S. Perotto, P. E. Farrell, A recovery-based error estimator for anisotropic mesh adaptation in CFD, Bol. Soc. Esp. Mat. Apl. SeMA 50 (2010) 115–137.

[54] O. C. Zienkiewicz, J. Z. Zhu, A simple error estimator and adaptive procedure for practical engineering analysis, Int. J. Numer. Meth. Engng 24 (1987) 337–357.

[55] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. I: The recovery technique, Int. J. Numer. Meth. Engng 33 (1992) 1331–1364.

[56] O. C. Zienkiewicz, J. Z. Zhu, The superconvergent patch recovery and a posteriori error estimates. II: Error estimates and adaptivity, Int. J. Numer. Meth. Engng 33 (1992) 1365–1382.

[57] S. Micheletti, S. Perotto, Anisotropic adaptation via a Zienkiewicz-Zhu error estimator for 2D elliptic problems, in: G. Kreiss, P. Lötstedt, A. Målqvist, M. Neytcheva (Eds.), Numerical Mathematics and Advanced Applications, Springer-Verlag Berlin Heidelberg, 2010, pp. 645–653.

[58] P.-L. George, H. Borouchaki, Delaunay Triangulation and Meshing. Application to Finite Elements, Editions Hermès, Paris, 1998.

[59] S. Micheletti, S. Perotto, Reliability and efficiency of an anisotropic Zienkiewicz-Zhu error estimator, Comput. Methods Appl. Mech. Engrg. 195 (9–12) (2006) 799–835.

[60] K. Kamalja, N. Khangar, Singular value decomposition for multidimensional matrices, Int. J. Eng. Res. Appl. 3 (6) (2013) 123–129.

[61] L. De Lathauwer, B. De Moor, J. Vandewalle, A multilinear singular value decomposition, SIAM J. Matrix Anal. Appl. 21 (4) (2000) 1253–1278.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

56/2018 Antonietti, P.F.; Manzini, G.; Verani, M.
The conforming virtual element method for polyharmonic problems

55/2018 Cerroni, D.; Laurino, F.; Zunino, P.
Mathematical analysis, finite element approximation and numerical solvers for the interaction of 3D reservoirs with 1D wells

54/2018 Dal Santo, N.; Deparis, S.; Manzoni, A.; Quarteroni, A.
Multi space reduced basis preconditioners for parametrized Stokes equations

53/2018 Giantesio, G.; Musesti, A.; Riccobelli, D.
A comparison between active strain and active stress in transversely isotropic hyperelastic materials

52/2018 Possenti, L.; di Gregorio, S.; Gerosa, F.M.; Raimondi, G.; Casagrande, G.; Costantino, M.L.; Z
A computational model for microcirculation including Fahraeus-Lindqvist effect, plasma skimming and fluid exchange with the tissue interstitium

51/2018 Stella, S.; Vergara, C.; Giovannacci, L.; Quarteroni, A.; Prouse, G.
Assessing the disturbed flow and the transition to turbulence in the arteriovenous fistula

50/2018 Gervasio, P.; Quarteroni, A.
The INTERNODES method for non-conforming discretizations of PDEs

47/2018 Stefanucci, M.; Sangalli, L.M.; Brutti, P.
PCA-based discrimination of partially observed functional data, with an application to Aneurisk65 dataset

48/2018 Arnone, E.; Azzimonti, L.; Nobile, F.; Sangalli, L.M.
Modeling spatially dependent functional data via regression with di fferential regularization

49/2018 Massi, M.C.; Ieva, F.; Lettieri, E.
Data Mining Application to Healthcare Fraud Detection: A Two-Step Unsupervised Clustering Model for Outlier Detection with Administrative Databases