Engineering with Computers (2024) 40:21–43
https://doi.org/10.1007/s00366-023-01823-w

ORIGINAL ARTICLE

Adaptive density-based robust topology optimization under uncertain loads using parallel computing

David Herrero-Pérez[1] · Sebastián Ginés Picó-Vicente[1] · Humberto Martínez-Barberá[2]

Received: 15 October 2021 / Accepted: 18 April 2023 / Published online: 10 May 2023

Abstract

This work presents an efficient parallel implementation of density-based robust topology optimization (RTO) using adaptive mesh refinement (AMR) schemes permitting us to address the problem with modest computational resources. We use sparse grid stochastic collocation methods (SCMs) for transforming the RTO problem into a weighted multiple-loading deterministic problem at the collocation points. The calculation of these deterministic problems and the functional sensitivity is computationally expensive. We combine distributed-memory parallel computing and AMR techniques to address the problem efficiently. The former allows us to exploit the computational resources available, whereas the latter permits us to increase performance significantly. We propose the parallel incremental calculation of the deterministic problems and the contribution to the functional sensitivity maintaining a similar memory allocation to the one used in the deterministic counterpart. The cumulative computing uses buffers to adapt the evaluation at the collocation points to the parallel computing resources permitting the exploitation of the embarrassing parallelism of SCMs. We evaluate the deterministic problems in a coarse mesh generated for each topology optimization iteration to increase the performance. We perform the regularization and design variable update in a fine mesh to obtain an equivalent design to the one generated in such a mesh. We evaluate the proposal in two- and three-dimensional problems to test its feasibility and scalability. We also check the performance improvement using computational buffers in parallel computing nodes. Finally, we compare the proposal to the same approach using different preconditioners without AMR schemes showing significant performance improvements.

Keywords Robust topology optimization · Sparse grid · Adaptivity · Parallel computing · Multigrid methods

1 Introduction

Topology optimization aims to find the optimal distribution of material within a design domain by minimizing a cost function subjected to a set of constraints. It is a powerful tool for engineers and scientists providing innovative and high-performance conceptual designs at the early stages of the design process without assuming any prior structural configuration. For this reason, it applies successfully to a broad spectrum of applications under deterministic conditions [16, 45]. However, ignoring the different sources of uncertainty in the design process can degrade the optimal design performance under real-world engineering conditions. These sources of uncertainty may affect the final structural design performance, safety, and reliability. We usually differentiate between epistemic and aleatory uncertainties. The former is due to things one could know but does not know in practice, such as inaccurate measurements due to limited data and unknowledge.

S. G. Picó-Vicente and Martínez-Barberá have contributed equally to this work.

✉ David Herrero-Pérez
david.herrero@upct.es

Sebastián Ginés Picó-Vicente
sebastian.pico@upct.es

Humberto Martínez-Barberá
humberto@um.es

[1] Structures, Construction and Graphic Expression, Technical University of Cartagena, Campus Muralla del Mar, Cartagena 30202, Murcia, Spain

[2] Department of Information and Communications Engineering, University of Murcia, Campus de Espinardo, Murcia 30100, Murcia, Spain

The latter includes the natural randomness in a process, such as manufacturing imperfections [49], uncertain loading conditions [36], and variations of the material properties [37], to name but a few. Introducing such sources of uncertainty in the optimization process provides more robust and reliable designs for real-world conditions.

Many topology optimization approaches incorporate sources of uncertainty in the design process. We can broadly classify such methods into non-probabilistic and probabilistic approaches [52]. The former [40] does not require statistical information about the uncertainty of the phenomenon but a qualitative notion about its magnitude. We can include the worst-case approach [12, 19] and the fuzzy techniques [35] in such a category. The worst-case methods formulate topology optimization as a min–max optimization problem, whereas the fuzzy techniques use fuzzy set theory for decision-making and uncertainty propagation. We can employ these techniques when we only have an approximate notion about the measure of the unknown variables, which we usually model using an interval where it could be. The latter approaches use the stochastic characterization of the uncertainty of the phenomenon. There are different stochastic formulations differing in the form uncertainty is taken into account by the functional and in the way is incorporated into the optimization process. We can mention robust topology optimization (RTO) and reliability-based topology optimization (RBTO). The RTO methods consider the statistical moments during optimization minimizing the sensitivity of designs to perturbations of the input data [6]. The RBTO approaches introduce reliability constraints into the deterministic topology optimization formulation to ensure an admissible failure probability to the optimized design [28]. We adopt the RTO formulation considering uncertain loading conditions [11, 21].

The efficient and accurate calculation of statistic moments is crucial for addressing real-world problems. The computational complexity of computing the statistical moments in RTO makes it a computational challenge. Adopting the stochastic collocation method (SCM) allows us to solve the integrals of such statistic moments considering loading uncertainty. We transform the RTO formulation into a weighted multiple-loading deterministic problem. The number of collocation points needed for accurately estimating the statistic moments becomes then the bottleneck of the RTO problem. The integration based on full tensor products suffers from the *curse of dimensionality*. The Smolyak sparse grid method significantly reduces the number of collocation points without losing much accuracy [64]. However, computing such collocation points still represents a computational challenge due to the memory consumption and the processing time when dealing with finite-element models of real-world applications. RTO can then become intractable due to the computational cost.

We can adopt different strategies to reduce the computational cost of calculating the set of deterministic problems used to compute the integrals of the statistic moments using sparse grid SCMs. We can mention rescaling large systems of equations to reduce the ill-conditioning [60], approximate reanalysis [3] to avoid many analyses of the modified design, using low-accurate approximations of the analysis solution [4], and adopting efficient preconditioning techniques to improve the performance of iterative methods [5]. We also have to mention the multiresolution schemes, which decouple analysis and design discretizations. These methods use a coarse mesh for the analysis and a fine grid for the design variables [33, 42] to improve the computational performance. However, we have a limitation in the maximum resolution of the design variables depending on the order and type of finite elements [22]. We also can improve the computational performance using High-Performance Computing (HPC) to address the computationally intensive tasks of the topology optimization process using multi-core [2, 9, 34] and many-core [24, 36, 44] computing.

Adaptive mesh refinement (AMR) techniques [29] using h-refinement and coarsening permit us to save computational cost and reduce the error by increasing and decreasing the number of finite elements needed in the regions of interest. We usually determine these areas of interest using an error estimator, which indicates either coarsening the unnecessary finite elements or refining them in the zone of interest. Such error estimators can focus on different magnitudes from the analysis and the design discretization. We can mention solution-based error estimators to enhance the approximation of a concerning measure in topology optimization, such as the stress estimation in stress-constrained topology optimization using AMR [47, 48]. We also can mention refining the interface in topology optimization for a better interface definition [41] and sensitivity calculation [7]. Since void material regions contribute meaningfully to the computing burden of topology optimization but little to the accuracy of the optimized design, we adopt an error estimation for detecting weak material regions to coarse them. This strategy improves the computational cost and provides an equivalent material distribution to the one generated on a uniformly fine mesh.

The early work of [14] uses the h-adaptive finite-element method in topology optimization following a recursive strategy. They assume the converged solution on a coarse grid to refine it and start a new optimization on the fine mesh. They do not revisit coarse meshes after the generation of the new fine mesh. For this reason, this approach generates a smoother solution of the resulting topology optimization in the coarse mesh instead of the counterpart solution in the fine mesh. In other words, the topology optimization in the fine mesh can result in a very different design from the solution refining the design obtained in the coarse meshes.

Therefore, this strategy may lead to suboptimal designs. That is why, [15] introduce the coarsening in the adaptive method to generate a similar result to the one generated on a uniformly fine mesh. They obtain significant computational improvements when the topology optimization requires relatively small volume fractions, since the design domain is highly void after a few iterations of the optimization process. It is well known that the computation of regions with weak material contributes significantly to the overall computational cost but little to the accuracy of the optimized design.

Some recent works use dynamic AMR with parallel computing to address high-resolution topology optimization using the minimization of structural compliance. [32] propose an adaptive shared-memory topology optimization framework using solid isotropic material with penalization (SIMP) technique. They restrict the region to analyze elastic deformations to a narrow band surrounding the high-density zone. They use a matrix-free implementation of the conjugate gradient solver preconditioned with a geometric multigrid (GMG) method for Cartesian grids. This strategy allows them to obviate the computational effort on large void regions on the fly using an assembly-free instance. [31] propose an adaptive distributed-memory topology optimization framework using the level-set method for evolving a high-resolution unstructured mesh. They use an algebraic multigrid (AMG) method as the preconditioner of the Krylov subspace iteration solver for evaluating the objective function and the reaction–diffusion equation used for the level-set update stage. The proposal achieves clear and high-resolution solid-void material boundary reducing the computational cost by coarsening the mesh distributed in the void domain.

[7] address large-scale stochastic topology optimization problems using adaptive mesh refinement and coarsening with a parallelization scheme distributing the computation through the nodes of large supercomputers. They use sparse grid SCM for calculating the stochastic moments included in the objective function. They group the computing resources into physical computing nodes using each group of threads to calculate the problems at the collocation points. This scheme minimizes inter-node communications reducing bandwidth problems. However, the number of required computing nodes increases as we use more collocation points. This work also uses a level-set method for evolving a high-resolution mesh focusing the computational effort on the zero level-set representing the design interface. A significant drawback of the approach is that it uses a progressive refinement approach similar to the scheme proposed by [14], performing a fixed number of topology optimization iterations with a maximum refinement level and increasing it in the next interval. Thus, this scheme can result in a refined design of the one obtained in the coarse mesh, i.e., it can generate suboptimal designs, as mentioned above.

The proposed work aims to reduce the computational cost of density-based RTO under uncertain loads by combining dynamic AMR techniques and parallel computing. We aim to obtain an equivalent design to the one generated on a uniform fine mesh using distributed memory computing but at a much cheaper computational cost. The bottleneck of RTO using sparse grid SCMs to integrate the statistical moments of structural compliance is the finite-element analysis (FEA) of the set of deterministic problems at the collocation points. We use a fine mesh for regularizing and optimizing to obtain a similar design than using such a fine mesh. For performance reasons, we use a dynamically coarse grid based on the thresholding of design variables for the FEA of the deterministic problems at the collocation points.

We propose the incremental calculation of the contribution of each collocation point in the dynamically coarse mesh using computing buffers [25]. This scheme can take advantage of reusing the displacement solution of the previous FEA [24], the assembled matrix of coefficients, and the setup stage of the multigrid method used for preconditioning the parallel solver. It also permits us to exploit the embarrassing level of parallelism of sparse grid SCMs. Besides, it provides adaptability by fitting the computational resources to the requirements of the RTO problem. On the other hand, we use parallel computing to accelerate the resolution of the system of equations of each deterministic problem. The main issue using this approach is how to organize the computing buffers. We follow the proposal of [7] by grouping the computing resources into physical computing nodes using each group of threads in a buffer to calculate the problems at the collocation points. This incremental calculation by the computing buffers and the thread organization by physical computing nodes allows us to use a similar memory allocation to the one used by the deterministic counterpart. In addition, it minimizes inter-node communications reducing bandwidth problems, and adapts the computing requirements of the RTO problem to the available computational resources.

We organize the remainder of the manuscript as follows. We devote Sect. 2 to reviewing the basis and theoretical background of density-based RTO. Section 3 introduces the parallel AMR technique and the error estimation criteria for coarsening the regions of interest. Section 4 presents the parallel scheme for addressing the RTO problem using parallel computing with distributed memory, detailing the partitioning, solving, incremental calculation of the stochastic moments, regularization, and optimization approach. Section 5 shows the numerical experiments evaluating the performance, feasibility, and scalability of the techniques adopted for taking advantage of parallel multi-core computing in density-based RTO using dynamic AMR techniques. Finally, Sect. 6 presents the conclusion and future works of the proposal.

2 Robust density-based topology optimization

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a complete probability space, and $D \subset \mathbb{R}^d$ is a bounded Lipschitz domain with its boundary decomposed into three disjoint parts $\partial D = \Gamma_u \cup \Gamma_t \cup \Gamma_0$, where Γ_u is the part of the boundary with prescribed displacements, Γ_t is the part of the boundary with traction forces, and Γ_0 is the part of the boundary with traction-free boundary conditions. Consider the linearized elasticity system under random input data

$$\begin{cases} -\nabla\sigma(u(x,\omega)) = p(x,\omega) & \text{in } D \times \Omega \\ u(x,\omega) = \bar{u} & \text{in } \Gamma_u \times \Omega \\ \sigma(u(x,\omega)) \cdot n = \bar{t}(x,\omega) & \text{in } \Gamma_t \times \Omega \\ \sigma(u(x,\omega)) \cdot n = 0 & \text{in } \Gamma_0 \times \Omega, \end{cases} \tag{1}$$

where u is the stochastic displacement field, σ is the Cauchy stress tensor, p and $\bar{t}$ are the body and surface forces, $\bar{u}$ is the prescribed displacement field, and n is the unit outward normal vector to ∂D. Note that the body and surface forces and the displacement field depend on a spatial variable $x \in D$ and a random event $\omega \in \Omega$. The stress tensor σ and the symmetric gradient of the displacement field ε relate by the constitutive equation

$$\mathbb{C}_{ijkl}(x,\omega)\varepsilon_{ij}(u(x,\omega)) = \sigma_{kl}(u(x,\omega)), \tag{2}$$

where $\mathbb{C}_{ijkl}$ represents the fourth-order constitutive tensor. Considering the mechanical design as a body occupying a domain D^m which is part of the reference domain D, we can split the design into two subdomains with the following characteristic function:

$$\Theta(x) = \begin{cases} 1 & x \in D^m \\ 0 & x \in D \setminus D^m. \end{cases} \tag{3}$$

We can formulate the stochastic topology optimization problem under random input data as the minimization of the structural compliance over admissible design and displacement fields subjected to the maximum material allowed, satisfying the equilibrium equation in its weak form as follows:

$$\begin{aligned} \min_{\Theta} J(\Theta,\omega) &= \int_D pu\, dx + \int_{\Gamma_t} \bar{t}u\, d\Gamma \\ \text{s.t. } &: a(\Theta,u,v) = l(v) \quad \forall\, v \in \mathcal{V} \\ &: \mathbb{C}_{ijkl}(\Theta,\omega) = \Theta\, \mathbb{C}^0_{ijkl}(\omega) \\ &: Vol(D^m) = \int_D \Theta\, dx \le V^*, \end{aligned} \tag{4}$$

where $u = u(x,\omega)$, $v = v(x,\omega)$, $\bar{t} = \bar{t}(x,\omega)$, $p = p(x,\omega)$, $\Theta = \Theta(x)$, $\mathcal{V}$ denotes the space of kinematically admissible displacement fields, $\mathbb{C}^0_{ijkl}(\omega)$ is the stiffness tensor for an elastic material, V^* is the volume target considering the pointwise volume fraction $\Theta(x)$ for a black-and-white design, and $a(\cdot,\cdot)$ and $l(\cdot)$ are the bilinear and linear forms, respectively, as follows:

$$\begin{aligned} a(\Theta,u,v) &= \int_D \sigma(u)\, \varepsilon(v) dx \\ &= \int_D \mathbb{C}_{ijkl}(\Theta,\omega)\, \varepsilon_{ij}(u)\, \varepsilon_{kl}(v) dx \\ l(v) &= \int_D pv\, dx + \int_{\Gamma_t} \bar{t}v\, d\Gamma. \end{aligned} \tag{5}$$

The formulation of the stochastic topology optimization problem provides a different solution for each realization of the random event $\omega \in \Omega$. To successfully deal with the problem, we transform the stochastic formulation into a deterministic one. We can then use the conventional optimization algorithms to address the stochastic topology optimization problem under random input data. We adopt the formulation of the RTO as a two-objective optimization problem where we consider the expected value and standard deviation of the structural compliance as a measure of structural robustness. We use a weighted approach to scalarize the multi-objective problem into a single-objective one as follows:

$$\begin{aligned} \min_{\Theta} \mathcal{J}_R(\Theta) &= \mathbb{E}[J(\Theta,\omega)] + \alpha\sqrt{Var[J(\Theta,\omega)]} \\ \text{s.t.}: &\ a(\Theta,u,v) = l(v) \quad \forall\, v \in \mathcal{V} \\ : &\ \mathbb{C}_{ijkl}(\Theta,\omega) = \Theta\, \mathbb{C}^0_{ijkl}(\omega) \\ : &\ Vol(D^m) = \int_D \Theta\, dx \le V^*, \end{aligned} \tag{6}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator, $Var[\cdot]$ stands for the variance operator, and $\alpha \ge 0$ is a weighting parameter balancing the first two stochastic moments of the performance functional, which we obtain as follows:

$$\mathbb{E}[J(\Theta,\omega)] = \int_\Omega J(\Theta,\omega) d\mathbb{P}(\omega), \tag{7}$$

$$\begin{aligned} Var[J(\Theta,\omega)] &= \int_\Omega J^2(\Theta,\omega) d\mathbb{P}(\omega) \\ &- \left(\int_\Omega J(\Theta,\omega) d\mathbb{P}(\omega) \right)^2. \end{aligned} \tag{8}$$

The SIMP method relaxes the integer-based problem (6) by introducing an interpolation scheme that penalizes a continuous density variable $\rho \in [0, 1]$ characterizing composite materials and allows us to use gradient-based approaches in the optimization. The following power-law interpolation function permits us to rewrite the constitutive tensor equation as:

$$\mathbb{C}_{ijkl}(\rho, \omega) = \mathbb{C}^{min}_{ijkl} + \rho^p(\mathbb{C}^0_{ijkl}(\omega) - \mathbb{C}^{min}_{ijkl}), \tag{9}$$

where $\rho = \rho(x)$, p > 1 is the penalization power, and $\mathbb{C}^{min}_{ijkl}$ is the fourth-order constitutive tensor of the soft material. This penalization function relates the design variable $\rho(x)$ and the material tensor $\mathbb{C}_{ijkl}$ in the equilibrium analysis, satisfying $\mathbb{C}_{ijkl}(0, \omega) = \mathbb{C}^{min}_{ijkl}$ and $\mathbb{C}_{ijkl}(1, \omega) = \mathbb{C}^0_{ijkl}(\omega)$. We can select p sufficiently big to penalize intermediate densities. According to [8], we usually require $p \geq 3$ to ensure that we do not violate the Hashin–Shtrikman bounds. We can then state the problem as

$$\begin{aligned}
\min_{\rho(x)} \ & \mathcal{J}_R(\rho) = \mathbb{E}[J(\rho, \omega)] + \alpha\sqrt{Var[J(\rho, \omega)]} \\
\text{s.t.}: \ & a(\rho, u, v) \ = \ l(v) \quad \forall\, v \in \mathcal{V} \\
: \ & \mathbb{C}_{ijkl}(\rho, \omega) = \mathbb{C}^{min}_{ijkl} + \rho^p(\mathbb{C}^0_{ijkl}(\omega) - \mathbb{C}^{min}_{ijkl}) \\
: \ & Vol(D^m) = \int_D \rho \, dx \leq V^*, \ \ \rho \in [0, 1],
\end{aligned} \tag{10}$$

where $\rho = \rho(x)$ is the design variable (constant within each finite element) ranging from solid ($\rho = 1$) to void ($\rho = 0$). However, density-based topology optimization is prone to numerical instabilities due to checker-board patterns appearing as penalizing intermediate material densities, mesh-dependency as refining the tessellation of the continuum, and the presence of local minima in the design space [51]. We can introduce a density measure $\tilde{\rho} = \tilde{\rho}(x)$ that tends to regularize the problem addressing these numerical instabilities. Besides, we use projection techniques to project the filtered designs $\tilde{\rho}$ into solid/void space $\bar{\tilde{\rho}} = \bar{\tilde{\rho}}(x)$, which produces designs with a clear physical interpretation [62]. We can then state the problem as

$$\begin{aligned}
\min_{\bar{\tilde{\rho}}} \ & \mathcal{J}_R(\bar{\tilde{\rho}}) = \mathbb{E}[J(\bar{\tilde{\rho}}, \omega)] + \alpha\sqrt{Var[J(\bar{\tilde{\rho}}, \omega)]} \\
\text{s.t.}: \ & a(\bar{\tilde{\rho}}, u, v) \ = \ l(v) \quad \forall\, v \in \mathcal{V} \\
: \ & \mathbb{C}_{ijkl}(\bar{\tilde{\rho}}, \omega) = \mathbb{C}^{min}_{ijkl} + \bar{\tilde{\rho}}^p(\mathbb{C}^0_{ijkl}(\omega) - \mathbb{C}^{min}_{ijkl}) \\
: \ & Vol(D^m) = \int_D \rho \, dx \leq V^*, \ \ \rho \in [0, 1],
\end{aligned} \tag{11}$$

where $\bar{\tilde{\rho}} = \bar{\tilde{\rho}}(x)$ is the projected and regularized design field.

Assuming finite-dimensional noise and ignoring the uncertainty in the constitutive tensor $\mathbb{C}_{ijkl}$, we can state that the random input data of the stochastic partial differential equation (PDE) of (1) depends on a finite number N of real-valued random variables $\{Y_n\}_{n=1}^N$ mapping $J(\bar{\tilde{\rho}}, y) : \Omega \rightarrow \mathbb{R}$. The problem can be then stated as

$$\begin{aligned}
\min_{\bar{\tilde{\rho}}} \ & \mathcal{J}_R(\bar{\tilde{\rho}}) = \mathbb{E}[J(\bar{\tilde{\rho}})] + \alpha\sqrt{Var[J(\bar{\tilde{\rho}})]} \\
\text{s.t.}: \ & a(\bar{\tilde{\rho}}, u, v) \ = \ l(v, y) \quad \forall\, v \in \mathcal{V}, y \in \Lambda \\
: \ & \mathbb{C}_{ijkl}(\bar{\tilde{\rho}}) = \mathbb{C}^{min}_{ijkl} + \bar{\tilde{\rho}}^p(\mathbb{C}^0_{ijkl} - \mathbb{C}^{min}_{ijkl}) \\
: \ & Vol(D^m) = \int_D \rho \, dx \leq V^*, \ \ \rho \in [0, 1],
\end{aligned} \tag{12}$$

where $\Lambda = Y(\Omega) \subset \mathbb{R}^N$ and the mean and variance of the stochastic objective function are as follows:

$$\mathbb{E}[J(\bar{\tilde{\rho}})] = \int_\Lambda J(\bar{\tilde{\rho}}, y)\, \phi(y)\, dy, \tag{13}$$

$$\begin{aligned}
Var[J(\bar{\tilde{\rho}})] = & \int_\Lambda J^2(\bar{\tilde{\rho}}, y)\, \phi(y)\, dy \\
& - \left(\int_\Lambda J(\bar{\tilde{\rho}}, y)\, \phi(y)\, dy \right)^2,
\end{aligned} \tag{14}$$

where $\phi(y)$ is the joint probability density function of $\{Y_n\}_{n=1}^N$.

We can approximate the integrals of the stochastic moments of (13) and (14) equations employing numerical quadrature as follows:

$$\mathbb{E}[J(\bar{\tilde{\rho}})] \approx \sum_{i=1}^N \Phi_i\, J(\bar{\tilde{\rho}}, y_i), \tag{15}$$

$$\begin{aligned}
Var[J(\bar{\tilde{\rho}})] \approx & \sum_{i=1}^N \Phi_i\, J^2(\bar{\tilde{\rho}}, y_i) \\
& - \left(\sum_{i=1}^N \Phi_i\, J(\bar{\tilde{\rho}}, y_i) \right)^2,
\end{aligned} \tag{16}$$

where the points $\{y_i\}_{i=1}^N$ and the weights $\{\Phi_i\}_{i=1}^N$ depend on the one-dimensional rule and the selection of tensors. We use the Smolyak sparse grid [53] collocation method for the numerical approximation of the multi-dimensional integrals. Such a technique significantly reduces the number of collocation points while preserving a high level of accuracy compared to the full tensor product quadrature [43, 54]. The Smolyak formulas are linear combinations of the full tensor product quadrature formulas. We refer the reader to [18] for the details of the Smolyak quadrature formula $\mathcal{A}(\ell, N)$ with the level ℓ that is independent of the random dimension N and the generation of the corresponding $\{y_i\}_{i=1}^N$ points and $\{\Phi_i\}_{i=1}^N$ weights used by (15) and (16).

We use the density measure introduced in [10], the so-called density filter, to regularize the density field $\rho(x)$. In particular, we perform the filtering operation by the convolution product of the filter F and density ρ functions as follows:

$$\tilde{\rho}(x) = (F * \rho)(x) = \int F(x - x')\rho(x')dx' \tag{17}$$

$$\int_{B_R} F(x)dx = 1, \tag{18}$$

where B_R denotes the open ball of radius $R > 0$ and the filter function satisfies $F \geq 0 \; \forall \, x \in B_R$. The filter requires that the volume is the same for the filtered and unfiltered fields, and thus, we can impose the volume constraint on the unfiltered field [30]. We usually replace the expression (17) by

$$\tilde{\rho} = \frac{\sum\limits_{i \in N_e} w(x_i)v_i\rho_i}{\sum\limits_{i \in N_e} w(x_i)v_i}, \tag{19}$$

where $\tilde{\rho}$ is the filtered design field, N_e is the neighborhood set of elements lying within the radius R, $w(\cdot)$ is the weighting function $w(x_i) = R - \|x_i - x'\|$, and v_i is the volume of each element associated with the design variable. This filtering technique allows us to cope with numerical problems in topology optimization, in particular, checker-board patterns and mesh-dependent designs. Besides, [10] proved the existence of the topology optimization solution mathematically using such a filter in topology optimization.

We also use the volume-preserving Heaviside filter proposed by [62] to prevent blurred boundaries in the material interface projecting $\tilde{\rho}$ to full or empty design variables. In particular, it projects the filtered density values $\tilde{\rho}$ above a threshold η to solid and the ones below such a threshold to void using the following smooth function:

$$\bar{\tilde{\rho}} = \begin{cases} \eta[e^{-\beta(1-\frac{\tilde{\rho}}{\eta})} - (1 - \frac{\tilde{\rho}}{\eta})e^{-\beta}], \\ \qquad\qquad 0 \leq \tilde{\rho} \leq \eta \\ (1-\eta)[1 - e^{-\beta\frac{(\tilde{\rho}-\eta)}{1-\eta}} + \frac{\tilde{\rho}-\eta}{1-\eta}e^{-\beta}] + \eta, \\ \qquad\qquad \eta \leq \tilde{\rho} \leq 1, \end{cases} \tag{20}$$

where the projection parameter β allows us to control the smooth function. We can obtain a similar projected field with the threshold value $\eta = 0$ than using the Heaviside step filter introduced by [20], ensuring a minimum length scale on the solid phase. The threshold value $\eta = 1$ performs similar filtering to the modified Heaviside filter introduced by [50], giving rise to a minimum length scale on the void phase. We can write the expression (20) as

$$\bar{\tilde{\rho}} = \frac{\tanh(\ \beta\eta\ ) + \tanh(\ \beta(\tilde{\rho} - \eta)\ )}{\tanh(\ \beta\eta\ ) + \tanh(\ \beta(1 - \eta)\ )}, \tag{21}$$

which provides an efficient alternative to calculate the projection avoiding the conditional statements [59].

We can derive the sensitivity of the objective function (12) to the design variable $\rho(x)$ using the chain rule as

$$\frac{\partial \mathcal{J}_R(\bar{\tilde{\rho}})}{\partial \rho} = \frac{\partial \mathcal{J}_R(\bar{\tilde{\rho}})}{\partial \bar{\tilde{\rho}}} \frac{\partial \bar{\tilde{\rho}}}{\partial \tilde{\rho}} \frac{\partial \tilde{\rho}}{\partial \rho}, \tag{22}$$

obtaining the different terms are as follows:

$$\begin{aligned} \frac{\partial \mathcal{J}_R(\bar{\tilde{\rho}})}{\partial \bar{\tilde{\rho}}} &= \sum_{i=1}^{N} \Phi_i \frac{\partial J(\bar{\tilde{\rho}}, y_i)}{\partial \bar{\tilde{\rho}}} \\ &\quad + \frac{\alpha}{\sqrt{Var[J(\bar{\tilde{\rho}}, y_i)]}} \left[\sum_{i=1}^{N} \Phi_i \, J(\bar{\tilde{\rho}}, y_i) \frac{\partial J(\bar{\tilde{\rho}}, y_i)}{\partial \bar{\tilde{\rho}}} \right. \\ &\quad \left. - \sum_{i=1}^{N} \Phi_i \, J(\bar{\tilde{\rho}}, y_i) \sum_{i=1}^{N} \Phi_i \frac{\partial J(\bar{\tilde{\rho}}, y_i)}{\partial \bar{\tilde{\rho}}} \right] \end{aligned} \tag{23}$$

$$\frac{\partial \bar{\tilde{\rho}}}{\partial \tilde{\rho}} = \frac{\beta(\ \mathrm{sech}\ (\ \beta(\tilde{\rho} - \eta))\ )^2}{\tanh(\ \beta\eta\ ) + \tanh(\ \beta(1 - \eta)\ )} \tag{24}$$

$$\frac{\partial \tilde{\rho}}{\partial \rho} = \frac{w(x_i)v_i}{\sum\limits_{i \in N_e} w(x_i)v_i}. \tag{25}$$

We update the design variables using the method of moving asymptotes (MMA) proposed by [57] for its excellent parallel scalability [1]. The MMA method is suitable for addressing inequality-constrained optimization problems, such as the formulation of (12), by generating and solving a series of approximate convex subproblems instead of the original non-linear problem. The algorithm stops when we reach the maximum number of iterations or when the change variable $||\rho_{e_{k+1}} - \rho_{e_k}||$ and the difference in the objective function $||f_{k+1} - f_k||$ falls below a prescribed value.

3 Adaptive mesh refinement and coarsening

Let G be an element containing *vertices* and *edges* in two-dimensional cases, including *faces* in three-dimensional cases. We define a mesh $\{G_i\}_{i=1}^{N_g}$ as the union of the N_g elements, such that $D = \cup_i G_i$ is a connected and bounded region. Let us denote boundary elements as ∂G and interior elements as $\langle G \rangle$. Assuming interior elements do not intersect, $\langle G_i \rangle \cap \langle G_j \rangle = \emptyset$ $\forall \ i \neq j$, we state that a mesh is a *conforming* grid if the set $\partial G_i \cap \partial G_j \; \forall \, i \neq j$ is a shared vertex, shared edge, shared face, or an empty set. We define a *hanging node* as a vertex lying in the interior of an edge or in a face in three-dimensional cases. We call *non-conforming* meshes the ones containing at least one *hanging node*.

Refinement of one *parent* element G_i consists of replacing it with at least two *child* elements G_{ik}, such that $G_i = \cup_k G_{ik}$. We remove parent elements after refining them but storing the links to the child elements in a *refinement tree*. This information facilitates the coarsening by reintroducing the parent element and removing the child elements. We say that a non-conforming mesh is consistent if lower-dimensional mesh entities (vertices, edges, and faces) are either identical or a proper subset of the other. We refer to the smaller entities as slaves and the larger ones containing other entities as masters. Otherwise, we call them conforming entities.

We adopt the method proposed by [58] for non-conforming AMR of unstructured meshes. This method cast the elimination of constrained degrees of freedom (DOFs) as a form of variational restriction decoupling the AMR method and the governing equation, facilitating the integration of the AMR functionalities in the physics simulation. We can discretize the weak variational problem of (12) constructing an approximate finite-dimensional subspace $\mathcal{V}_h \subset \mathcal{V}$ on the $\{G_i\}_{i=1}^{N_g}$ mesh assembling the stiffness matrix $A(\bar{\bar{\rho}})$ and the vector of load forces $f(y_i)$, such that

$$v_h{}^T A(\bar{\bar{\rho}}) u_h = v_h{}^T f(y_i) \quad \forall v_h \in \mathbb{R}^d, \tag{26}$$

where d is the dimension of the problem and y_i is the corresponding stochastic collocation point. In non-conforming meshes, we have a larger partially conforming space $\widehat{\mathcal{V}}_h \supset \mathcal{V}_h$ with $\widehat{\mathcal{V}}_h \in \mathbb{R}^{\hat{d}}$ and $\hat{d} > d$ since slave entities containing hanging nodes have DOFs independent of master DOFs. We can restore the conformity in the interfaces containing hanging nodes constraining slave entities (slave DOFs) by interpolating the finite-element functions of their masters. We obtain the solution vector $\hat{u}_h$ in the partially conforming space $\widehat{\mathcal{V}}_h$ as

$$\hat{u}_h = \begin{pmatrix} u_h \\ w_h \end{pmatrix} \quad u_h \in \mathbb{R}^d,\ w_h \in \mathbb{R}^{\hat{d}-d}, \tag{27}$$

where w_h represents all slave DOFs that we can evaluate by the linear interpolation $w_h = w u_h$ using the interpolation operator w. We can write (27) as

$$\hat{u}_h = P_c u_h, \quad \text{with } P_c = \begin{pmatrix} I \\ w \end{pmatrix}, \tag{28}$$

where I is the identity matrix and P_c is the conforming prolongation matrix. We can assemble the stiffness matrix $\hat{A}(\bar{\bar{\rho}})$ and load vector $\hat{f}(y_i)$ in the space $\widehat{\mathcal{V}}_h$ obviating the hanging nodes in the mesh. The corresponding linear system $\hat{A}(\bar{\bar{\rho}})\hat{u}_h = \hat{f}(y_i)$ provides a non-conforming solution where the slave DOFs are not constrained. Using the expression (28), the variational formulation of equation (26) on v_h becomes

$$v_h^T P_c^T \hat{A}(\bar{\bar{\rho}}) P_c u_h = v_h^T P_c^T \hat{f}(y_i) \quad v_h \in \mathbb{R}^d. \tag{29}$$

Thus, we can solve the smaller problem $P_c^T \hat{A}(\bar{\bar{\rho}}) P_c u_h = P_c^T \hat{f}(y_i)$ for the corresponding stochastic collocation points y_i where the slave DOFs are eliminated and then prolonging u_h to obtain the conforming solution $\hat{u}_h \in \widehat{\mathcal{V}}_h$. We remark that factoring P_c out of the AMR governing equation facilitates the incorporation of AMR in different problems. The challenge is then the efficient construction of this P_c operator. We refer to the work of [58] and the implementation of the AMR technique using the mfem [39] and hypre [27] libraries for the details of the serial and parallel construction of the P_c operator.

The AMR technique requires an error estimator to determine the regions of interest. We choose the areas with void material as regions of interest, because they contribute significantly to the overall computational cost but little to the accuracy of the optimized design. Besides, such areas contribute meaningfully to the ill-conditioning of the elasticity system to solve using FEA. Thus, the error estimators identifying sets of design variables with void material for coarsening them can meaningfully increase the performance [61]. Since we focus on finding the same solution as the one obtained in a fine mesh but at a cheaper computational cost, we use the following error estimator:

$$\varepsilon_v = \frac{\int_{D_{rt}} \bar{\bar{\rho}}(x) dx_{rt}}{\int_{D_{rt}} dx_{rt}}, \tag{30}$$

where ε_v is the error estimator based on the design variable, and $D_{rt} \subset D$ is the part of the domain in the different levels of the *refinement tree*: the parent element for refinement and the set of child elements for coarsening to the parent element. We can choose the threshold $\varepsilon_v \leq E^{min} + \varepsilon$, with E^{min} the elastic modulus used for the constitutive tensor $\mathbb{C}_{ijkl}^{min}$ of void material and ε a small value for coarsening the void elements by thresholding.

4 Parallel framework

Parallel computing approaches divide complex problems into smaller and more workable subproblems by distributing their computation across computational resources. The performance of such methods depends on the workload of each subdomain and the communications needed between them. We can optimize the latter using efficient partitioning techniques. We do this by minimizing the number of interface elements and enforcing contiguous partitioning in the tessellation [26]. One key advantage of density-based topology optimization methods is that they operate on a fixed mesh. Thus, the domain partitioning and communications between subdomains are the same during the optimization.

Figure 1 shows the flowchart of the parallel framework using distributed memory for density-based RTO using AMR techniques. This framework aims to increase the performance addressing the computational bottleneck of RTO formulation: the calculation of the stochastic moments of the structural compliance of (12) using sparse grid SCMs. Such a calculation requires the evaluation of N collocation points $\{y_i\}_{i=1}^{N}$ needing an intensive use of computational resources, including computing time and large memory allocation. We evaluate the deterministic problems at the collocation points y_i and their contribution to the sensitivity functional in a coarse mesh, which requires the projection of the design variables to penalize the constitutive tensor $\mathbb{C}_{ijkl}$ of the corresponding deterministic models.

First, we tessellate the model using the optimization criteria mentioned above. We then initialize the communications needed to perform the distributed operations: the coarsening and refinement, the regularization of design variables, and the solving of the system of equations of the deterministic problems at the collocation points y_i using FEA. We use computing buffers to organize and adapt the parallel computing resources to the RTO problem. This strategy permits us to exploit the embarrassing parallelism of SCMs as it increases the number of computing nodes. It also allows computing buffers to reuse information and operators from the FEA of previous collocation points.

Computing buffers make intensive use of intra-node communications during the analysis of the deterministic problems using FEA and the calculation of the contribution to the stochastic sensitivity of the objective function. We also use the intra-node communications on one computing buffer to regularize the design variables and update them using the parallel implementation of the MMA optimization algorithm proposed by [1]. We minimize inter-node communications to gather contributions from computing nodes to calculate the stochastic objective function and sensitivities, which reduces bandwidth problems significantly.

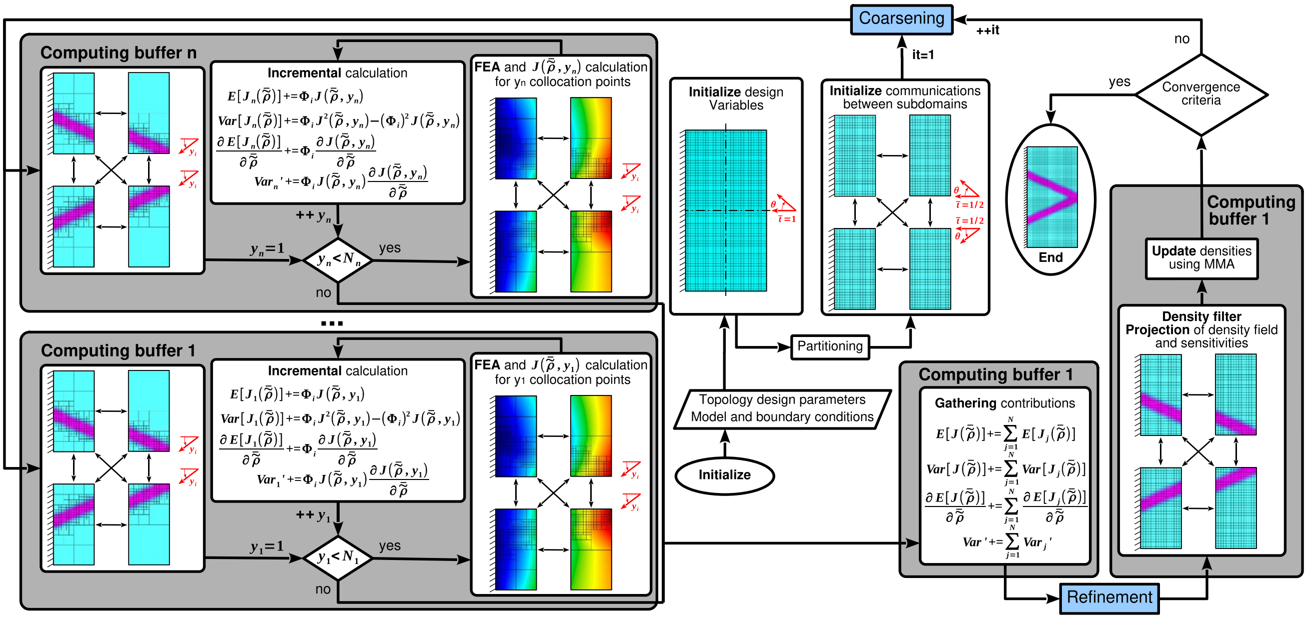
The following sections present the ingredients adopted to exploit the different levels of parallelism of the RTO formulation. Section 4.1 describes the techniques adopted for parallel solving the elasticity system of the deterministic problems at the collocation points y_i using FEA. Section 4.2 introduces the strategy adopted using computing buffers to adapt the available resources to the RTO problem. We also describe the incremental calculation approach for computing the stochastic moments of the objective function and the contribution of each deterministic problem at the collocation points y_i to the functional sensitivity. Section 4.3 details the parallel method for regularizing the design variables.

4.1 Parallel solving

Let us consider the conforming linear system of equations to solve the problem stated in (29) as

$$Au_h = b, \tag{31}$$

where $A = P_c^T \hat{A}(\bar{\tilde{\rho}}) P_c \in \mathbb{R}^{n_u \times n_u}$ is the coefficient matrix, $b = P_c^T \hat{f}(y_i) \in \mathbb{R}^{n_u \times 1}$ is the right-hand side vector for the corresponding collocation point y_i, $u_h \in \mathbb{R}^{n_u \times 1}$ is the solution vector, and n_u is the number of unknowns. We adopt



Coarsening
++it
it=1
no
yes
Convergence criteria
Computing buffer n
Incremental calculation
$E[J_n(\bar{\tilde{\rho}})] += \Phi_i J(\bar{\tilde{\rho}}, y_n)$
$Var[J_n(\bar{\tilde{\rho}})] += \Phi_i J^2(\bar{\tilde{\rho}}, y_n) - (\Phi_i)^2 J(\bar{\tilde{\rho}}, y_n)$
$\frac{\partial E[J_n(\bar{\tilde{\rho}})]}{\partial \bar{\tilde{\rho}}} += \Phi_i \frac{\partial J(\bar{\tilde{\rho}}, y_n)}{\partial \bar{\tilde{\rho}}}$
$Var_n' += \Phi_i J(\bar{\tilde{\rho}}, y_n) \frac{\partial J(\bar{\tilde{\rho}}, y_n)}{\partial \bar{\tilde{\rho}}}$
FEA and $J(\bar{\tilde{\rho}}, y_n)$ calculation for yn collocation points
y_i
$++y_n$
$y_n=1$
$y_n<N_n$
yes
no
...
Computing buffer 1
Incremental calculation
$E[J_1(\bar{\tilde{\rho}})] += \Phi_i J(\bar{\tilde{\rho}}, y_1)$
$Var[J_1(\bar{\tilde{\rho}})] += \Phi_i J^2(\bar{\tilde{\rho}}, y_1) - (\Phi_i)^2 J(\bar{\tilde{\rho}}, y_1)$
$\frac{\partial E[J_1(\bar{\tilde{\rho}})]}{\partial \bar{\tilde{\rho}}} += \Phi_i \frac{\partial J(\bar{\tilde{\rho}}, y_1)}{\partial \bar{\tilde{\rho}}}$
$Var_1' += \Phi_i J(\bar{\tilde{\rho}}, y_1) \frac{\partial J(\bar{\tilde{\rho}}, y_1)}{\partial \bar{\tilde{\rho}}}$
FEA and $J(\bar{\tilde{\rho}}, y_1)$ calculation for y1 collocation points
$++y_1$
$y_1=1$
$y_1<N_1$
yes
no
Initialize design Variables
θ
$\bar{t}=1$
Initialize communications between subdomains
$\bar{t}=1/2$
End
Computing buffer 1
Update densities using MMA
Density filter
Projection of density field and sensitivities
Partitioning
Topology design parameters
Model and boundary conditions
Initialize
Computing buffer 1
Gathering contributions
$E[J(\bar{\tilde{\rho}})] += \sum_{j=1}^{N} E[J_j(\bar{\tilde{\rho}})]$
$Var[J(\bar{\tilde{\rho}})] += \sum_{j=1}^{N} Var[J_j(\bar{\tilde{\rho}})]$
$\frac{\partial E[J(\bar{\tilde{\rho}})]}{\partial \bar{\tilde{\rho}}} += \sum_{j=1}^{N} \frac{\partial E[J_j(\bar{\tilde{\rho}})]}{\partial \bar{\tilde{\rho}}}$
$Var' += \sum_{j=1}^{N} Var_j'$
Refinement

Fig. 1 Flowchart of the parallel adaptive robust density-based topology optimization framework

the parallel version of the compressed sparse row (ParCSR) format for the distributed representation of the coefficient matrix [17] across n_p computing processes. We use a conjugate gradient solver with the ParCSR format using the standardized and portable MPI for communications. We use this Krylov subspace method for solving the deterministic problems at the collocation points y_i of (12) using multicore computing. We commonly use multigrid approaches in mechanical problems as efficient preconditioners of iterative solvers to obtain the response of the system of equations of elasticity.

We can broadly classify multigrid approaches into two groups [56]: GMG and AMG. GMG uses the geometry of the problem to define the coefficient matrices at the different multigrid l levels and the transfer operators between them. On the other hand, AMG methods only use the information available in the matrix of coefficients, allowing their use as a "black-box" function in finite-element codes. Multigrid approaches have two stages: setup and solving. The former aims to construct the coefficient matrices at the multigrid l levels and the proper transfer operators between them, whereas the latter performs the recursive prolongation and restriction operations between multigrid levels. In this work, we use a V-cycle for the solving stage. The construction of the transfer operators in the setup stage is hard to parallelize, whereas the solving stage consists of matrix–vector products prolonging and restricting between multigrid levels, which we can run in parallel. In this work, we use multigrid methods as preconditioners of the parallel conjugate gradient solver to evaluate the collocation points y_i of the RTO. In particular, we use the implementation BoomerAMG [23] of the classical AMG method [46] for preconditioning the iterative solver in the adaptive coarse mesh.

The setup stage of GMG requires the model geometry to construct the coefficient matrix $A \in \mathbb{R}^{n_u \times n_u}$, the number of pre- and post-smoothing steps φ_1 and φ_2, and the smoother parameters. In this work, the experiments using a GMG preconditioner use Jacobi smoothing, which requires tuning the damping factor w_s. On the other hand, the AMG setup requires the coefficient matrix $A \in \mathbb{R}^{n_u \times n_u}$, a strength threshold θ for the coarsening, and a truncation factor τ for constructing the interpolation operator. The strength threshold θ [63] allows us to calibrate the strong dependence criterion for grouping and separating the coefficients. We use the Hybrid Modified Independent Set (HMIS) algorithm for parallel coarsening. This coarsening approach combines the Parallel Modified Independent Set (PMIS) algorithm [55] with a one-pass Ruge-Stüben scheme. We also require the truncation factor τ to remove the coefficients of the interpolation operator smaller than such a factor and thus avoid an excessive number of nonzero coefficients. We use the parallel implementation of sparse approximate inverse preconditioner ParaSails [13] to reduce the oscillatory error components. This smoother uses a priori sparsity patterns and least-squares minimization.

4.2 Computing buffers for parallel incremental calculation

The parallel evaluation of the deterministic problems at the y_i collocation points is embarrassingly parallelizable. However, the concurrent memory allocation of the $\{y_i\}_{i=1}^{N}$ finite-element models can require a large amount of RAM. We also have to mention that the intensive use of communications can give rise to bandwidth problems. For these reasons, the embarrassingly parallel computing approach often requires a computer cluster or supercomputer to address relatively large RTO problems [7].

We propose using computing buffers [25] for sequentially calculating the corresponding contribution of sets of deterministic problems at the collocation points y_i. Such computing buffers calculate each collocation point $\{y_i\}_{i=1}^{N}$ using parallel computing. We adopt an implicit approach to distribute the N collocation points between the n available computing buffers. The computing buffers self-assign a similar number of stochastic points N_i, with $i = \{1, \dots, n\}$, depending on the global ID of the computing buffer and the total number of computing buffers n to calculate the collocation points of (12).

We calculate the structural response of each collocation point y_i in the coarse mesh using the parallel strategy presented above with distributed memory using the MPI standard. The computing buffers store the contribution of each collocation point y_i for the calculation of the first two stochastic moments of structural compliance: $\mathbb{E}[\bar{\bar{\rho}}]$ and $Var[\bar{\bar{\rho}}]$. They also store the results of some operations needed to calculate the contribution of $\partial\mathcal{J}_R(\bar{\bar{\rho}})/\partial\bar{\bar{\rho}}$ to the sensitivity of the functional of (22): $\sum_{i=1}^{N} \Phi_i \dfrac{\partial J(\bar{\bar{\rho}}, y_i)}{\partial\bar{\bar{\rho}}}$ and $\sum_{i=1}^{N} \Phi_i \, J(\bar{\bar{\rho}}, y_i) \dfrac{\partial J(\bar{\bar{\rho}}, y_i)}{\partial\bar{\bar{\rho}}}$. These amounts, along with the $\mathbb{E}[\bar{\bar{\rho}}]$, allow us to calculate $\partial\mathcal{J}_R(\bar{\bar{\rho}})/\partial\bar{\bar{\rho}}$ of (23) by adding the sequential calculation of the set of collocation points per computing node. The master node gathers the contributions from computing nodes to calculate the stochastic objective function and sensitivities in the coarse mesh. We then project the stochastic objective functional and sensitivity to the fine mesh for regularizing and updating the design field.

This strategy permits us to adapt the parallel computing resources to the RTO problem, allowing us to address the problem even with modest computational resources. It supports running the RTO problem using only one computing node and the embarrassingly parallel execution using one computing node per collocation point, as in [7]. The organization of computing resources is of paramount importance to maximize performance. Figure 2 depicts the computing thread organization of each computing buffer, grouping them by physical computing cores. This organization uses the system bus for communications between the threads of the computing buffer (intra-node communications) and the bus network for communications between computing nodes (inter-node communications). We reduce inter-node communications to gather the resulting contributions of computing nodes to calculate the stochastic objective function and sensitivities, preventing bandwidth problems.

Adopting the strategy of using one computing buffer per computing node, we use a similar memory allocation per computing node to address the deterministic counterpart. Besides, it presents some features that permit us to increase the performance of RTO under uncertain loads. We can use the same distributed assembled matrix of coefficients A for calculating all the deterministic problems at collocation points y_i. Besides, the setup stage of the multigrid approach used as a preconditioner of the Krylov space solvers is similar for all the collocation points y_i in a topology optimization iteration. We have to remark that the construction of the transfer operators in the AMG or GMG setup stage is hard to parallelize, and thus, the reuse of such structures usually improves the performance significantly. We also can reuse the solution of the previously evaluated collocation point as the initial seed of the iterative solver, which is rewarding by accelerating the iterative solver convergence in topology optimization [24].

Finally, we have to remark that we use dynamic AMR in each topology optimization iteration. The coarsening and refinement operations of the AMR technique have a computational cost. We usually have to find a trade-off between such a cost and the increasing performance in the operations performed in the coarse mesh. The sequential calculation of collocation points y_i in the computing buffer benefits from performing coarsening and refinement operations only once per topology optimization iteration, improving the performance as we increase the number of collocation points evaluated in such computing buffers.

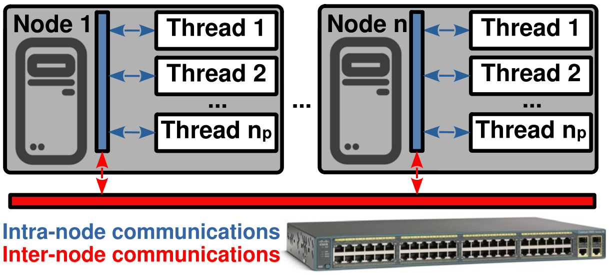


Fig. 2 Grouping of computing threads

4.3 Parallel regularization

The density filter [10] regularizes the dependence of design variables as the weighted average of the convolution operator of (17). The convolution operator requires the design variables of the elements surrounding each finite element within the radius of the open ball B_R defined in (18). We also can refer to the convolution operator as a conic filter. We have to consider the values of the design variables in the different layers of the *refinement tree* of neighbor elements by applying the conic filter to the density field in the coarse mesh. However, searching the neighbor elements in different layers has a high computational cost in distributed-memory parallel implementations. Besides, the resulting search in the layers of the *refinement tree* is different during the iterations of the topology optimization, which requires updating the communications needed by the density filter, increasing the computational cost meaningfully. Since our proposal performs the filtering in the fine mesh, the search of the neighbor elements is only performed in the initialization of the topology optimization and is not needed to evaluate the different layers. Nevertheless, the parallel implementation of the conic filter in distributed-memory systems requires the intensive use of communications for sharing design variables between subdomains.

Figure 3 shows the problem of elements close to the subdomain border in the parallel implementation of the conic filter. The communications required by each design variable depend on the domain partitioning, and we only need to share information between close subdomains. The full circles in the center of elements represent the design variables to regularize. We detail the communications used for filtering two design variables, where we depict the local and remote design variables needed to perform the convolution operation using gray and empty circles, respectively. We also show the communications required between subdomains to compute (19). Since the performance of this operation is of paramount importance to achieve an efficient implementation of density-based RTO, we search the information about the neighbor elements and the communications needed by each design variable as a preprocessing step.

We store the information needed for regularizing each design variable using hash tables. These hash tables include the following data by design variable: element indexes in the same subdomain within the radius R defined in (17), element indexes in a different subdomain inside such a radius R, and the Euclidean distance to the design variables inside the radius R. We can then use point-to-point connections linking the computing processors during the regularization. The use of point-to-point communications facilitates scalability. Gathering the information for parallel regularization increases the

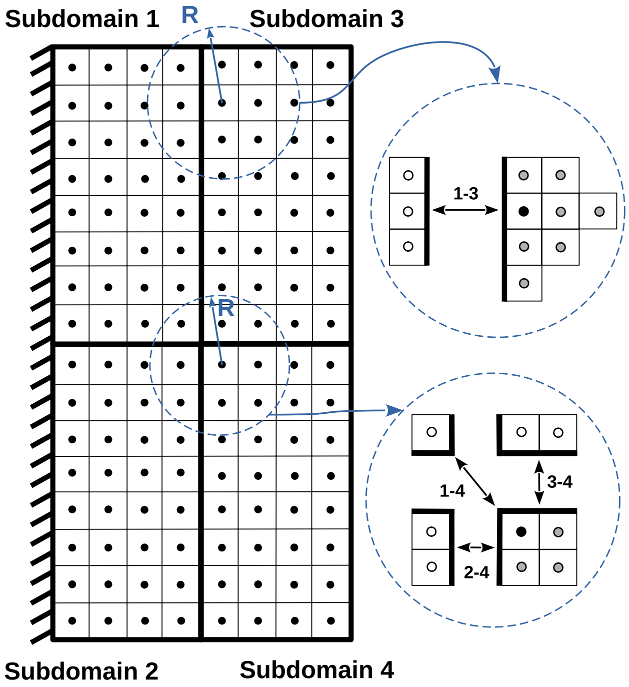


Fig. 3 Parallel regularization using the conic filter including communications between subdomains

computational performance at the cost of increasing the memory requirements to cache the hash tables.

Alternatively, we could use a reaction–diffusion PDE to regularize the design variables [30]. This technique requires determining the relationship between the length scale parameter of the PDE and the radius R of (18) in the conic filter. We have to remark that the geometric complexity increases as $R \to 0$ [47], requiring a small length scale parameter in the PDE, which can induce fluctuations in the resulting design. When this problem appears, it is hard maintaining the filtered design variables in the range [0, 1]. We can solve the PDE using similar parallel iterative techniques to those used for solving the elasticity system of equations in Sect. 4.1. Therefore, it is a scalable approach. However, we require tolerances of a higher order of magnitude to the elastic modulus E^{min} for void material to keep the robustness of the dynamic AMR approach based on the thresholding of density variables presented in Sect. 3. Such tolerances for the iterative solver can increase the computational cost meaningfully. For these reasons, we consider that the regularization using a PDE is a more suitable option for relatively large R values.

5 Numerical experiments

The parallel framework of density-based RTO using AMR schemes aims to reduce the computational bottleneck of the optimization, the numerical approximation of the multidimensional integrals using sparse grid SCM evaluating each collocation point using FEA. We test the performance of preconditioning using AMG and GMG methods during the optimization with the incremental calculation of the stochastic moments of the structural compliance obviating AMR approaches. We also check the improvement in performance using AMR strategies in two- and three-dimensional topology optimization problems. Besides, we evaluate the performance using a different number of computing buffers. We present the cumulative wall-clock time for the RTO process to show the computing benefits while providing similar results to obviating the use of AMR techniques.

We run the numerical experiments using two computational systems: one workstation and two computing nodes connected with a 10 Gigabit Ethernet network working as a computer cluster. The former allows us to evaluate the performance using a modern workstation with modest computational resources, whereas the latter shows the scalability increasing the parallel computational resources. We equip the former workstation using an AMD Ryzen$^{\text{TM}}$ 9 3950X CPU with 16 cores (32 simultaneous multithreading) at 3.5 GHz (turbo core speed of 4.7 GHz) and 64GB of RAM, whereas we use two Intel Xeon E5-2687W v4 CPUs with 12 cores (24 hyper-threading) at 3.0 GHz (turbo boost speed of 3.5 GHz) per CPU connected with dual LGA2011-v3 and 256GB of RAM per computing node in the computer cluster. Thus, the computing nodes can run 24 processes in parallel.

The battery of experiments consists of four topology optimization problems using structured meshes. In particular, the experiments consist of the topology optimization of a short cantilever in two and three dimensions, a 2D Michell-type structure, and a 3D double-hook structure. We parameterize the geometry of all the experiments using $H = 1$. All the experiments use a tolerance 10^{-6} for the parallel conjugate gradient method we use for solving the deterministic problems at collocation points y_i using FEA. We use a penalization power $p = 3$ for the power-law interpolation function of (12). We set Young's modulus $E^0 = 1$ for the constitutive tensor $\mathbb{C}^0_{ijkl}$ of solid material and $E^{min} = 10^{-9}$ for the constitutive tensor $\mathbb{C}^{min}_{ijkl}$ of void material. We use an error estimator $\varepsilon_v = 2E^{min}$ to determine the void regions. The coarsening of void areas allows us to obtain similar solutions to the one obtained in the fine mesh but at a cheaper computational cost.

The numerical experiments using the AMG preconditioner use a strength threshold $\theta = 0.5$ for the coarsening, a truncation factor $\tau = 0.0$ (no truncation) for constructing the interpolation operator, and the HMIS algorithm for parallel coarsening. The numerical experiments using the GMG preconditioning use the damping factor w_s for the Jacobi smoothing that we calibrate to ensure convergence in the different experiments [38]. Both multigrid approaches use $\varphi_1 = \varphi_2 = 4$ pre- and post-smoothing steps in the preconditioning of the iterative solver. We follow a continuation strategy for the β parameter of the projection of (21), following a heuristic strategy for each experiment.

We solve the experiments using multi-core computing with distributed memory with standardized and portable MPI mechanisms. We regularize and update the design variables using the fine mesh, whereas the computing buffers use the dynamically coarse mesh for the incremental evaluation of the collocation points y_i using FEA and calculating contributions to the functional sensitivity. This strategy requires adaptive coarsening and refinement up to the fine mesh at all topology optimization iterations. We test the increment in performance of the whole process with the best reference implementation without using AMR. We present below the numerical results of the four density-based RTO experiments evaluating the computational aspects of the parallel AMR strategy using multi-core computation.

5.1 Short-cantilever experiment

The two-dimensional short-cantilever experiment evaluates the qualitative effects of introducing uncertain loads in the optimal design. Figure 4 shows the geometric configuration of the short cantilever using the H parameter. It also shows the boundary conditions anchoring the left edge of the cantilever and applying a unit force $\overline{t} = 1$ at the middle of the right edge with uncertainty in the bearing θ centered in the horizontal symmetry axis. We set the target volume $V^* = 15\%$ of the initial design domain. We use a Gauss–Hermite quadrature rule with a level $\ell = 3$ to approximate the multi-dimensional integrals for calculating the stochastic moments of the objective function. Since the random dimension is $N = 1$, we only have to solve four occurrences, collocation points y_i, of the state equation for the numerical approximation of the objective function.

The top center of Fig. 4 shows the deterministic design aiming to support the compressive load structurally. The top center designs also show the effects of modifying the weighting parameter α balancing the first two stochastic moments of the objective function for a given Gaussian distribution of the bearing $\theta \sim N(0, (\frac{\pi}{12})^2)$. By increasing

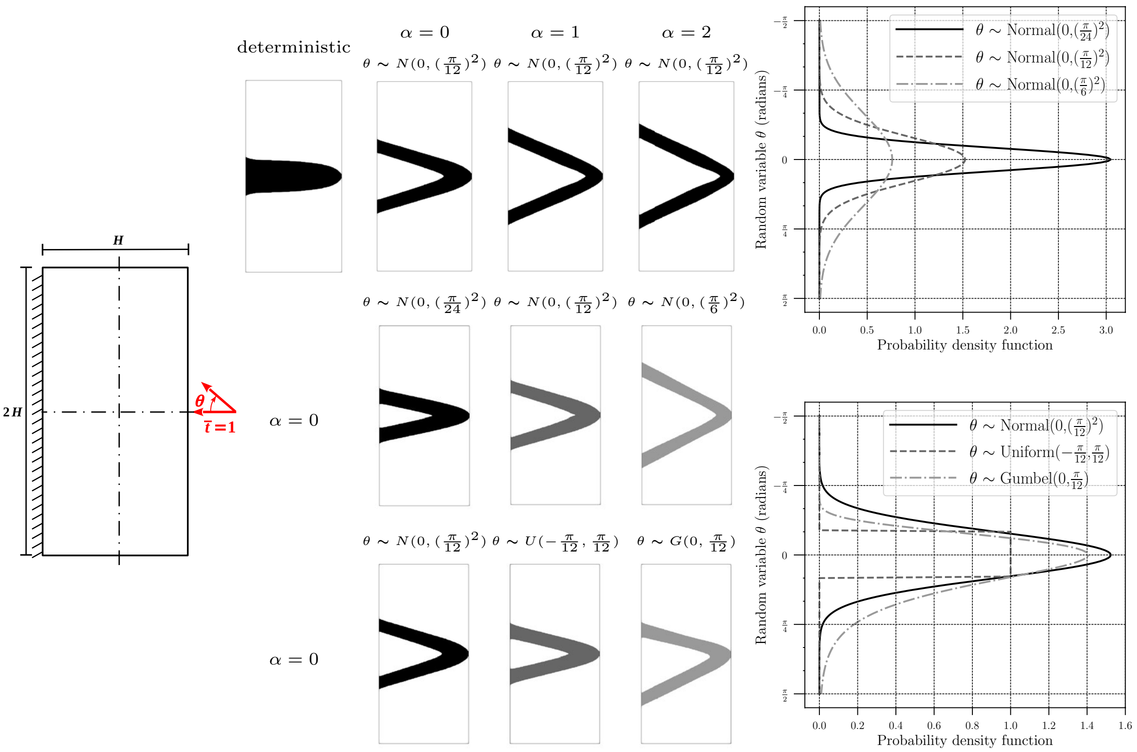


Fig. 4 Short-cantilever experiment introducing uncertainty in the bearing θ of the compressive load: (left) geometric configuration and boundary conditions, (center) deterministic and robust designs, and (right) probability density functions used for modeling the uncertainty of θ

the weight α of the standard deviation in the objective function, we usually obtain more robust structural designs at the cost of decreasing their optimality. The robust designs with $\alpha = 0$ shown in the middle center of Fig. 4 show the effect of the uncertainty in the bearing θ of the compressive load following θ a Gaussian distribution with different variance values: $(\frac{\pi}{24})^2$, $(\frac{\pi}{12})^2$, and $(\frac{\pi}{6})^2$. We can observe as increasing the uncertainty of the bearing θ increments the separation between the bars supporting the possible orientations of the unitary force. The top right of Fig. 4 shows the probability density function (PDF) of the Gaussian distribution of the bearing θ for the different variance values. Note that the color of the designs corresponds to the color of the PDF modeling the uncertainty of θ.

Finally, the bottom middle of Fig. 4 shows the effect of modeling the uncertainty of the bearing θ following non-normal stochastic distributions: Uniform and Gumbel probabilistic distributions. The right of Fig. 4 depicts the PDFs for the Gaussian, Uniform, and Gumbel distributions used in the RTO with $\alpha = 0$. The color of the robust design also corresponds to the PDF used for modeling the uncertainty of θ. We can observe that the robust designs capture the possible orientations of the solicitation. One can conclude that uncertainty modeling using different probability distributions has a relevant influence on the final designs.

5.2 Michell-type experiment

The Michell-type experiment evaluates the computational performance of the proposed parallel framework in two-dimensional RTO problems using modest computational resources. Figure 5 shows the geometric configuration of the beam using the H parameter and the boundary conditions, including a roller and fixed supports at the bottom right-hand and left-hand corners, respectively. We apply three vertical uncertain concentrated forces to the bottom surface: one load $\bar{t}_2 = 2$ at the vertical symmetry axis and two loads $\bar{t}_1 = \bar{t}_3 = 1$ at a distance $2H/5$ of both sides of the vertical symmetry axis. We characterize the bearing of these concentrated loads using three independent random variables. We assume that the heading θ_2 of $\bar{t}_2$ follows a normal distribution $\theta_2 \sim N(0, (\frac{\pi}{6})^2)$, whereas the orientations of the other loads follow the normal distributions $\theta_1 \sim N(0, (\frac{\pi}{12})^2)$ and $\theta_3 \sim N(0, (\frac{\pi}{12})^2)$. We set the target volume $V^* = 15\%$ of the initial design domain. We use a Gauss–Hermite quadrature rule with a level $\ell = 3$ to approximate the multi-dimensional integrals for calculating the stochastic moments of the objective function. Since the random dimension is $N = 3$, we have to solve 69 occurrences of the state equation using FEA for the numerical approximation of the objective function.

We tessellate the domain using 768 × 320 elements, generating a model with 245760 finite elements and 493698 unknowns. We use a radius R=$2e_{size}$ for regularizing the density field, with e_{size} the edge length of the elements. The dynamic AMR performs the coarsening up to six levels $l_{ref} = 6$ from the grid of 768 × 320 elements. We initialize the projection parameters $\eta = 0.5$ and $\beta = 1.0$ of (21) following a heuristic strategy duplicating the value of β every 15 iterations from iteration 50 until the $\beta = 8$ value. We solve the RTO problem using the proposal and obviating the AMR approach to evaluate the performance improvement using the adaptive method. The resolution obviating the AMR technique uses AMG and GMG preconditioners, which are more

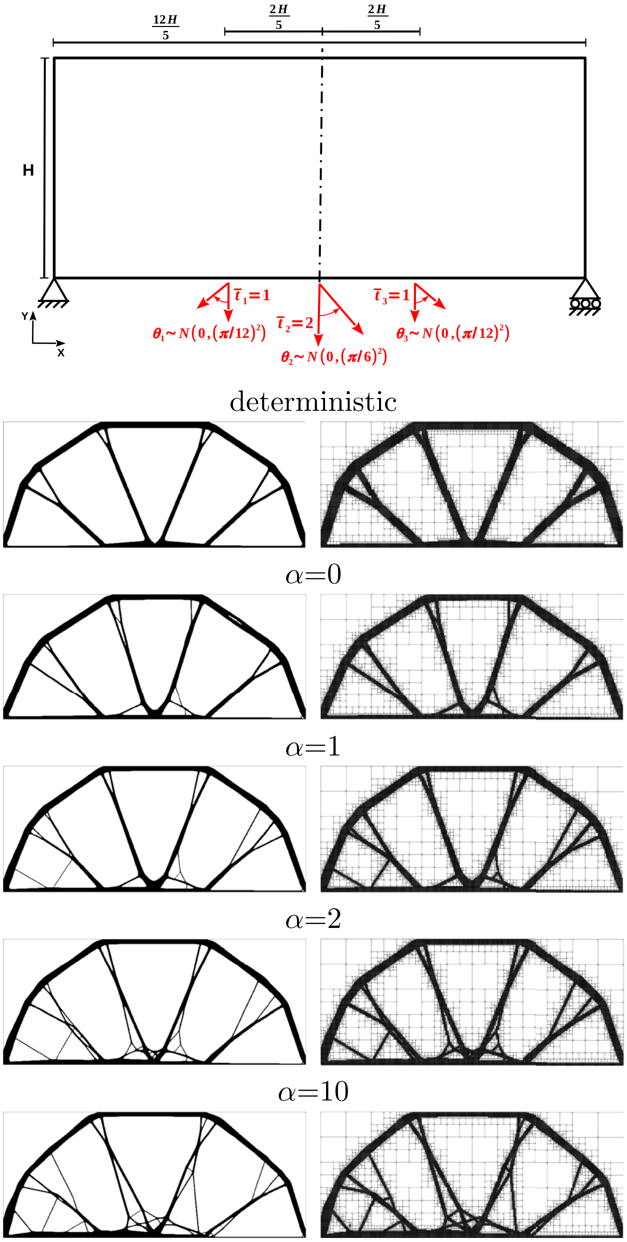


Fig. 5 Michell-type experiment with 768 × 320 elements (493698 unknowns): (top) geometric configuration, boundary conditions, and (bottom) deterministic and robust designs using AMR and different α values

suitable given the type of problem. We tune the damping factor $w_s = 0.6$ for the Jacobi smoothing to ensure convergence using GMG preconditioning. The GMG method uses six V-cycle levels from a grid of 768×320 elements to a grid of 12×5. The AMG preconditioner uses the parameters mentioned above. We use one workstation with an AMD Ryzen$^{\text{TM}}$ 9 3950X CPU for all the calculations running 16 processes in parallel. We use one computing buffer of the parallel framework for calculating the stochastic objective function and its sensitivity.

Figure 5 shows the deterministic and robust designs using different α values of the weighting parameter balancing the first two stochastic moments of the performance functional. The deterministic design provides one symmetric result, whereas robust designs are asymmetrical. We can observe how increasing the value of α increments the number of structural members in the RTO designs. We attribute this increment in the number of members to covering the variability of the orientation of the loads introducing new load paths. Figure 5 also shows the coarse mesh used by the computing buffer to calculate the objective function and the functional sensitivity.

Figure 6a shows the evolution of the expectation $\mathbb{E}[\bar{\tilde{\rho}}]$ and standard deviation $\sqrt{Var[\bar{\tilde{\rho}}]}$ of the structural compliance during the robust optimization with and without using the AMR approach. We can observe that the evolution of the first two statistics moments of structural compliance using GMG and AMG preconditioning and AMG preconditioning combined with dynamic AMR is similar. Figure 6b depicts the evolution of the objective function $\mathcal{J}_R(\bar{\tilde{\rho}})$ using AMR for different α values weighting the mean and the standard deviation of the structural compliance. We can observe that we can increase the robustness by incrementing the weight α at the cost of decreasing the optimality of the design. Figure 6c shows the number of finite elements used to calculate the stochastic objective function during the RTO process. The initial topology optimization iterations using the AMR method use the same number of finite elements as the optimization methods obviating ARM techniques, reducing the number of finite elements meaningfully after some topology optimization iterations. In particular, the number of finite elements used for calculating the stochastic moments after 45 topology optimization iterations is almost five times less using $l_{ref} = 6$ refinement levels.

The top of Fig. 7 shows the wall-clock time per iteration for solving the Michell-type RTO problem obviating and using the AMR approach. The timing includes the setup stage for the corresponding preconditioner, the dynamically coarsening and refinement up to six refinement levels $l_{ref} = 6$, and the incremental solving using the computing buffer of the 69 collocation point y_i using FEA with 16 cores on the workstation with an AMD Ryzen$^{\text{TM}}$ 9 3950X CPU. We can observe that the RTO

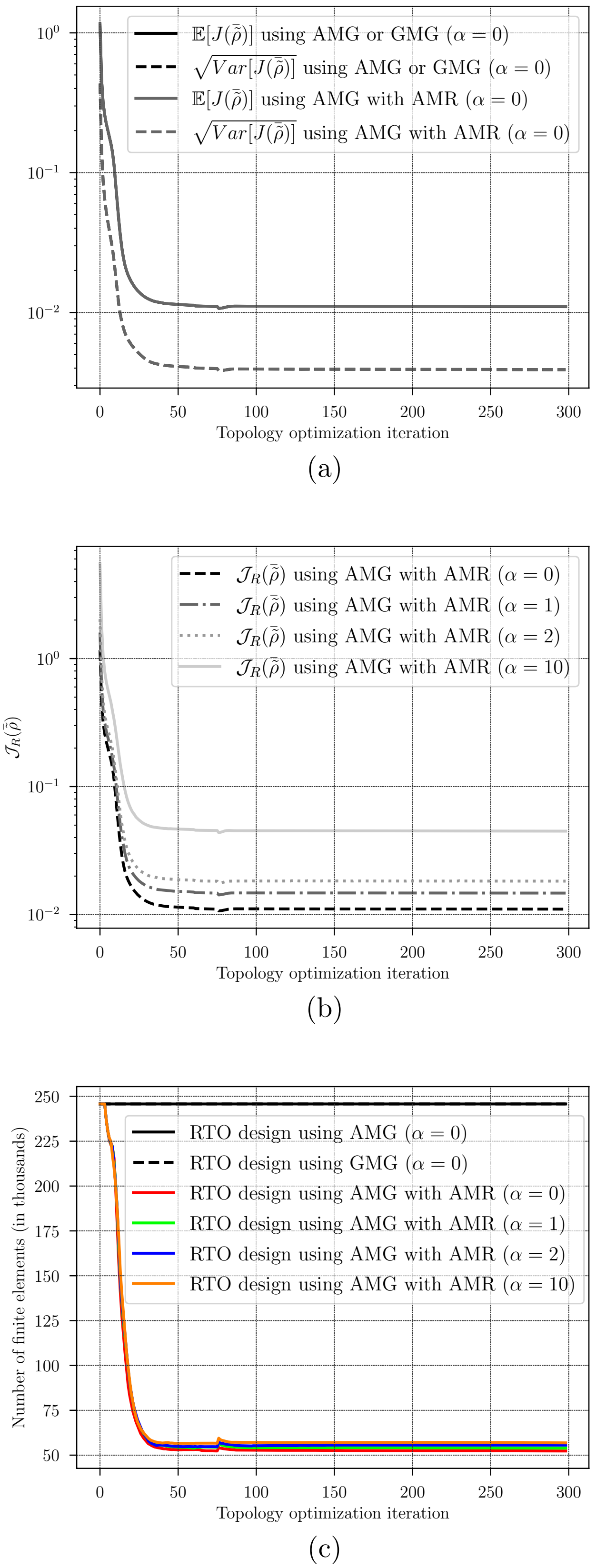


Fig. 6 Michell-type experiment: (**a**) evolution of the stochastic moments of the objective function using and obviating the AMR technique, (**b**) evolution of $\mathcal{J}_R(\bar{\tilde{\rho}})$ for different α values, and (**c**) the number of finite elements used to calculate the stochastic objective function

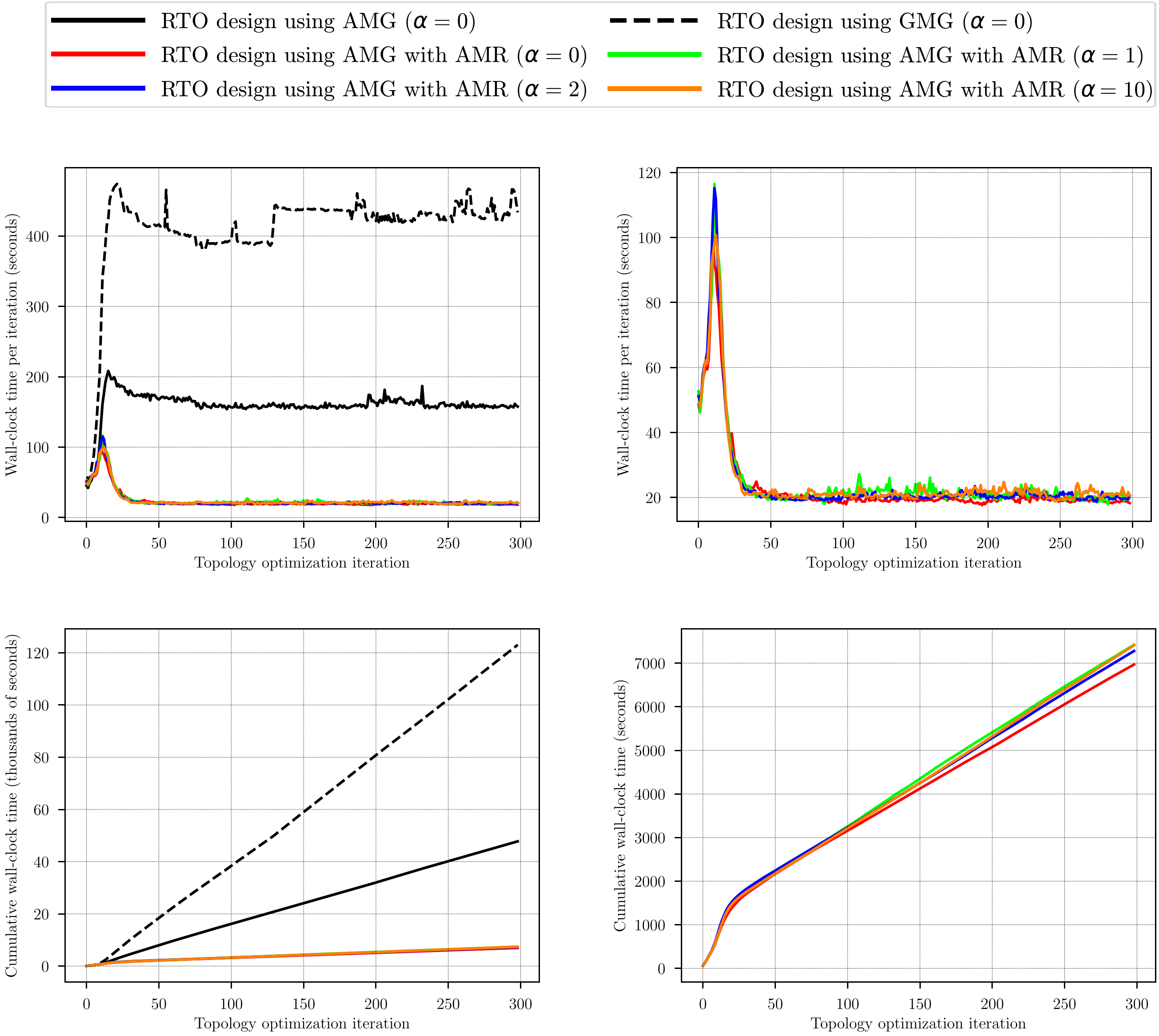


Fig. 7 Michell-type experiment: (top) wall-clock time per topology optimization iteration and (bottom) total wall-clock time for the RTO using different α values

iterations using the AMG preconditioner are significantly faster than the ones using a GMG preconditioner for two-dimensional problems. We also note a significant reduction in the wall-clock time for solving using AMR strategies. We reduce the wall-clock time for the numerical approximation of the multi-dimensional integrals using sparse grid SCM about nine and twenty two times in the last iterations of the RTO process. We also can observe that the acceleration is similar for different α values weighting the mean and the standard deviation of the structural compliance. However, the speedup of the whole process is meaningfully lower because the number of finite elements in the first iterations of RTO is similar to the approach without using AMR techniques. The bottom of Fig. 7 depicts the cumulative wall-clock time for the RTO with and without the AMR approach. We can observe a relevant speedup of 6.5x and 18x considering the implementation using the AMG and GMG preconditioner, respectively, as reference. We also note that the total wall-clock time has a similar order of magnitude for different α values.

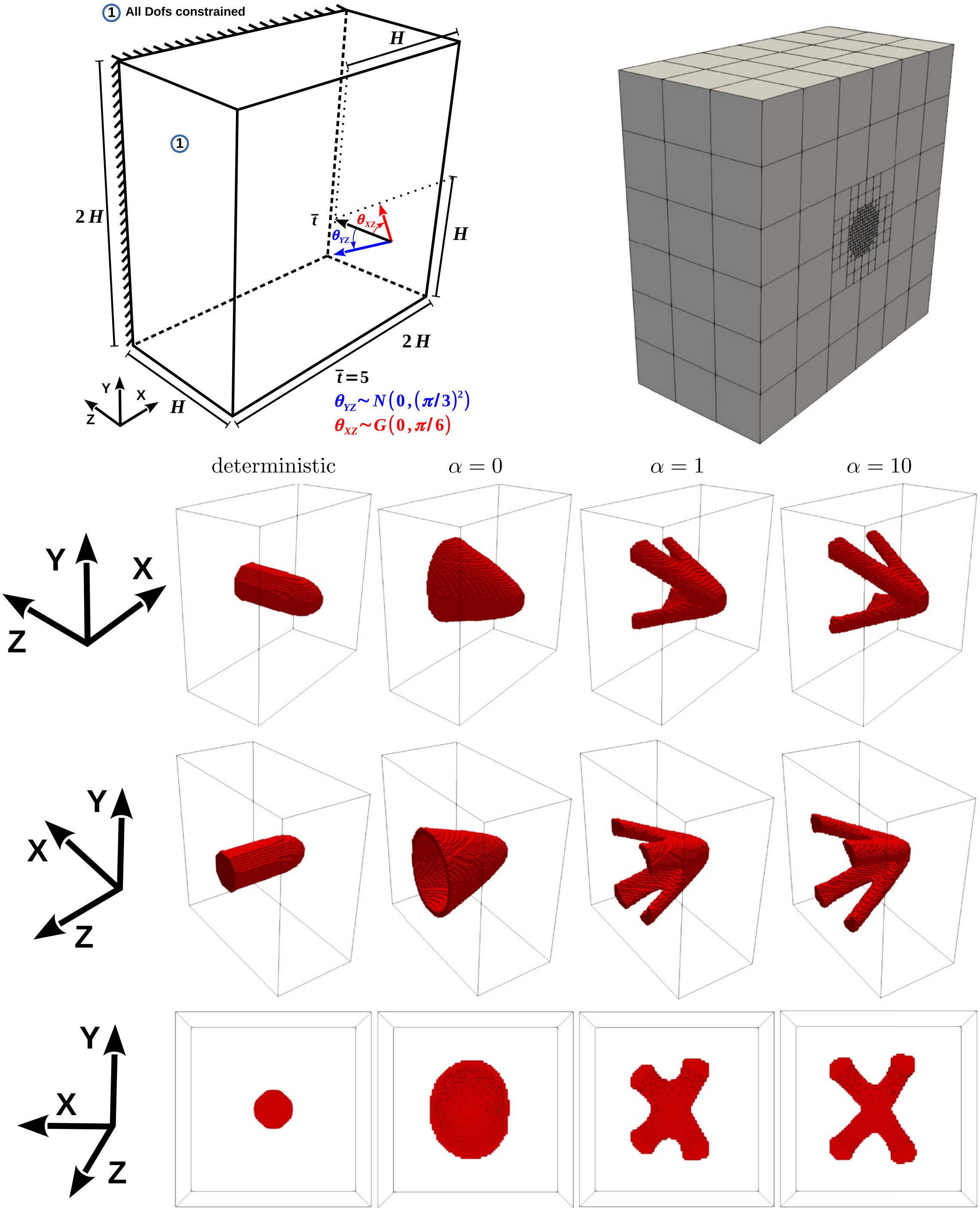
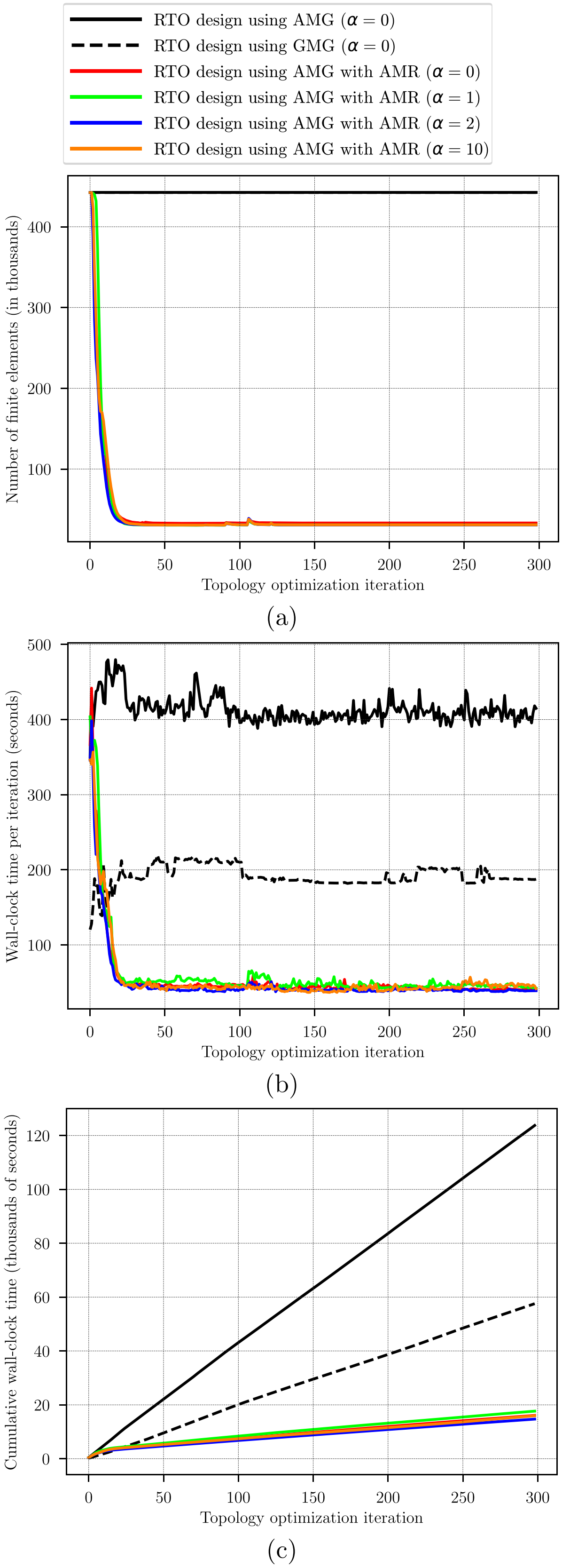


Fig. 8 3D cantilever experiment with 96 × 96 × 48 elements (1383123 unknowns): (top) geometric configuration, boundary conditions, AMR coarse mesh, and (bottom) deterministic and robust designs with different α values

5.3 3D cantilever experiment

The 3D cantilever experiment evaluates the computational performance of the framework in three-dimensional RTO problems combining different probability distributions for modeling the uncertainty. Figure 8 shows the geometric configuration of the three-dimensional short beam using the $H = 1$ parameter. It also shows the boundary conditions anchoring one face of the cantilever and applying a concentrated force $\bar{t} = 5$ at the center of the opposite face with

Fig. 9 3D cantilever experiment: (**a**) number of finite elements used to calculate the stochastic functional, (**b**) wall-clock time per topology optimization iteration, and (**c**) cumulative wall-clock time for the RTO with different α values



different uncertainty in the bearing θ_{YZ} of plane YZ and the bearing θ_{XZ} of plane XZ. In particular, the bearing θ_{YZ} follows a Gaussian distribution $\theta_{YZ} \sim N(0, (\frac{\pi}{3})^2)$, whereas the bearing θ_{XZ} follows a Gumbel distribution $\theta_{XZ} \sim G(0, \frac{\pi}{6})$. We set the target volume $V^* = 3\%$ of the initial design domain. We use a Gauss–Hermite quadrature rule with a level $\ell = 3$ to approximate the multi-dimensional integrals for calculating the stochastic moments of the objective function. Since the random dimension is $N = 2$, we have to solve 29 occurrences of the state equation using FEA for the numerical approximation of the stochastic objective function.

We tessellate the domain using $96 \times 96 \times 48$ elements, which generates a model with 442368 finite elements and 1383123 unknowns. We use a radius of $R{=}2e_{size}$ for regularizing with e_{size} the edge length of the hexahedral finite elements. We use four refinement levels $l_{ref} = 4$ from the grid of $96 \times 96 \times 48$ elements for dynamically coarsening constructing the *refinement tree*. We initialize the projection parameters $\eta = 0.5$ and $\beta = 1.0$ of (21) following a heuristic strategy duplicating the value of β every 15 iterations from iteration 60 until the $\beta = 16$ value. We solve the RTO problem using the proposal and obviating the AMR technique to evaluate the performance improvement using the adaptive method. The resolution obviating the AMR technique uses AMG and GMG preconditioners. We tune the damping factor $w_s = 0.6$ for the Jacobi smoothing to ensure convergence using GMG preconditioning. The GMG method uses a V-cycle scheme with four multigrid levels from a grid of 96 $\times 96 \times 48$ elements to one of $6 \times 6 \times 3$ elements. The AMG preconditioner uses the parameters mentioned above. We use one workstation with an AMD Ryzen$^{\text{TM}}$ 9 3950X CPU for all the calculations running 16 processes in parallel. We also use one computing buffer of the parallel framework for calculating the stochastic objective function and its sensitivity.

The upper part of Fig. 8 also shows the coarse mesh used for the incremental calculation of the objective function and the contribution to the functional sensitivity. The bottom of Fig. 8 shows the deterministic and robust designs for different α values weighting the expectation $\mathbb{E}[\bar{\tilde{\rho}}]$ and standard deviation $\sqrt{Var[\bar{\tilde{\rho}}]}$ of the structural compliance during the RTO. It shows the designs from different perspectives. We can observe the effect of uncertainty modeling in the resulting robust designs. The distribution of θ_{XZ} following a Gumbel distribution removes the symmetry of the result adapting the material distribution to the possible orientations of the load. We can observe that the increase of α value tends to concentrate the material forming beams following the probable load paths, whereas the "mean robust" design ($\alpha = 0$) generates a kind of dome.

Figure 9a shows the number of finite elements used by the computing buffer for calculating the stochastic objective function during the RTO process. We use the same number of finite elements using and obviating the AMR technique in the initial topology optimization iterations, reducing it significantly using AMR after some topology optimization iterations. In particular, the number of finite elements used for calculating the stochastic moments after 25 topology optimization iterations is more than 14 times less than the initial number of finite elements using $l_{ref} = 4$ refinement levels. The coarsening in this three-dimensional problem generates a significantly higher reduction of the number of finite elements compared to the two-dimensional problem presented above.

Figure 9b depicts the wall-clock time per iteration for solving the three-dimensional short-cantilever RTO problem by obviating and using the AMR approach. The timing includes the setup stage for the corresponding preconditioner, the dynamically coarsening and refinement up to four refinement levels $l_{ref} = 4$, and solving the 29 collocation points y_i using FEA by the computing buffer with 16 cores on the workstation with an AMD Ryzen$^{\text{TM}}$ 9 3950X CPU. Contrary to the two-dimensional RTO problem, the topology optimization iterations using the GMG preconditioner are significantly faster than iterations using an AMG preconditioner. We also can observe a significant reduction of the wall-clock time for solving using AMR strategies. We reduce the wall-clock time for the numerical approximation of the multi-dimensional integrals using sparse grid SCM about four and ten times in the last iterations of the RTO process using the implementation with GMG and AMG preconditioner, respectively, as reference. We observe that the acceleration for different α values weighting the mean and the standard deviation of the structural compliance has a similar order of magnitude.

However, the speedup of the whole process is meaningfully lower, because the number of elements in the first iterations of RTO is similar to the approach without using AMR techniques. Figure 9c depicts the cumulative wall-clock time for the RTO with and without the AMR method. We can observe a relevant speedup of 7.4x and 3.5x considering the implementation with the AMG and GMG preconditioner, respectively, as the reference.

5.4 3D double-hook experiment

The 3D double-hook experiment evaluates the computational performance and scalability of the proposal in three-dimensional RTO problems using a different number of computing buffers running in two computational nodes. In particular, we compare the performance of the parallel framework using one and two computing buffers. The upper part of Fig. 10 shows the geometric configuration of the double-hook experiment using the $H = 1$ parameter. It also shows the boundary conditions anchoring the top face of the T-leg and applying a concentrated force $\bar{t}_1 = 5$ at the center of the surface at the end of the T-shafts in the T-leg direction (z-axis). We model the uncertainty of the load introducing two force components for the robust design. One force component is in the T-shafts bearing (x-axis), and the other is in the y-axis bearing of the global reference system. We use the symmetry simplification shown in the top center of Fig. 10. We then constrain the displacements of the face of the symmetry axis to have the same structural behavior than obviating the simplification. Both force components modeling the load uncertainty follow a normal distribution with zero mean $\mu_1 = \mu_2 = 0$ and standard deviation $\sigma_1 = \sigma_2 = 0.5$. We set the target volume $V^* = 5\%$ of the initial design domain. We use a Gauss–Hermite quadrature rule with a level $\ell = 3$ to approximate the multi-dimensional integrals for calculating the stochastic moments of the objective function. Since the random dimension is $N = 2$, we have to solve 29 occurrences of the state equation for the numerical approximation of the stochastic objective function and its sensitivity.

We tessellate the domain with the symmetry simplification using two boxes of $96 \times 64 \times 64$ finite elements for the T-shaft and $32 \times 64 \times 160$ finite elements for the T-leg to generate a model of 720896 finite elements and 2252835 unknowns. We use a radius of R=2e_{size} for the regularization with e_{size} as the edge length of the hexahedral finite elements. We use four refinement levels $l_{ref} = 4$ for dynamically coarsening from the fine grid to construct the *refinement tree*. We initialize the projection parameters $\eta = 0.5$ and $\beta = 2.0$ of (21) following a heuristic strategy duplicating the value of β every 15 iterations from iteration 60 until the $\beta = 16$ value. We solve the RTO problem using the proposal and obviating the AMR approach to evaluate the performance improvement using the adaptive method in distributed-memory systems. The resolution obviating the AMR uses AMG and GMG preconditioners with one computing buffer. We tune the damping factor to $w_s = 0.6$ for the Jacobi smoothing to ensure convergence using GMG preconditioning. The GMG method adopts a V-cycle scheme with four multigrid levels. The AMG preconditioner uses the parameters mentioned above. The experiment runs using two computing nodes with two Intel Xeon E5-2687W v4 CPUs connected through a 10 Gigabit Ethernet network. The two computing nodes allow us to run 48 computing processes in parallel.

The upper part of Fig. 10 also shows the coarse mesh used by the computing buffers for the last evaluation of the stochastic objective function and its sensitivity. The bottom of Fig. 10 shows the deterministic and robust designs for different magnitudes of the α value balancing the expectation and standard deviation of the structural compliance

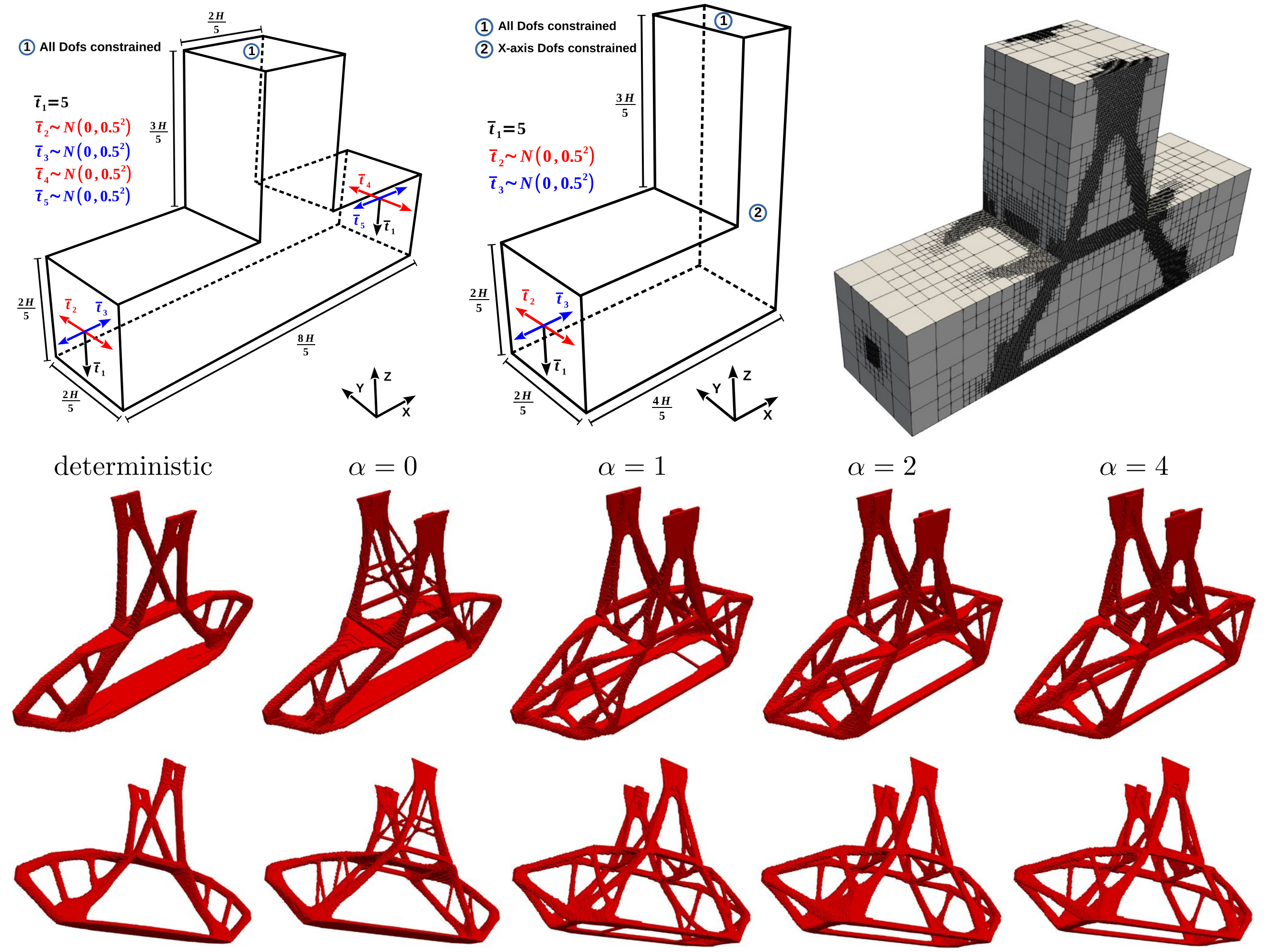
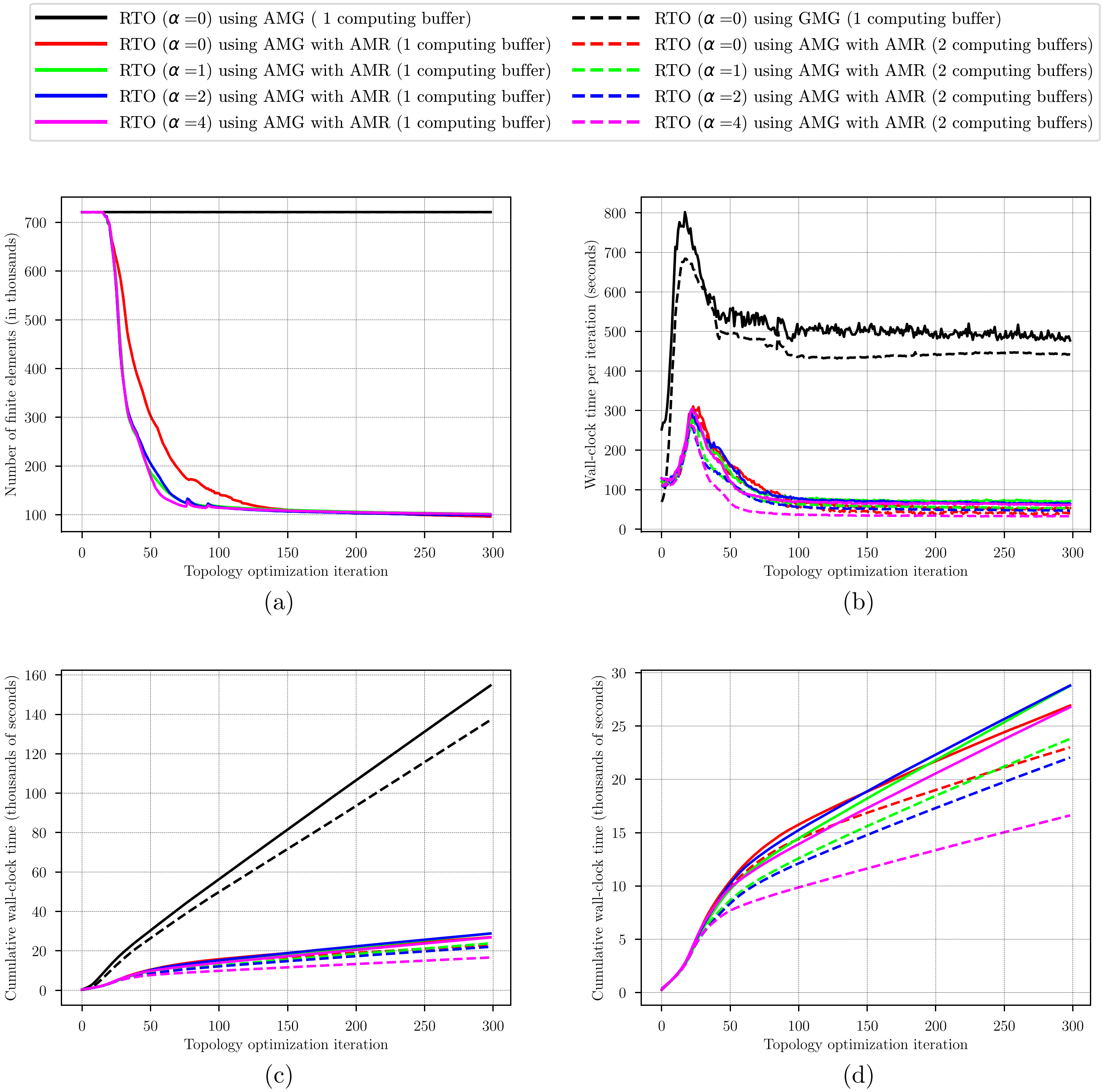


Fig. 10 Double-hook experiment with (96 × 64 × 64 + 32 × 64 × 160) elements (2252835 unknowns): (top) geometric configuration, boundary conditions, symmetry simplication, AMR coarse mesh, and (bottom) deterministic and robust designs with different α values

during the RTO. It depicts the final designs from different perspectives. We can observe significant differences between the deterministic optimal design and the robust one. The "mean robust" design with $\alpha = 0$ introduces the typical cross bracing for ensuring higher resistance than the deterministic optimal design. It also increments the number of members to diversify the load paths. We can observe an increment in the effect of the load path diversification as increasing the magnitude of the α parameter. We also noted that the material distribution of robust designs incorporating the standard deviation ($\alpha > 0$) in the stochastic functional is further from the symmetry axis in the T-shaft (x-axis) bearing. Such a material distribution increases the stiffness in the direction of the y-axis. We also noticed how the distribution of the material in the supports is different as increasing α to improve the resistance to the bending in the x-axis direction. The RTO improves the stiffness in such a bearing by introducing a kind of stiffener in the supports.

We use two computing nodes with 24 computing cores per node connected through a 10 Gigabit Ethernet network. We run the numerical experiments using the 48 computing threads available in the two computing nodes. We evaluate using one and two computer buffers to calculate the 29 deterministic problems at the collocation points y_i to estimate the stochastic objective function and its sensitivity. We group the computing threads of computing buffers by physical computing nodes in the experiments using two computing buffers. The two computer buffers self-assign them implicitly the deterministic problems that they have to solve sequentially to calculate the corresponding contributions to the stochastic objective function and its sensitivity. On the other hand, the numerical experiments using one computing buffer use all the computational resources to calculate the deterministic problems at the collocation points sequentially.

Figure 11a shows the number of finite elements in the coarse mesh used by the computing buffers during the RTO



(a)

(b)

(c)

(d)

Fig. 11 Double-hook experiment: (**a**) number of finite elements used to calculate the stochastic objective function, (**b**) wall-clock time per topology optimization iteration, and (**c**)–(**d**) cumulative wall-clock time for the RTO using two computing nodes with 48 cores

process. The initial topology optimization iterations using the AMR method use the same number of finite elements as the optimization obviating AMR techniques, reducing the number of finite elements after some topology optimization iterations. The number of finite elements is similar using a different number of computing buffers. The coarse mesh used for calculating the stochastic objective function and the contributions to the sensitivity has less than seven times fewer finite elements than the fine mesh after 60 topology optimization iterations. We observe that the number of finite elements decreases faster as the α value increases. We attribute this fact to better convergence of this problem as introducing the second stochastic moment in the objective function, reaching the optimization algorithm void regions faster.

Figure 11b depicts the wall-clock time per RTO iteration using one and two computing buffers. We can observe that the wall-clock time per RTO iteration using GMG and AMG preconditioning and obviating AMR using one computing buffer is closer than the previous experiment without network communications. We attribute such a difference to the significantly higher amount of sharing data by the GMG preconditioner. The computing processes with intensive use of communication take a relevant part of the computing time. Concerning the timing using the proposed framework, we can observe a significant reduction in the wall-clock time using one and two computing buffers. This acceleration is different for different α values weighting the mean and the standard deviation of the structural compliance. Figure 11c shows the cumulative wall-clock time for the RTO with and without the AMR technique. Figure 11d details the cumulative wall-clock time for the RTO with dynamic AMR using a different scale. We can observe a significant speedup of 7.5x and 5.0x for the experiments using the proposed framework considering the implementations with one computing buffer without AMR using the AMG and GMG preconditioners, respectively, as references.

Using computing buffers organized by computing nodes seems rewarding, especially as decreasing the number of finite elements for evaluating the collocation points y_i. We attribute this fact to the smaller size of the subdomains using one computing buffer with a higher amount of threads for the same finite-element model. We also can observe similar wall-clock times using one and two computing buffers at the beginning of the RTO. This behavior is because the method used to solve the elasticity system at the collocation points shows good scalability properties, and dividing the computation into different computing buffers reduces the benefits of reusing information from previous calculations in the sequential computing of collocation points. In any case, grouping the threads of computing buffers by physical nodes presents key design advantages concerning the minimization of inter-node communications to prevent bandwidth problems and the scalability running on large computer clusters.

6 Conclusion and future works

We combine parallel computing using distributed memory and dynamic AMR techniques to address density-based RTO problems efficiently. The proposed strategy allows us to address the computationally intensive problem of the numerical approximation of multi-dimensional integrals using sparse grid SCMs. We introduce computing buffers in the framework to adapt the evaluation of deterministic problems at the collocation points to the parallel computing resources. Such computing buffers calculate the collocation points of the SCM incrementally, requiring a memory allocation similar to the one used in the deterministic counterpart if we group the computing resources by physical nodes into computing buffers. The sequential calculation of the collocation points permits us to save heavy computations, improving the performance using modest computational resources. We can mention the coarsening and refinement only once per RTO iteration and computing buffer, which requires a relevant computational cost with intensive use of communications.

The numerical experiments show that the proposal permits us to simulate efficiently a significant number of collocation points using FEA. The Michell-type model with almost 250k finite elements requires 20700 analyses using FEA in 300 topology optimization iterations, which takes less than 2 h for the density-based RTO using the AMR strategy. The incremental parallel calculation of the collocation points of the sparse grid SCM used by the computing buffer also allows us to address the problem obviating the AMR method, requiring more than 12 h using the best preconditioning using AMG for the two-dimensional problem. The three-dimensional short-cantilever and double-hook models have 442k and 720k finite elements, respectively. We evaluate the second experiment using a different number of computing buffers, showing performance increments as exploiting the embarrassing level of parallelism of sparse grid SCMs. Both experiments require 8700 simulations, because they require solving the same number of occurrences of the state equation. Such experiments take 5 and 7 h using the AMR strategy. We can address these RTO problems obviating the AMR strategy, but the resolution can require up to a couple of days. The proposal allows us to address RTO problems with high computational requirements using modest computational resources.

The experimental results show a significant increment in computing performance for the overall computing time using AMR techniques. We also evaluate the scalability of the proposal by organizing the computing buffers in different forms, showing that grouping the computing threads of the computing buffers by physical computing nodes presents computing advantages and permits us to reduce the intensive use of inter-node communications. In this sense, we have to propose the performance evaluation of using many computing buffers per computing node as future work. We also have to mention the introduction of solution-based error estimators needed to coarse and refine solid material regions for stress-constrained RTO using AMR as future work. Finally, we have to mention as a future extension of the work evaluating the performance using some PDE filter for regularizing, which will permit us to exploit load balance in the non-conforming mesh used for solving the deterministic problems at the collocation points.

Acknowledgements This work has been supported by the AEI/FEDER and UE under Contract No. DPI2016-77538-R.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Aage N, Lazarov BS (2013) Parallel framework for topology optimization using the method of moving asymptotes. Struct Multidisc Opt 47(4):493–505. https://doi.org/10.1007/s00158-012-0869-2
2. Aage N, Andreassen E, Lazarov BS (2015) Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework. Struct Multidiscip Opt 51(3):565–572. https://doi.org/10.1007/s00158-014-1157-0
3. Amir O, Bendsøe MP, Sigmund O (2009) Approximate reanalysis in topology optimization. Int J Numer Methods Eng 78(12):1474–1491. https://doi.org/10.1002/nme.2536
4. Amir O, Stolpe M, Sigmund O (2010) Efficient use of iterative solvers in nested topology optimization. Struct Multidiscip Opt 42(1):55–72. https://doi.org/10.1007/s00158-009-0463-4
5. Amir O, Aage N, Lazarov BS (2014) On multigrid-CG for efficient topology optimization. Struct Multidiscip Opt 49(1):815–829. https://doi.org/10.1007/s00158-013-1015-5
6. Asadpoure A, Tootkaboni M, Guest JK (2011) Robust topology optimization of structures with uncertainties in stiffness - Application to truss structures. Comput Struct 89:1131–1141. https://doi.org/10.1016/j.compstruc.2010.11.004
7. Baiges J, Martínez-Frutos J, Herrero-Pérez D et al (2019) Large-scale stochastic topology optimization using adaptive mesh refinement and coarsening through a two-level parallelization scheme. Comput Methods Appl Mech Engrg 343:186–206. https://doi.org/10.1016/j.cma.2018.08.028
8. Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69(9–10):635–654. https://doi.org/10.1007/s004190050248
9. Borrvall T, Petersson J (2001) Large-scale topology optimization in 3D using parallel computing. Comput Methods Appl Mech Eng 190(46–47):6201–6229. https://doi.org/10.1016/S0045-7825(01)00216-X
10. Bourdin B (2001) Filters in topology optimization. Int J Numer Methods Eng 50(9):2143–2158. https://doi.org/10.1002/nme.116
11. Carrasco M, Ivorra B, Ramos AM (2015) Stochastic topology design optimization for continuous elastic materials. Comput Meth Appl Mech Eng 289:131–154. https://doi.org/10.1016/j.cma.2015.02.003
12. Cherkaev E, Cherkaev A (2003) Principal Compliance and Robust Optimal Design. J Elast 72(1–3):71–98. https://doi.org/10.1023/B:ELAS.0000018772.09023.6c
13. Chow E (2000) A priori sparsity patterns for parallel sparse approximate inverse preconditioners. SIAM J Sci Comput 21(5):1804–1822. https://doi.org/10.1137/S106482759833913X
14. Costa JCA, Alves MC (2003) Layout optimization with h-adaptivity of structures. Int J Numer Meth Eng 58(1):83–102. https://doi.org/10.1002/nme.759
15. De Sturler E, Paulino G, Wang S (2008) Topology optimization with adaptive mesh refinement. Int. Conf. on Computation of Shell and Spatial Structures, Spanning Nano to Mega, Ithaca, NY, USA, pp 1–4
16. Deaton JD, Grandhi RV (2014) A survey of structural and multidisciplinary continuum topology optimization: post 2000. Struct Multidiscip Opt 49(1):1–38. https://doi.org/10.1007/s00158-013-0956-z
17. Falgout RD, Jones JE, Yang UM (2006) The Design and Implementation of *hypre*, a Library of Parallel High Performance Preconditioners. In: Bruaset AM, Tveito A (eds) Numerical Solution of Partial Differential Equations on Parallel Computers, Lecture Notes in Computational Science and Engineering, vol 51. Springer, Berlin, Heidelberg, p 267–294, https://doi.org/10.1007/3-540-31619-1_8
18. Gerstner T, Griebel M (1998) Numerical integration using sparse grids. Numer Algorithms 18(3–4):209–232. https://doi.org/10.1023/A:1019129717644
19. Gournay FD, Allaire G, Jouve F (2008) Shape and topology optimization of the robust compliance via the level set method. ESAIM—Control Opt Calc 14(1):43–70. https://doi.org/10.1051/cocv:2007048
20. Guest J, Prévost J, Belytschko T (2004) Achieving minimum length scale in topology optimization using nodal design variables and projection functions. Int J Numer Meth Eng 61(2):238–254. https://doi.org/10.1002/nme.1064
21. Guest JK, Igusa T (2008) Structural optimization under uncertain loads and nodal locations. Comput Meth Appl Mech Eng 198(1):116–124. https://doi.org/10.1016/j.cma.2008.04.009
22. Gupta DK, van Keulen F, Langelaar M (2020) Design and analysis adaptivity in multiresolution topology optimization. Int J Numer Methods Eng 121(3):450–476. https://doi.org/10.1002/nme.6217
23. Henson VE, Yang UM (2002) BoomerAMG: A parallel algebraic multigrid solver and preconditioner. Appl Numer Math 41(1):155–177. https://doi.org/10.1016/S0168-9274(01)00115-5
24. Herrero-Pérez D, Martínez-Castejón P (2021) Multi-GPU acceleration of large-scale density-based topology optimization. Adv Eng Softw 157–158(103):006. https://doi.org/10.1016/j.advengsoft.2021.103006
25. Herrero-Pérez D, Martínez-Frutos J (2017) A Multi-GPU Framework for Structural Optimization under Uncertainty. In: Iványi P, Topping B, Várady G (eds) Advances in Parallel, Distributed, Grid and Cloud Computing for Engineering. Saxe-Coburg Publications, Stirlingshire, UK, chap 2, p 9–27, https://doi.org/10.4203/csets.40.2
26. Herrero-Pérez D, Picó-Vicente SG, Martínez-Barberá H (2022) Efficient distributed approach for density-based topology optimization using coarsening and h-refinement. Comput Struct 265(106):770. https://doi.org/10.1016/j.compstruc.2022.106770
27. Hypre (2021) A library of high performance preconditioners. http://www.llnl.gov/CASC/hypre/
28. Kharmanda G, Olhoff N, Mohamed A et al (2004) Reliability-based topology optimization. Struct Multidiscip Opt 26:295–307. https://doi.org/10.1007/s00158-003-0322-7
29. Kikuchi N, Chung KY, Torigaki T et al (1986) Adaptive finite element methods for shape optimization of linearly elastic structures. Comp Meth Appl Mech Eng 57(1):67–89. https://doi.org/10.1016/0045-7825(86)90071-X

30. Lazarov BS, Sigmund O (2011) Filters in topology optimization based on Helmholtz-type differential equations. Int J Numer Methods Eng 86(6):765–781. https://doi.org/10.1002/nme.3072
31. Li H, Yamada T, Jolivet P et al (2021) Full-scale 3D structural topology optimization using adaptive mesh refinement based on the level-set method. Finite Elem Anal Des 194(103):561. https://doi.org/10.1016/j.finel.2021.103561
32. Liu H, Hu Y, Zhu B et al (2018) Narrow-band topology optimization on a sparsely populated grid. ACM Trans Graph 37(6):1–14. https://doi.org/10.1145/3272127.3275012
33. Liu H, Wang Y, Zong H et al (2018) Efficient structure topology optimization by using the multiscale finite element method. Struct Multidiscip Opt 58:1411–1430. https://doi.org/10.1007/s00158-018-1972-9
34. Liu H, Tian Y, Zong H et al (2019) Fully parallel level set method for large-scale structural topology optimization. Comput Struct 221:13–27. https://doi.org/10.1016/j.compstruc.2019.05.010
35. Luo Z, Chen LP, Yang J et al (2006) Fuzzy tolerance multilevel approach for structural topology optimization. Comput Struct 84(3–4):127–140. https://doi.org/10.1016/j.compstruc.2005.10.001
36. Martínez-Frutos J, Herrero-Pérez D (2016) Large-scale robust topology optimization using multi-GPU systems. Comp Meth Appl Mech Eng 311:393–414. https://doi.org/10.1016/j.cma.2016.08.016
37. Martínez-Frutos J, Herrero-Pérez D, Kessler M et al (2016) Robust shape optimization of continuous structures via the level set method. Comput Meth Appl Mech Eng 305:271–291. https://doi.org/10.1016/j.cma.2016.03.003
38. Martínez-Frutos J, Martínez-Castejón P, Herrero-Pérez D (2017) Efficient topology optimization using GPU computing with multilevel granularity. Adv Eng Softw 106:47–62. https://doi.org/10.1016/j.advengsoft.2017.01.009
39. MFEM (2021) Modular finite element methods. http://mfem.org
40. Moens D, Vandepitte D (2005) A survey of non-probabilistic uncertainty treatment in finite element analysis. Comput Meth Appl Mech Eng 194(12–16):1527–1555. https://doi.org/10.1016/j.cma.2004.03.019
41. Nana A, Cuillière J, Francois V (2016) Towards adaptive topology optimization. Adv Eng Soft 100:290–307. https://doi.org/10.1016/j.advengsoft.2016.08.005
42. Nguyen TH, Paulino GH, Song J et al (2010) A computational paradigm for multiresolution topology optimization (MTOP). Struct Multidiscip Opt 41(4):525–539. https://doi.org/10.1007/s00158-009-0443-8
43. Nobile F, Tempone R, Webster CG (2008) A Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data. SIAM J Numer Anal 46(5):2309–2345. https://doi.org/10.1137/060663660
44. Ramírez-Gil FJ, Nelli-Silva E, Montealegre-Rubio W (2016) Topology optimization design of 3D electrothermomechanical actuators by using GPU as a co-processor. Comp Meth Appl Mech Eng 302:44–69. https://doi.org/10.1016/j.cma.2015.12.021
45. Ribeiro TP, Bernardo LFA, Andrade JMA (2021) Topology Optimisation in Structural Steel Design for Additive Manufacturing. Appl Sci 11(5):2112. https://doi.org/10.3390/app11052112
46. Ruge JW, Stüben K (1987) Algebraic Multigrid. In: McCormick SF (ed) Multigrid Methods. Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics (SIAM), chap 4, p 73–130, https://doi.org/10.1137/1.9781611971057.ch4
47. Salazar de Troya M, Tortorelli D (2018) Adaptive mesh refinement in stress-constrained topology optimization. Struct Multidiscip Opt 58:2369–2386. https://doi.org/10.1007/s00158-018-2084-2
48. Salazar de Troya M, Tortorelli D (2020) Three-dimensional adaptive mesh refinement in stress-constrained topology optimization. Struct Multidiscip Opt 62:2467–2479. https://doi.org/10.1007/s00158-020-02618-z
49. Schevenels M, Lazarov BS, Sigmund O (2011) Robust topology optimization accounting for spatially varying manufacturing errors. Comput Meth Appl Mech Eng 200(49–52):3613–3627. https://doi.org/10.1016/j.cma.2011.08.006
50. Sigmund O (2007) Morphology-based black and white filters for topology optimization. Struct Multidiscip Opt 33(4–5):401–424. https://doi.org/10.1007/s00158-006-0087-x
51. Sigmund O, Petersson J (1998) Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima. Struct Opt 16(1):68–75. https://doi.org/10.1007/BF01214002
52. da Silva GA, Cardoso EL, Beck AT (2020) Comparison of robust, reliability-based and non-probabilistic topology optimization under uncertain loads and stress constraints. Probabilistic Eng Mech 59(103):039. https://doi.org/10.1016/j.probengmech.2020.103039
53. Smolyak SA (1963) Quadrature and interpolation formulas for tensor products of certain classes of functions. Sov Math Doklady 4(5):240–243
54. Stefanou G (2009) The stochastic finite element method: Past, present and future. Comput Meth Appl Mech Eng 198(9–12):1031–1051. https://doi.org/10.1016/j.cma.2008.11.007
55. Sterck HD, Falgout R, Nolting J et al (2008) Distance-two interpolation for parallel algebraic multigrid. Numer Linear Algebra Appl 15(2–3):115–139. https://doi.org/10.1002/nla.559
56. Stüben K (2001) A review of algebraic multigrid. J Comput Appl Math 128(1–2):281–309. https://doi.org/10.1016/S0377-0427(00)00516-1
57. Svanberg K (1987) The method of moving asymptotes-a new method for structural optimization. Int J Numer Methods Eng 24(2):359–373. https://doi.org/10.1002/nme.1620240207
58. Červený J, Dobrev V, Kolev T (2019) Nonconforming Mesh Refinement for High-Order Finite Elements. SIAM J Sci Comput 41(4):C367–C392. https://doi.org/10.1137/18M1193992
59. Wang F, Lazarov BS, Sigmund O (2011) On projection methods, convergence and robust formulations in topology optimization. Struct Multidiscip Opt 43:767–784. https://doi.org/10.1007/s00158-010-0602-y
60. Wang S, De Sturler E, Paulino G (2007) Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. Int J Numer Methods Eng 69(12):2441–2468. https://doi.org/10.1002/nme.1798
61. Wang Y, Kang Z, He Q (2014) Adaptive topology optimization with independent error control for separated displacement and density fields. Comput Struct 135:50–61. https://doi.org/10.1016/j.compstruc.2014.01.008
62. Xu S, Cai Y, Cheng G (2010) Volume preserving nonlinear density filter based on heaviside functions. Struct Multidiscip Opt 41:495–505. https://doi.org/10.1007/s00158-009-0452-7
63. Yang UM (2006) Parallel Algebraic Multigrid Methods – High Performance Preconditioners. In: Bruaset AM, Tveito A (eds) Numerical Solution of Partial Differential Equations on Parallel Computers, Lecture Notes in Computational Science and Engineering, vol 51. Springer, Berlin, Heidelberg, p 209–236, https://doi.org/10.1007/3-540-31619-1_6
64. Zhao Q, Chen X, Ma ZD et al (2015) Robust Topology Optimization Based on Stochastic Collocation Methods under Loading Uncertainties. Math Probl Eng 580980. https://doi.org/10.1155/2015/580980

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.