Large-scale stochastic topology optimization using adaptive mesh refinement and coarsening through a two-level parallelization scheme

Joan Baiges[1], Jesús Martínez-Frutos[2*], David Herrero-Pérez[2], Fermin Otero[3,4], Alex Ferrer[1,4]

[1] *Universitat Politècnica de Catalunya, Jordi Girona 1-3, Edifici C1, 08034 Barcelona, Spain*
[2] *Computational Mechanics and Scientific Computing Group, Technical University of Cartagena, Campus Muralla del Mar, 30202 Cartagena, Murcia, Spain*
[3] *INEGI, Rua Dr. Roberto Frias 400, 4200-465 Porto, Portugal*
[4] *Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Edifici C1, Campus Nord UPC C/ Gran Capità S/N 08034 Barcelona, Spain*

Abstract

Topology Optimization Under Uncertainty (TOUU) of large-scale continuum structures is a computational challenge due to the combination of large finite element models and uncertainty propagation methods. The former aims to address the ever-increasing complexity of more and more realistic models, whereas the latter is required to estimate the statistical metrics of the TOUU formulation. In this work, the computational burden of the problem is addressed using a sparse grid stochastic collocation method, to calculate the statistical metrics of the TOUU formulation, and a parallel Adaptive Mesh Refinement (AMR) method, to efficiently solve each of the stochastic collocation nodes. A two-level parallel processing scheme (TOUU-PS2) is proposed to profit from parallel computation on distributed memory systems: the stochastic nodes are distributed through the distributed memory system, and the efficient computation of each stochastic node is performed partitioning the problem using a domain decomposition strategy and solving each subdomain using an AMR method. A dynamic load-balancing strategy is used to balance the workload between subdomains, and thus increasing the parallel performance by reducing processor idle time. The topology optimization problem is addressed using the topological derivative concept in combination with a level-set method. The performance and scalability of the proposed methodology are evaluated using several numerical benchmarks and real-world applications, showing good performance and scalability up to thousands of processors.

Keywords:
Adaptive mesh refinement; Large scale; Parallel computing; Robust topology optimization; Topological derivative; Sparse grid;

*Corresponding author
Email address: `jesus.martinez@upct.es` (Jesús Martínez-Frutos[2])

1. Introduction

Topology optimization aims to find the optimal layout of material within a design domain for a given set of boundary conditions such that the resulting material distribution meets a set of performance targets [1]. Contrary to other disciplines within structural optimization such as size and shape optimization, in topology optimization the material distribution is obtained without assuming any prior structural configuration. This provides a powerful tool to find the best conceptual design that fulfills the requirements at the early stages of the structural design process [2]. Topology optimization methods have been successfully applied to improve the design of complex industrial problems assuming deterministic conditions [3]. However, it obviating the uncertainty in the design process may influence notably the optimal design performance under real-world engineering conditions. The different sources of uncertainty may affect not only the safety and reliability of structural designs but also their performance. Such sources of uncertainty include epistemic uncertainties, typically due to limited data and knowledge, and aleatory uncertainties, which are the natural randomness in a process, including manufacturing imperfections, unknown loading conditions, variations of the material properties, etc. The introduction of uncertainty to model realistic conditions in the design process has shown to be a key issue for solving real-world engineering problems in several fields, such as civil [4], automotive [5], aerospace [6], and mechanical [7] engineering, to name but a few.

The topology optimization methods incorporating uncertainty in their formulation are embraced under the term of Topology Optimization Under Uncertainty (TOUU) methods. The formulation of such methods differs from each other in the design objective as well as in the way the uncertainty is incorporated into the optimization formulation [8] to deal with the wide concept of "structural robustness". By adopting a probabilistic approach, two formulations are commonly used in the literature: Robust Topology Optimization (RTO) and Reliability-Based Topology Optimization (RBTO). RTO incorporates the first two statistical moments of the cost function to obtain optimal designs which are less sensitive to variations in the input data [9, 10]. On the other hand, RBTO aims at minimizing a deterministic prescribed criterion while explicitly considering the effects of uncertainty in terms of the probability of constraint violation (probability of failure) [11]. In the same vein, Risk-Averse Topology Optimization (RATO) aims at minimizing a risk function that quantifies the expected loss related to the damages [12]. In this work, we focus on the RTO problem considering uncertainty in the loading conditions [13, 14].

One of the main challenges in TOUU is the computational burden addressing real-world engineering problems, which is especially exacerbated for large-scale finite element models. This problem still remains even though significant numerical and theoretical advances have been achieved in the last years. In fact, the solving of few hundred analyses of expensive finite element models requires the use parallel processing in large-scale deterministic topology optimization problems [15]. Such large-scale models are needed to obtain the details of many problems, such as structural (e.g. wing aircraft structures [16]) and heat transfer (e.g. heat sinks cooled by natural convection [17]) ones. Multi-core [18, 19] and many-core [20, 21] architectures, or the combination of both [22], are commonly needed to address such large-scale deterministic problems. However, the number of simulations required by the deterministic design is far from the required number of samples for estimating the probabilities required by the formulation of TOUU methods. The TOUU problem, in general, and the RTO problem addressed in this work, in particular, can become unaffordable due to the combination of two computationally demanding processes, namely large-scale topology optimization and uncertainty propagation. The former involves diverse demanding tasks, such as the solving and assembly of large systems of equations, which may increase meaningfully the memory consumption and the processing time when dealing with large finite element models. The latter requires a large number of finite element model runs [23] that can increase the processing time significantly. This issue has led to many developments using high performance computing techniques to address uncertainty quantification problems [24, 25]. Besides, the uncertainty propagation problem suffers from the so-called *curse of dimensionality* [26, 27]: the computational cost grows exponentially with the number of random variables defining the underlying stochastic domain.

Several works have made use of adaptive coarsening and refinement of the mesh, namely Adaptive Mesh Refinement (AMR), to reduce the computational cost of topology optimization problems.

This technique is especially useful when relatively small volume fractions are required since after a few iterations the domain of computation is largely void [28]. The computation in void regions contributes significantly to the overall computational cost but little to the accuracy of computation and design. The underline idea of AMR is to save computational cost by reducing the total number of elements and having fine elements only where and when necessary with the aim of obtaining an equivalent design to the one that would be obtained on a uniformly fine mesh, but at a much cheaper computational cost. Besides, AMR offers the possibility to increase the resolution of the material distribution and to reduce the error in the estimated behavior providing more accurate results in the region of interest [29]. The early work of Kikuchi et al. [30] introduced adaptive grid design combining numerical grid-generation methods and adaptive finite element methods, including r-, h-, and p-methods, to address the shape optimal design problem. Ramm et al. [31] proposed an adaptive density-based topology optimization method for optimizing structures with an elastoplastic material. The Solid Isotropic Material Penalization (SIMP) method with AMR techniques is also proposed in recent works [29, 32]. Adaptive mesh refinement schemes have also been used with genetic algorithms to address the topology optimization problem [33]. Ramm et al. [31] proposed an adaptive density-based topology optimization method for optimizing structures with an elastoplastic material. The Solid Isotropic Material Penalization (SIMP) method with AMR techniques is also proposed in recent works [29, 32]. Adaptive mesh refinement schemes have also been used with genetic algorithms to address the topology optimization problem [33].

This work aims to overcome the prohibitive computational cost of large-scale TOUU problems in general and the RTO problem in particular by using an adaptive two-level parallelization scheme (TOUU-PS2). This is done by selecting and developing suitable techniques for estimating the system response under uncertainties in modern computing infrastructures. The response of the stochastic system is calculated on distributed memory systems using sparse grid stochastic collocation methods. These methods are embarrassingly parallel requiring the solving of the model in the collocation points. The stochastic collocation points are split into smaller problems using domain decomposition methods, which are solved using AMR and coarsening to focus the computational effort on the spatial regions of interest; in particular, the solid-void interface. A key point to increase the parallel performance is the use of a dynamical parallel repartitioning strategy during the optimization to balance the workload between computational processes [34]. The solid-void interface is located using the topological sensitivity [35, 36], and the h-method for AMR is used for improving the objective function minimization and reducing the computational burden. Another key point of the proposed methodology is the communication between subdomains and stochastic nodes. The adjacent subdomains communicate between them to calculate the system response at the stochastic nodes, and then the solution at each subdomain is used to compute the statistical moments required by RTO formulation. The proposed parallelization scheme permits to minimize the communication at the two levels of the computation: the adjacent subdomains communicate between them in the first level, where the system response is calculated for each stochastic node, and only the response the corresponding subdomain of all the stochastic points is required to calculate the statistical moments in the second level. The numerical experiments evaluate the strong and weak scalability of the proposed parallelization scheme.

The paper is organized as follows. The basis and theoretical background of TOUU problems and the RTO formulation are briefly reviewed in section 2. Section 3 presents the numerical resolution of the RTO problem using the topological derivative. The parallel implementation of the proposal is described in section 4. Section 5 is devoted to the numerical experiments used for validating the proposed method. Finally, section 6 presents the conclusion of the proposed method for addressing RTO problems efficiently using modern computing infrastructures.

2. Topology optimization of structures under uncertainty (TOUU)

2.1. Setting of the problem

Let $(\Omega, \mathcal{F}, P)$ be a complete probability space, and let $D \subset \mathbb{R}^d$ ($d = 2$ or $d = 3$) be a bounded Lipschitz domain whose boundary is decomposed into three disjoint parts $\partial D = \Gamma_D \cup \Gamma_N \cup \Gamma_0$. Consider the linearized elasticity system under random input data

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

where u is the stochastic displacement field, σ is the Cauchy stress tensor, b 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. The stress tensor σ and the symmetric gradient of the displacement field ε are related by means of the following constitutive equation:

$$\sigma(x,\omega) = \mathbb{C}(x,\omega) : \varepsilon(x,\omega), \tag{2}$$

where $\mathbb{C}$ represents the fourth order constitutive tensor. Notice that compared to its deterministic counterpart, the body and surface forces and the constitutive tensor depend on a spatial variable x and on a random event $\omega \in \Omega$.

In the topology optimization problem, the structural boundary splits the domain D into two different subdomains D^+, corresponding to a stiff material, and D^-, corresponding to a soft material, such that $D^+ \cup D^- = D$ and $D^+ \cap D^- = \emptyset$. This subdivision into subdomains can be represented by means of a characteristic function χ such that

$$\chi(x) = \begin{cases} 1 & x \in D^+ \\ 0 & x \in D^- \end{cases}. \tag{3}$$

This allows us to rewrite the constitutive tensor equation defined in all the domain as:

$$\mathbb{C}(x,\omega) = \chi \; \mathbb{C}^+(x,\omega) + (1-\chi) \; \mathbb{C}^-(x,\omega), \tag{4}$$

where $\mathbb{C}^+$ and $\mathbb{C}^-$ are the fouth order constitutive tensors of the stiff and the soft material respectively.

Let us now define the following Hilbert spaces, required for appropriately dealing with problem (1):

$$V = \left\{ v \in H^1(D)^d : v|_{\Gamma_D} = 0 \text{ in the sense of traces} \right\},$$

equipped with the $H^1(D)^d$-norm, and:

$$L^2_P(\Omega; V) = \left\{ f : \Omega \to V, \; f \text{ is } \mathcal{F}\text{-measurable and } \|f\|^2_{L^2_P(\Omega;V)} = \textstyle\int_\Omega \|f(\omega)\|^2_V dP(\omega) < \infty \right\}.$$

The following assumptions on the uncertain input parameters of the system (1) are made:

(A1) Lamé parameters: $\lambda(x,\omega), \mu(x,\omega) \in L^\infty_P(\Omega; L^\infty(\mathrm{D}))$ and there exist positive constants μ_{min}, μ_{max}, λ_{min}, λ_{max} such that

$$0 < \mu_{min} \le \mu(x,\omega) \le \mu_{max} < \infty \quad \text{a.e. } x \in D, \quad \text{a.s. } \omega \in \Omega,$$

and

$$0 < \lambda_{min} \le 2\mu(x,\omega) + d\lambda(x,\omega) \le \lambda_{max} < \infty \quad \text{a.e. } x \in D, \quad \text{a.s. } \omega \in \Omega,$$

(A2) $b = b(x,\omega) \in L^2_P\left(\Omega; L^2(D)^d\right)$,

(A3) $\bar{t} = \bar{t}(x,\omega) \in L^2_P\left(\Omega; L^2(\Gamma_N)^d\right)$,

(A4) Finite dimensional noise: the random input data of (1), i.e. Lamé coefficients and loads, depends on a finite number N of real-valued random variables $\{Y_n\}_{n=1}^N$.

The TOUU problem is then formulated as the minimization of the structural compliance under random input data subjected to the maximum material allowed, as follows:

$$\begin{aligned} \min_{\chi \in D_L} \quad & J(\chi,\omega) = \int_D b(x,\omega) u \; dx + \int_{\Gamma_N} \bar{t}(x,\omega) u \; ds \\ \text{s. t. :} \quad & a(\chi,u,v,\omega) = l(v,\omega) \quad \forall\, v \in V, \forall\, \omega \in \Omega \\ & D_L = \left\{\chi \in L^\infty(D,\{0,1\}), \int_D \chi dx = L|D|\right\} \end{aligned} \tag{5}$$

where D_L is the feasible domain restricted to a volume constraint denoted as a fraction $0 < L < 1$ of the domain D, and $a(\chi,\cdot,\cdot,\omega)$ and $l(\cdot,\omega)$ are the bilinear and linear forms which are given by:

$$a(\chi,u,v,\omega) = \int_D \sigma(u(x,\omega)) : \varepsilon(v(x,\omega)) \; dx \tag{6}$$

$$l(v,\omega) = \int_D b(x,\omega) v \; dx + \int_{\Gamma_N} \bar{t}(x,\omega) v \; ds \; . \tag{7}$$

The general formulation of the TOUU problem (5) provides a different solution for each realization of the random event. The random functional should be transformed into a deterministic one to address the optimization problem using conventional optimization algorithms. Different formulations have been proposed in the literature depending on how the uncertainty is incorporated in the formulation. In this work, the formulation is focused on the RTO problem, which aims to find designs less sensitive to variations in the design variables and the input parameters. This is normally done by formulating the RTO problem as a two-objective optimization problem where the expected value and standard deviation of the compliance are considered as a measure of structural robustness. A weighted approach can be used to scalarize the multi-objective problem into a single-objective one. The RTO problem can be formulated as follows

$$\begin{aligned} \min_{\chi \in D_L} \quad & \mathcal{J}_R(\chi) = \mathbb{E}[J(\chi,\omega)] + \alpha\sqrt{Var[J(\chi,\omega)]} \\ \text{s. t. :} \quad & \int_\Omega a(\chi,u,v,\omega) dP(\omega) = \int_\Omega l(v,\omega) dP(\omega) \\ & \qquad\qquad \forall\, v \in L^2_P(\Omega,V), \\ & D_L = \left\{\chi \in L^\infty(D,\{0,1\}), \int_D \chi \; dx = L|D|\right\}, \end{aligned} \tag{8}$$

where $\mathbb{E}[\cdot]$ denotes the expectation operator, $Var[\cdot]$ denotes the variance operator, and $\alpha \geq 0$ is a weighting parameter which balances the mean and the variance of the performance function. In this probabilistic setting, the mean and the variance of the performance function are obtained as follows

$$\begin{aligned} \mathbb{E}[J(\chi,\omega)] &= \int_\Omega J(\chi,\omega) dP(\omega), \\ Var[J(\chi,\omega)] &= \int_\Omega J^2(\chi,\omega) dP(\omega) \\ &\quad - \left(\int_\Omega J(\chi,\omega) dP(\omega)\right)^2. \end{aligned} \tag{9}$$

2.2. RTO using the topological derivative concept

In order to solve problem (8), several approaches exist. In this work we favor the use of the topological derivative concept [37] together with a level-set approach in order to advance to the optimal topology.

The topological derivative studies the sensitivity of a given functional with respect to the apparition of an infinitesimal inclusion of a different material in a given point of the domain of interest. In the problem studied in this work, the objective is to obtain the sensitivity of the robust functional $J_R(\chi)$ with respect to the inclusion of soft material in the stiff D^+ subdomain, or with respect to the inclusion of stiff material in the soft D^- subdomain.

Using the topological-shape sensitivity analysis proposed in [35], a formal computation of the topological derivative of the cost functional $D_T\mathcal{J}_R(\chi)$ leads to:

$$D_T\mathcal{J}_R(x,\chi) = \int_\Omega \Big(1 + \frac{\alpha J(\chi,\omega)}{\sqrt{Var[J(\chi,\omega)]}} - \frac{\alpha \mathbb{E}[J(\chi,\omega)]}{\sqrt{Var[J(\chi,\omega)]}}\Big)\sigma(x,\omega) : \mathbb{P}(x,\omega) : \varepsilon(x,\omega) dP(\omega)$$

where $\mathbb{P}$ stands for the fourth order *Pólya-Szegö* polarization tensor, whose expression can be found in [38]. In the isotropic case, it basically depends on the Young's modulus and Poisson ratio in $D^+(E^+,\nu^+)$ and $D^-(E^-,\nu^-)$. In this work, the isotropic 2D polarization tensor has been used as an approximation for the appropriate 3D polarization tensor.

The properties of the polarization tensor ensure that:

$$\begin{aligned} \sigma(x,\omega) : \mathbb{P}(x,\omega) : \varepsilon(x,\omega) \geq 0 \quad \forall x \in D^+ \\ \sigma(x,\omega) : \mathbb{P}(x,\omega) : \varepsilon(x,\omega) \leq 0 \quad \forall x \in D^-. \end{aligned}$$

We can now define a signed topological derivative such that:

$$\bar{D}_T\mathcal{J}_R(x,\chi) = \begin{cases} +D_T\mathcal{J}_R(x,\chi) & \text{in} \quad x \in D^+ \\ -D_T\mathcal{J}_R(x,\chi) & \text{in} \quad x \in D^- \end{cases},$$

Let us now introduce the following interpretation of the signed topological derivative, which will be used in the subsequent sections of this work. For a given topology, computing the topological derivative allows one to know, for each given spatial point, how would the robust cost functional change if the material was switched. As a consequence, once the optimal value for the characteristic function χ, solution to (8), has been reached, the following condition holds:

$$\bar{D}_T\mathcal{J}_R(x,\chi) \geq \bar{D}_T\mathcal{J}_R(y,\chi), \forall x \in D^+, \forall y \in D^-. \tag{10}$$

This allows one to construct a level set function ψ, which will implicitly characterize D^+ and D^-. This level set function is defined as:

$$\psi(x,\chi) = \bar{D}_T\mathcal{J}_R(x,\chi) + \lambda,$$

where $\lambda \in \mathbb{R}$ is a scalar responsible for ensuring that the volume restriction in (8) is fulfilled and which can be computed by enforcing:

$$\int_D H(\psi(x,\chi)) = L|D|,$$

where H is the Heaviside step function:

$$H(\psi) = \begin{cases} 1 & \text{if } \psi \geq 0 \\ 0 & \text{if } \psi < 0 \end{cases}.$$

From (10), it can be observed that for the solution of (8) it holds:

$$\chi = H(\psi).$$

3. Numerical resolution of RTO problem using the topological derivative

In this section we describe the methodology used for the solution of the topology optimization problem using a finite element method, which will be coupled with an adaptive mesh refinement technique in the following sections.

3.1. Stochastic collocation Finite Element Method

Following the infite dimensional noise assumption (A4), the uncertainty in the input data of the PDE is represented by a finite set of random variables $Y = (Y_1, \cdots, Y_N)$ which are mapped from the sample space Ω to $\mathbb{R}$ following the joint probability density function $\rho = \rho(y)$. The variational formulation of (1) can be rewritten as:

$$a(\chi, u, v, y) = l(v, y) \quad \forall v \in V, \quad y \text{ in } \Lambda. \tag{11}$$

where $\Lambda = Y(\Omega) \subset \mathbb{R}^N$. Thus, the stochastic boundary-value problem (1) has been transformed into a deterministic PDE system with an N-dimensional parameter. For the solution of (11) we use a finite element method, with the particularity that an accurate integration scheme capable of exactly integrating the variational form in elements cut by the material interface is used. The variational form of the finite element problem is: find $u_h \in V_h$ such that:

$$a(\chi, u_h, v_h, y) = l(v_h, y) \quad \forall v_h \in V_h, \quad y \text{ in } \Lambda. \tag{12}$$

where V_h denotes a finite element space approximating the Sobolev space V. Since the domain is described by a level-set function and in order to exactly integrate (12), the quadrature points at the elements cut by the material interface need to be integrated carefully, because the constitutive tensor $\mathbb{C}$ has a discontinuity in these elements. For this, the quadrature rule in the cut elements is modified, selecting new quadrature points in such a way that integrals are computed exactly. In this sense the method proposed in this work can be understood as a Cut-Mesh method.

3.2. Numerical approximation of the integrals in the random domain using sparse grids

The stochastic collocation method leads to a set of uncoupled deterministic sub-problems collocated at some stochastic nodes $y^k \in \Lambda$. As a consequence, the statistical quantities of interest can be obtained by solving the multidimensional integrals:

$$\begin{aligned} \mathbb{E}[J_h(\chi, y)] &= \int_\Lambda J_h(\chi, y)\rho(y)\,dy, & (13)\\ Var[J_h(\chi, y)] &= \int_\Lambda J_h^2(\chi, y)\rho(y)\,dy - \left(\int_\Lambda J_h(\chi, y)\rho(y)\,dy\right)^2. & (14) \end{aligned}$$

In practice, these multi-dimensional integrals cannot be evaluated analytically, and their numerical computation becomes computationally intractable as the dimension of the random domain increases. This issue has motivated the development of efficient methods to address this issue, such as dimension reduction methods and sparse grid collocation methods. In this work, due to the smoothness of the solution of the elasticity system, the numerical approximation in the random domain is performed using anisotropic sparse grid collocation method described in [39, 40]. For an integer $\ell \in \mathbb{N}_+$, called the level, and a vector of weights for the different stochastic directions $g = (g_1, \cdots, g_N)$, consider the multi-index set:

$$\mathbb{X}_g(\ell, N) = \left\{ i = (i_1, \cdots, i_N) \in \mathbb{N}_+^N, \quad i \geq 1 : \quad \sum_{n=1}^{N} (i_n - 1)\, g_n \leq \ell \underline{g} \right\}, \tag{15}$$

where $\underline{g} = \min_{1 \leq n \leq N} g_n$. Similarly to the isotropic Smolyak quadrature rule [41], the anisotropic Smolyak quadrature formula applied to $J_h(\chi, y)$ is given by:

$$\begin{aligned} \mathcal{A}_g(\ell, N)\, J_h(\chi, y) &:= \sum_{i \in \mathbb{X}_g(\ell, N)} \left(\Delta^{i_1} \otimes \cdots \otimes \Delta^{i_N}\right) J_h(\chi, y) \\ &= \textstyle\sum_{r_1=1}^{R_{i_1}} \cdots \sum_{r_N=1}^{R_{i_N}} J_h\left(x; y_1^{r_1}, \cdots, y_N^{r_N}\right) w_{i_1}^{r_1} \cdots w_{i_N}^{r_N} \end{aligned} \tag{16}$$

where $\Delta^{i_n} = \mathcal{Q}^{i_n} - \mathcal{Q}^{i_n-1}$, with $\mathcal{Q}^0 = 0$, is a quadrature rule in which the coordinates $y_n^{r_n}$ of the nodes are those for the 1D quadrature formula $\mathcal{Q}^{i_n}$ and its associated weights $w_{i_n}^{r_n}$ are the difference between those for the i_n and $i_n - 1$ levels. The number of collocation points in the nth direction is denoted by R_{i_n}. By using (16) the statistical moments of $J_h(\chi, y)$ can be approximated as:

$$\mathbb{E}[J_h(\chi, y)] \approx \mathcal{A}_g(\ell, N)\, J_h(\chi, y) \tag{17}$$

$$Var[J_h(\chi, y)] \approx \mathcal{A}_g(\ell, N)\, J_h^2(\chi, y) - \mathcal{A}_g(\ell, N)\, J_h(\chi, y)^2 \tag{18}$$

3.3. *Description of the iterative topology optimization algorithm*

The last required ingredient is an algorithm to arrive to the solution of problem (8). For this, an iterative process will be defined: initially, a level set function $\psi_h \in V_h$ is defined with unit initial value:

$$\psi_h^0(x) = 1 \quad \text{in } D,$$

where the superscript indicates iteration number. From this level set value, a characteristic function can be built:

$$\chi^i(x) = H(\psi_h^{i-1}(x)),$$

which allows one to solve the solid mechanics problem (12) and compute the topological derivative $\bar{D}_T \mathcal{J}_R^i(x, \chi^i)$. For convergence aspects, the algorithm also requires of an intermediate function $\phi_h^i(x, \chi^i)$. This function is initially defined as the projection onto the finite element space of the normalized topological derivative $\bar{D}_T \mathcal{J}_R^i(x, \chi^i)$ in order to bound the level-set function with a relaxation scheme introduced as the iterative process advances, i.e.:

$$\phi_h^i(x, \chi^i) = \kappa^i \frac{P_h\left(\bar{D}_T \mathcal{J}_R^i(x, \chi^i)\right)}{\left\|P_h\left(\bar{D}_T \mathcal{J}_R^i(x, \chi^i)\right)\right\|} + (1 - \kappa^i) \frac{P_h\left(\bar{D}_T \mathcal{J}_R^{i-1}(x, \chi^{i-1})\right)}{\left\|P_h\left(\bar{D}_T \mathcal{J}_R^{i-1}(x, \chi^{i-1})\right)\right\|}, \tag{19}$$

where κ is a relaxation parameter, and P_h indicates a projection onto the finite element space. In the numerical examples, P_h is computed by using a lumped mass matrix approach for computational efficiency. This approach plays the role of standard filtering in topology optimization. Finally, the level set function at the current iteration is defined as:

$$\psi_h^i(x) = \phi_h^i(x, \chi^i) + \lambda^i,$$

where λ^i is computed by using the secant method to solve the following equation:

$$\int_D H(\psi_h^i(x)) = L|D|.$$

There exist several possible strategies to compute κ in (19), in this work we use a heuristic approach which turns out in a good performance in the numerical examples. Firstly, a spatial oscillation indicator is computed:

$$\xi^i(x, \chi^i) = \operatorname{sign} \frac{\phi_h^i(x, \chi^i) - \phi_h^{i-1}(x, \chi^{i-1})}{\phi_h^{i-1}(x, \chi^{i-1}) - \phi_h^{i-2}(x, \chi^{i-2})}.$$

Note that $\xi^i(x, \chi^i) = 1$ if the iterative algorithm for computing the topological derivative is advancing monotonically in the preceding iterations, and $\xi^i(x, \chi^i) = -1$ otherwise. This indicator allows one to detect if there are oscillations in the iterative process. If there are oscillations the value for κ needs to be decreased, otherwise, it can be increased (up to a maximum of $\kappa = 1$). Since $\xi^i(x, \chi^i)$ is a spatial function, the information on the oscillations needs to be averaged so that a scalar value for κ can be obtained, which is done as follows:

$$
\begin{aligned}
\mu^i(x,\chi^i) &= \left\{ \begin{array}{cc} c_{\kappa 1}\kappa^{i-1} & \text{if } \xi^i(x,\chi^i) = 1 \\ c_{\kappa 2}\kappa^{i-1} & \text{if } \xi^i(x,\chi^i) = -1 \end{array} \right. , \\
\kappa^i &= \min\left[\left(\frac{\int_D \left(\mu^i(x,\chi^i)\right)^{c_{\kappa 3}}}{\int_D (\mu^i(x,\chi^i))} \right)^{-c_{\kappa 3}}, 1 \right],
\end{aligned}
$$

where $c_{\kappa 1} \geq 1$, $c_{\kappa 2} \leq 1$ and $c_{\kappa 3} \leq 1$ are algorithmic parameters. In the numerical examples c_{k1}=1.1, c_{k2}=0.5 and c_{k3}=0.1 are used. As a stopping criteria, we consider the evolution of the objective functional to minimize, the algorithm stops when after several iterations the functional has not decreased a certain percentage of its value. Also, a limit in the number of iterations to be performed is set.

4. Parallel implementation on distributed memory systems

At the practical level, the strategy described in the previous sections is implemented in a parallel finite element code capable of doing adaptive mesh refinement and coarsening in distributed memory machines. For this, a domain decomposition strategy is used for the finite element problem together with the Message Passing Interface library (MPI) which handles the communications between processors and subdomains. A second level of parallelism is introduced in order to be able to integrate data referred to stochastic analysis, as it will be explained in the following .

Let us just summarize here how the parallel domain decomposition of the finite element problem is handled: firstly a partition of the nodes of the mesh is performed, which assigns each node to a processor. This is followed by a distribution of elements, with the particularity that interface elements with nodes belonging to multiple subdomains are replicated in each of the subdomains. This means that the information of nodes adjacent to the interface between subdomains is also replicated in each of the subdomains, which results in the so-called ghost nodes. Ghost nodes are nodes which do not belong to a given subdomain, but which are replicated in the subdomain because they belong to interface elements.

In the rest of this section, the merging of the finite element strategy with an adaptive mesh refinement library and its application to topology optimization problems, together with a two-level parallelism approach for stochastic analysis are described.

4.1. Adaptive refinement analysis of the topology optimization problem

As it has been remarked previously, the main objective for using an adaptive mesh refinement and coarsening technique is to be able to concentrate the elements of the finite element mesh in the regions close to the interface between the stiff and the void materials. Thanks to this, we expect that the topology of the optimized solution can be captured with a high resolution at a relatively low computation cost.

4.2. Adaptive algorithm for parallel computing: `RefficientLib`

The adaptive mesh refinement strategy that we use is specifically targeted for distributed memory machines, and has been implemented in the adaptive mesh refinement Fortran 2003 library `RefficientLib.` The algorithmic details can be found in [34], but the main capabilities and how they have been used in this work are described next.

`RefficientLib` allows any parallel distributed memory finite element code to efficiently modify the mesh in a hierarchical manner (allowing to both refine and coarsen the finite element mesh) and in this way it allows to have a fine mesh only on the areas of the computational domain where it is required, thus alleviating the total computational cost by using coarse elements in non-important areas. Most importantly, it features good scalability up to thousands of processors, it is capable of doing load rebalancing when a processor gets too loaded, and it has an easy to use interface with a given finite element code (see [34]).

The adaptive refinement and coarsening algorithm is based on a nodal parallel partition. This means that each node of the mesh belongs to a processor, whereas an element can belong to multiple processors if it owns nodes from more than one subdomain. This feature introduces some particularities which need to be taken into account when designing the adaptive parallel algorithms.

Additionally, the fact that the algorithm results in meshes with hanging nodes needs to be taken into account in the finite element solver. Hanging nodes are nodes which appear in the faces of adjacent elements with different refinement levels (see Figure 1). If nothing is done, this results in non-conforming finite element spaces. Although this is an issue which can be easily dealt with by using selective discontinuous finite element spaces (see for instance [42]), in this work we favor the option of eliminating the nodal unknowns associated to hanging nodes by restricting their value to an interpolation of the nodal values of the finite element unknown in its parent nodes. As a consequence, hanging node values are not an unknown of the finite element problem, but they are postprocessed from nodal values in the parent nodes.

Another important issue, specially in topology optimization, is how the spatial fields are transfered between successively refined meshes. For this, the strategy used in [43] is adopted for the transmission of spatial fields. Particularly, two types of fields need to be transfered: the first type are fields which are stored in the nodes of the finite element mesh, in the solid mechanics problem a

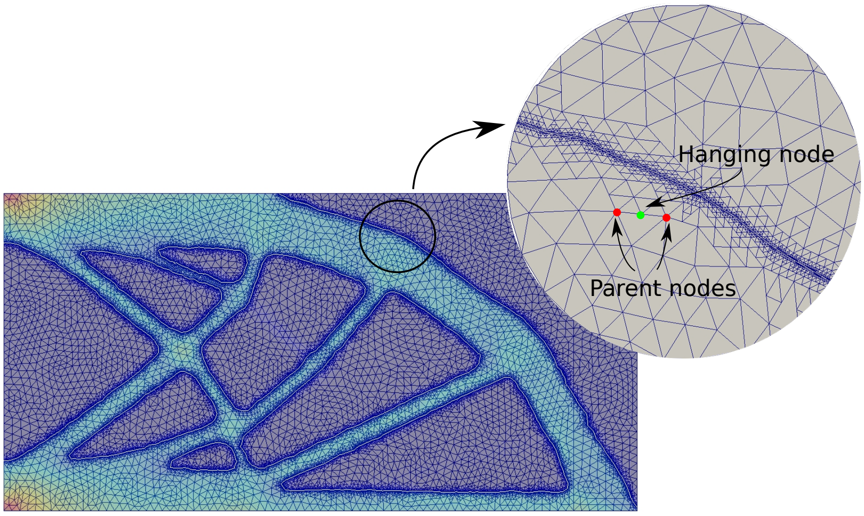


Figure 1: Hanging nodes in a finite element mesh.

typical example are the nodal displacements. In this case, the process for transferring the information between successive meshes is straightforward, since for nodes which are in both the old and the new mesh the field values do not change, and for new nodes they can be interpolated from values in the parent nodes. The second type of fields are fields which are stored in the quadrature points of each element. For solid mechanics, these fields typically correspond to stresses and strains. In topology optimization problems it also includes the characteristic function χ associated to each quadrature point. The transmission of data between successive meshes for this type of fields is a bit more complex to implement, because quadrature points of parent and children elements do not coincide in general. In our approach, we have solved this issue by means of an L^2 nodal projection process. Although this approach introduces some dissipation due to the several interpolation steps, it turns out in good results for the iterative topology optimization process.

4.2.1. Refinement criteria

In topology optimization, the part of the domain where one requires the largest resolution is the area surrounding the interface between the stiff and the void material. Thus, a simple refinement criteria can be set, where a number of layers of elements adjacent to the interface, represented by the level set function, are strongly refined. From the parallel execution point of view, this refinement criteria is implemented as follows (see [44] for a similar approach in free surface flows): a first step consists in identifying the elements which are cut by the level set interface, this is easily done by selecting elements with both positive and negative nodal level set values. Secondly, a number of layers of surrounding elements need to be selected for refinement. This requires some parallel communications because these layers might propagate from one parallel subdomain to another as illustrated in Figure 2. In order to address this issue, a nodal array is used to mark the nodes belonging to the layers of elements surrounding the level set interface. Successive layers of nodes are then marked one layer at a time, and the nodal array ghost values are communicated between neighbor processors before advancing to the next layer. This ensures that the result in a parallel execution coincides with the serial one.

Also, a progressive refinement approach is adopted, which means that the first few iterations of the topology optimization process are done with the original coarse mesh. Once the general shape of the solution has been obtained, the adaptive refinement process starts. This allows one to save a lot of computational effort, because the initial iterations which are far away from the final solution are performed in the coarse, cheap mesh. Thus, during first iterations large changes in topology are performed whereas accurate changes in the shape are obtained in the last iterations.

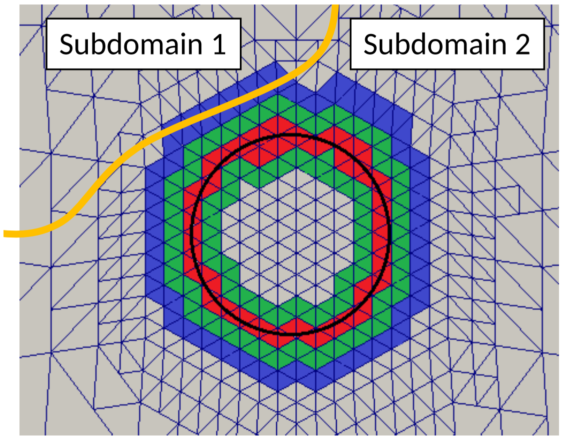


Figure 2: Propagation of the adaptive refinement criteria between parallel subdomains. 3 adjacent layers of elements are selected to be refined. A boundary node communication from subdomain 2 to subdomain 1 is required so that elements in subdomain 1 are also refined.

4.3. Two-level parallelism approach for stochastic analysis in distributed memory machines

In the previous sections the parallel implementation of the adaptive analysis of a deterministic topology optimization problem has been explained. This section deals with the parallel implementation when the adaptive topology optimization analysis is to be applied in a stochastic setting.

As explained previously, computing the topological derivative in stochastic analysis for a given topology requires to solve multiple problems, each of them corresponding to a stochastic collocation point of the sparse grid, and adding some function of the deterministic topological derivatives of each of the stochastic collocation points together. From the implementation point of view, several approaches are possible, but not all of them are suitable for large scale computing. Moreover, the fact that adaptive mesh refinement is used poses additional restrictions for the parallel strategy.

The approach we propose for dealing with the stochastic topology optimization problem consists in a two-level parallelism strategy which can be easily implemented in distributed memory machines using the MPI library. The proposed algorithm exploits both the domain decomposition and the stochastic parallelism levels and it overcomes the performance loss associated to disk access by completely avoiding it. Moreover, it allows to easily communicate information between simulations at different stochastic collocation points when using adaptive simulations.

The parallel approach is illustrated in Figure 3, and can be summarized as follows: instead of running an independent execution for each stochastic collocation point (and maybe in each execution exploiting the parallelism at the domain decomposition level), we propose to run a single execution in which stochastic and domain decomposition parallelism are exploited at the same time. For this, once the execution starts, the total number of available processors is distributed, firstly by assigning an equal number of processors to each stochastic collocation point, and secondly, in each stochastic collocation point, a domain decomposition algorithm is used to partition the finite element mesh into several parallel subdomains, each of them assigned to one processor.

At this point, the finite element problem is replicated in each stochastic collocation point, where in turn a subdomain partition of the finite element problem has been done. If a deterministic partitioning algorithm is used (in the sense that for a given finite element problem and number of processors it always returns the same distribution into subdomains), the subdomains are the same at each stochastic collocation point. This is very convenient, because it means that when the stochastic topological derivative is composed, communications need to be done only between processors associated to the same subdomain in different stochastic collocation points. In fact, composing the stochastic topological derivative requires of an all-reduce communication of the topological derivative field between the processors associated to the same subdomain. Although this operation can pose a bottleneck if a very large number of stochastic collocation points is used, it is still much more efficient than the alternative, which would consist in storing this information to disk for each stochastic collocation point and reading it afterwards.

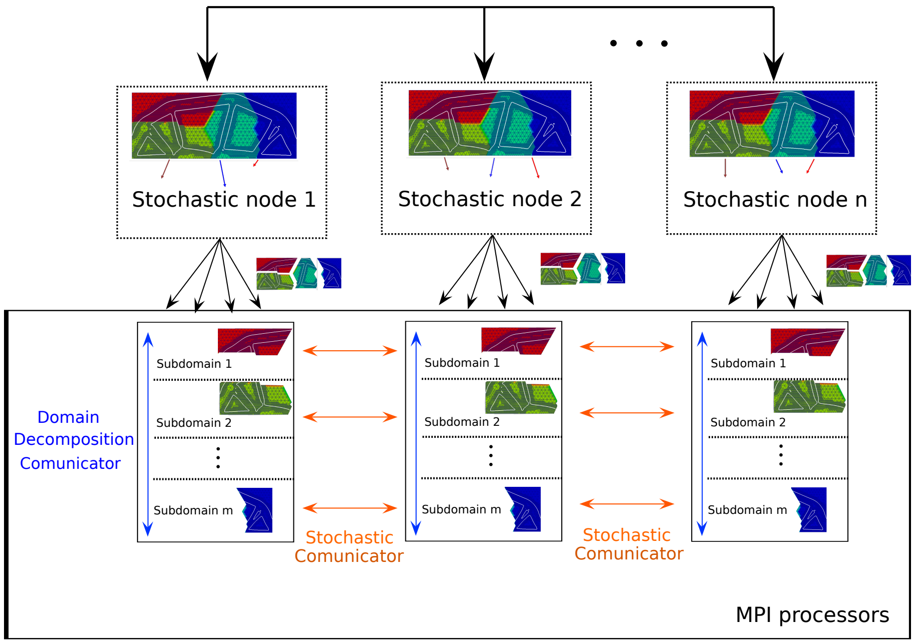


Figure 3: Two-level parallel approach. Several executions run at different stochastic collocation points are shown. Each of the executions is run in parallel using a domain decomposition approach. Additional communications are required for the stochastic analysis. The general communication pattern is shown in the lower part of the figure.

Moreover, this approach suits very well the adaptive refinement and coarsening strategy described in Subsection 4.2.1. Note that the refinement criteria depends exclusively on the values of the level set function (particularly on the position of the interface between the two types of material). In turn, the position of this interface depends on the stochastic topological derivative, which is shared and equal in every stochastic collocation point. Due to this, the refinement criteria will coincide in every stochastic collocation point. This means that the pattern of communications of the topological derivative between processors in different stochastic collocation points will not change in successive iterations of the topology optimization process. Even if load rebalancing is required at the domain decomposition parallelism level, this communication pattern will remain unchanged.

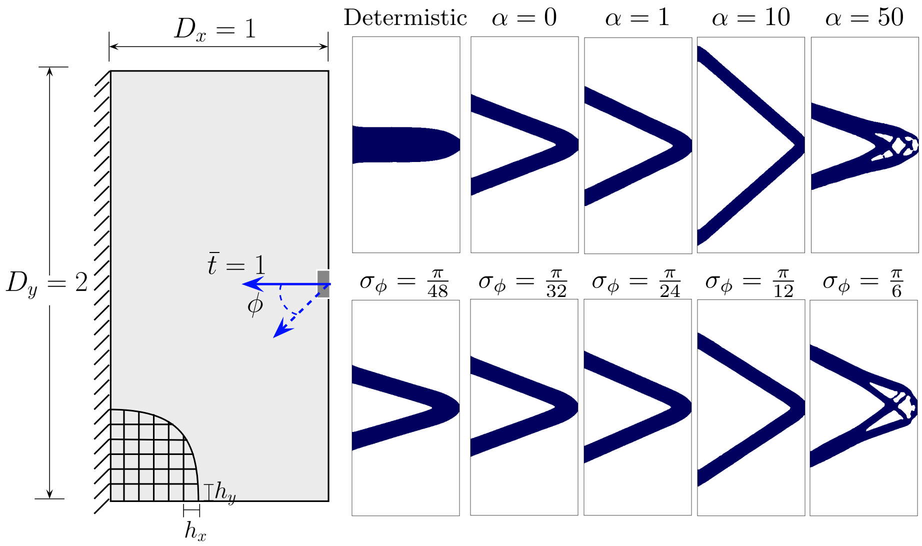


Figure 4: Design domain and boundary conditions for the beam-to-cantilever problem. The figure shows the deterministic and robust designs with different values of α with $\varphi \sim N(0, \pi/12)$ (upper) and values for a constant value of $\alpha = 10$ and different levels of uncertainty in the bearing of the unit load σ_ϕ with $\alpha = 4$ (lower).

5. Numerical experiments

In this section several numerical examples are presented, which illustrate the performance of the proposed strategy. Firstly, some benchmark examples are used to validate the numerical implementation. Secondly, large scale examples are used in order to test the capabilities of the proposed strategy to solve engineering problems.

5.1. Beam-to-cantilever problem

The beam-to-cantilever problem consists in the shape optimization of a two-dimensional cantilever under uncertain loading conditions. Such a benchmark is widely used [45, 14] to show the effects of uncertain loading in robust optimal design. The left edge of the cantilever is anchored and a unit force with uncertainty in direction, centered in the horizontal line, is applied at the middle of the right edge. The design domain is a 1x2 rectangle which is tessellated using P_1 triangular elements. Uncertain loading, boundary conditions and tessellation of the design domain are depicted in Figure 4. The material parameters are $\mathrm{E}^+ = 1$, $\mathrm{E}^- = 10^{-6}$ and $\nu^+ = \nu^- = 0.3$.

The configuration and the numerical resolution of the robust shape optimization problem is as follows. The target volume is set to 15% of the initial design domain. The direction ϕ of the unit-load follows a probabilistic distribution centered at the horizontal line, $\phi = 0$, with the deterministic loading state as the mean value $\mu_\phi = 0$. The influence of the level of uncertainty on the robust shape design is shown in Figure 4 by the consideration of $\sigma_\phi = \{\pi/6, \pi/12, \pi/24, \pi/32, \pi/48\}$ as standard deviation values. For this, a level 3 quadrature, giving raise to 3 different collocation stochastic points is used. As expected, it can be observed that as the influence of the level of uncertainty increases, the angle of separation between both branches of the topology increases. When the influence of uncertainty becomes very large ($\alpha = 50$ in the first row, $\sigma_\phi = \pi/6$ in the second row), the algorithm converges to locally optimal solutions which minimize the standard deviation, but which are far from optimality with respect to the mean compliance value.

5.2. Michell-type structure design

This numerical example presents the results for the robust topology optimization of the well known michell-type structure problem. The boundary conditions and the domain discretization are shown in Figure 5. The design domain has a roller support in the bottom right-hand corner and a fixed support in the bottom left-hand corner. The structure is subjected to three uncertain concentrated loads applied at its bottom. The magnitude of the loads are characterized by three independent random variables. These random variables are assumed to follow normal distributions with mean values $\mu_{g_1} = \mu_{g_2} = \mu_{g_3} = 1$ and standard deviations $\sigma_{g_1} = 0.5$, $\sigma_{g_2} = 0.1$ and $\sigma_{g_3} = 0.2$.

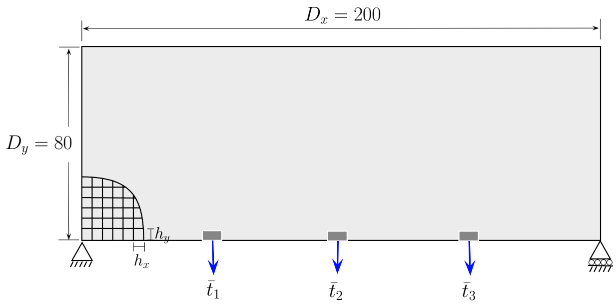
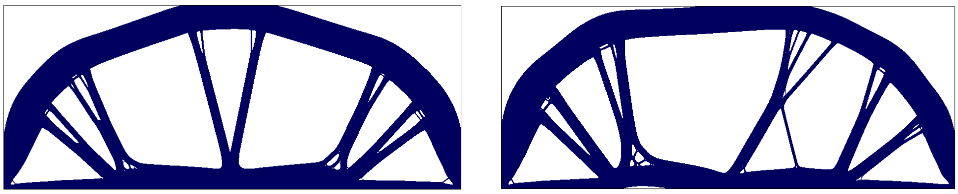


Figure 5: Design domain and boundary conditions for the Michell-type structure problem.



(a) DTO (b) Robust design

Figure 6: The Michell-type structure design problem: (a) DTO design and (b) Robust design. The adaptive meshes allow to capture details of the topology in specific regions.

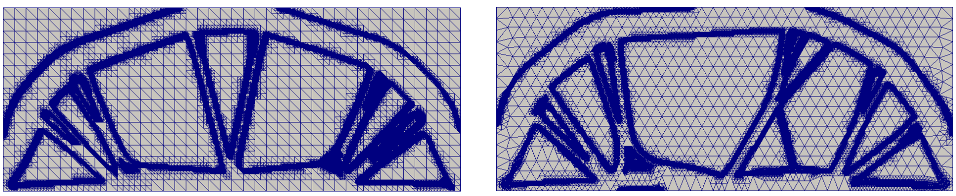
Regarding the stochastic discretization, a level 3 stochastic quadrature rule is used, giving raise to 25 different collocation stochastic points. The material parameters are $\mathrm{E}^+ = 1$, $\mathrm{E}^- = 10^{-6}$ and $\nu^+ = \nu^- = 0.3$.

The solutions obtained with a deterministic design and a robust design ($\alpha = 5$) are shown in Figure 6. The deterministic approach should provide a symmetric design, although lack of symmetry in the topology details is observed due to the lack of symmetry of the initial mesh. On the contrary, the robust design is non-symmetric due to the larger uncertainty in the left-most load. Figure 7 shows the adaptive meshes used for the topology optimization problem. The adaptive refinement and coarsening strategy allows to heavily refine the mesh in the area where topological details occur, and the computational cost is kept low because a coarse mesh can be used in the rest of the domain. Note that different adaptive meshes are used for the deterministic and the robust design cases. Figure 8 shows the convergence of the compliance mean and compliance standard deviation. The proposed iterative algorithm rapidly converges to a close to optimal topology. Every 50 iterations, a new refinement level is introduced (departing from the original coarse mesh at the initial iteration), which causes the small oscillations in the objective function observed in the figure.

5.3. Performance evaluation

5.3.1. Non-Adaptive vs. Adaptive performance

In this section we compare the performance of the adaptive vs. the non-adaptive version of the topology optimization algorithm. For this, we use the deterministic Michell-type structure



(a) DTO (b) Robust design

Figure 7: Adaptive meshes used for the michell type structure. The mesh is heavily refined in the area close to the interface.

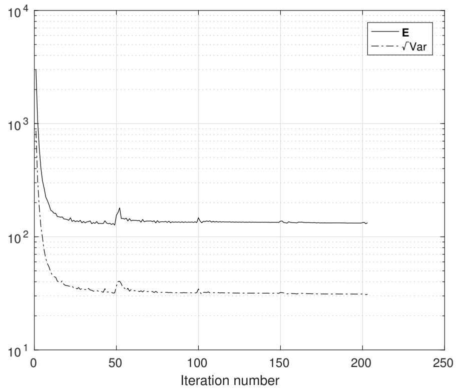


Figure 8: Evolution of the compliance mean and standard deviation for the robust design of the michell type structure

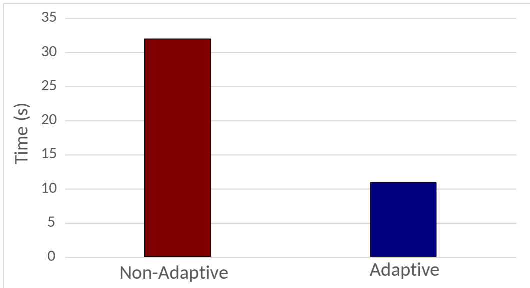


Figure 9: Non-adaptive versus adaptive computational time comparison for two equivalent precision solutions.

problem. For the non-adaptive simulation, we use a mesh of 36000 elements which provides an accurate enough resolution for the problem of interest, and we perform the topology optimization process. The simulation runs for approximately 31 seconds (141 topology optimization iterations) until the compliance value converges and the simulation stops.

The adaptive simulation departs from a mesh of 2276 elements. The refinement criteria is the following: for the first 50 iterations, the coarse mesh is used. In iterations between 50 and 100, at most one level of refinement is allowed in the region surrounding the material interface. From the 100th iteration on, a maximum of two levels of mesh refinement are allowed in the region surrounding the material interface. The final level of accuracy of the refined mesh is equivalent to the one of the non-adaptive mesh (same element size at the interface). The simulation runs for approximately 11 seconds (128 topology optimization iterations) until the compliance value converges and the simulation stops. The final topology optimized designs are qualitatively very similar in the adaptive and non-adaptive cases. A graphical computation time comparison is shown in Figure 9. For the same level of accuracy of the final solution, the adaptive mesh refinement strategy can provide huge savings in computational cost when provided with a suitable refinement criteria.

5.3.2. Parallel Scalability tests

In this section we present the scalability tests for the proposed algorithm. The scalability tests are done on the Michell-type geometry, for which several steps of the adaptive topology optimization process are performed and both weak and strong scalability are measured. For the linear system solver, the HYPRE preconditioner (see [46]) together with a conjugate gradient method are used, both of them accessed through the PETSc library (see [47]).

The first set of scalability tests are run on the LaPalma Supercomputer, located at the Instituto de AstroFísica de Canarias. The LaPalma supercomputer is composed of 252 IBM dx360 M4 compute nodes. Each node has sixteen E5-2670 cores at 2.6GHz with 32 GB of RAM memory. Available RAM per core (discounting the operative system) is approximately 1.4 Gbytes. The maximum number of tasks per job is 2400.

The first scalability test is a deterministic case with adaptive mesh refinement, where several refinement plus topology optimization steps are performed. The simulation is performed by using from 8 to 256 CPUs and the speed up is measured. The results are shown in Figure 10-a). The speed up is very good up to 128 processors and starts to degrade at 256 processors (speed up of 210). The reason for this is that, for the test case run, the number of finite element nodes per processor when using 256 processors very low, approximately 20000, which results in subdomain border communications becoming a bottleneck both for the linear system solution and for the adaptive refinement process communications. Also, load rebalancing becomes more frequently necessary when the number of processors increases, which also affects performance. Cases with a larger number of nodes per processor could not be run for this strong scalability test because they did not fit in the 1.4 Gbyte RAM per processor memory limitation when using only 8 processors. Although performance is good, it clearly limits the scalability for a larger number of processors in this system. However, this result can be improved by combining domain decomposition and stochastic parallelism as it is shown in the next test.

The second scalability test consists in solving the same case but in a stochastic setting, particularly by considering stochastic values for the forces modulus. The used stochastic integration rule requires 25 stochastic integration points. The domain discretization is the same as in the previous example, so a minimum of 8 processors per stochastic integration point are required. As done previously, we increase the number of processors per stochastic integration point and we measure the speed up. The results are shown in Figure 10-b). It can be observed that good performance is obtained up to 2345 processors (with a speed up of approximately 1800).

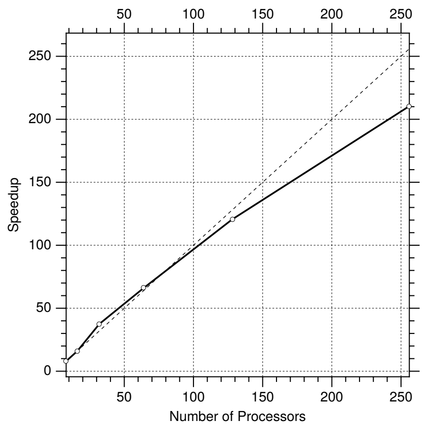
The third scalability test consists in solving again the Michell-type problem with the following particularities: a coarser spatial discretization is used, so that the problem can fit into a single processor's RAM memory. In this case, we do not increase the number of domain decomposition partitions, but we consider a case in which we are interested in increasing the accuracy of the stochastic integration rule. For this, we progressively increase the number of stochastic integration points (together with the number of processors), and we measure the weak scalability. The results are shown in Figure 10-c). The weak scalability plot shows an expected jump when switching from 1 (deterministic analysis, no communications required) to 25 stochastic points, but then shows an optimal behavior and remains almost flat up to 1233 stochastic integration points. However, the computational time starts to increase when the next stage of the integration rule is used (2097 stochastic integration points). Although the LaPalma Supercomputer can be fully exploited in this case, for larger processor count the stochastic communications will start to penalize efficiency. This bottleneck can be overcome by combining stochastic and domain decomposition parallelism, as we do in the next example.

The fourth scalability test is run in the Curie Supercomputer. The Curie system is composed of 5040 Bullx B510 nodes, each equiped with 16 Intel E5-2680 Sandy-Bridge processors working at 2.7 GHz, and 32 GB of NUMA RAM memory. The total processor count is 80640, although the job submission is limited to approximately 25000 processors. The objective of this test is to show the capability of the proposed algorithm to properly scale up to the order of the tenths of thousands of processors. In this test the Michell case is solved by using a partition into 128 subdomains, and then the accuracy of the stochastic integration rule is progressively increased, ranging from 1 to 165 stochastic collocation points. Weak scalability is measured with processor count which ranges from 128, corresponding to 1 stochastic node, to 21120 processors, corresponding to 165 stochastic nodes. Results are shown in Figure 10-d). The attained performance is almost optimal for this case, and the Curie Supercomputer can be fully exploited without performance losses.

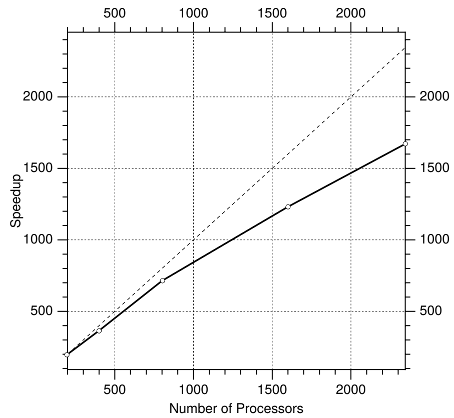
5.4. A large scale case. Airplane bearing bracket under random loading

The bearing bracket is a common dynamic component that interfaces with moving parts, and has to conform to a certain geometrical envelope as well as being able to sustain large loading forces in various directions. The optimization of this design can provide weight savings and reduce fuel consumption of airplanes.

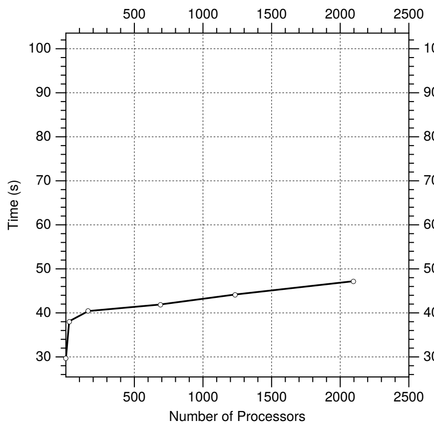
Figure 11-a) shows the geometry and the boundary conditions of the Airplane Bearing Bracket (ABB) benchmark. The Elastic Modulus (E) is 200 GPa, the Poisson Ratio is 0.27 and the material is assumed to be isotropic linear elastic. The boundary conditions consist in a rigid plate which is



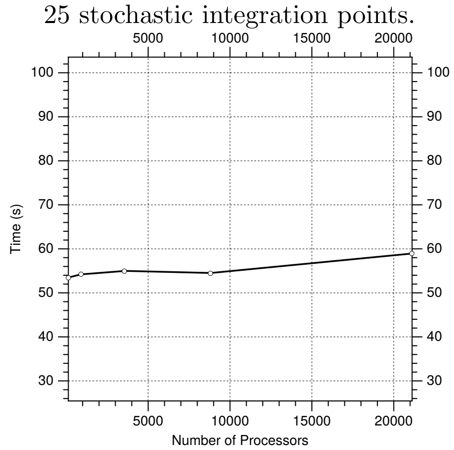
a) Strong scalability.



b) Strong scalability,
25 stochastic integration points.



c) Weak scalability.



d) Weak scalability,
partition of 128 subdomains.

Figure 10: Scalability results

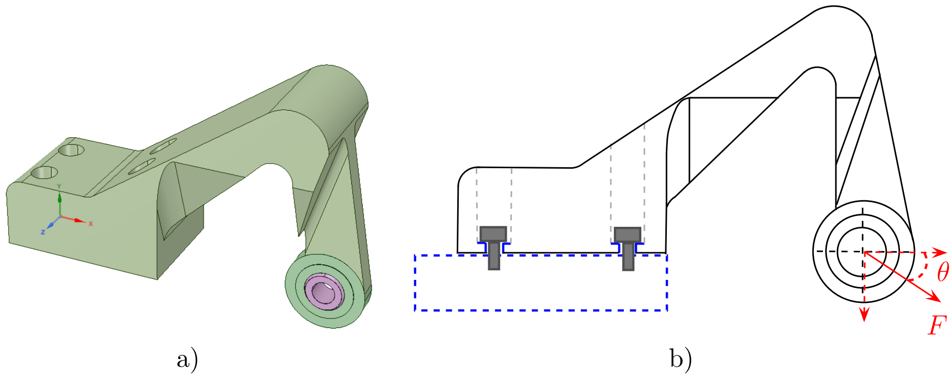


Figure 11: Design domain and boundary conditions for the Airplane Bearing Bracket (ABB) problem (a) and loading conditions (b).

attached to the bracket by means of four high strength bolts. Regarding the loading conditions, a load with a $\pi/4$ inclination with respect to the horizontal axis is considered, with an orientation standard deviation of $\sigma_\phi = \pi/12$. Uncertainty in the load direction is considered only in the z or spanwise direction. The loading conditions are depicted in Figure 11-b). A level 3 quadrature, giving raise to 3 different collocation stochastic points, is used.

Three cases are evaluated: in the first case an objective volume fraction of 0.5 is set, in the second one the objective volume fraction is 0.2, while in the third one the objective volume fraction is 0.1. Since both the geometry and the loads are symmetric, only one half of the bracket is simulated, with symmetry boundary conditions in the symmetry plane. The numerical simulations are performed departing from a finite element mesh of 1.2 million elements, and are progressively refined until a mesh of between 50 million elements (for the 0.5 volume fraction case) and 120 million elements (for the 0.1 volume fraction case) is obtained. The simulations run for between 75 and 100 iterations until the final solution is obtained. The cases were run in the LaPalma Supercomputer by using 1536 processors and took roughly 20 minutes to complete.

The 0.5 volume fraction solution is shown in Figure 12. The stiff material is mostly positioned around the high strength bolts, and the region surrounding the head of the bracket. The obtained geometry is quite simple without complex patterns. The 0.2 volume fraction case is represented in Figure 13. The final result presents a more complex pattern with few material in the high strength bolts area. Finally, Figure 14 shows the results for the 0.1 volume fraction. The result is similar to the previous one, but several additional holes appear in the intermediate area due to the scarcity of stiff material. Details of the obtained geometry are shown in Figure 15.

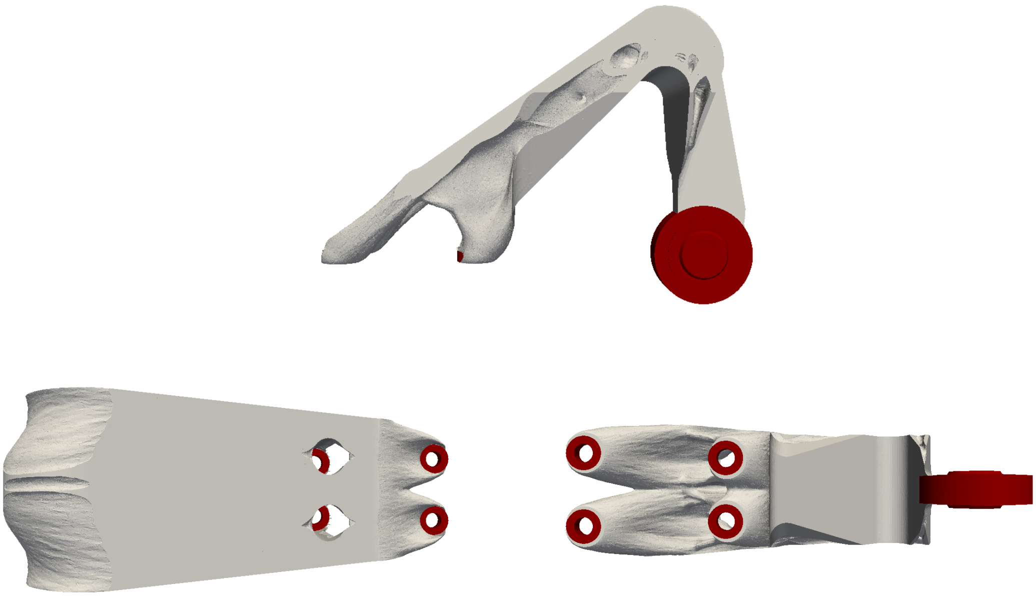


Figure 12: Results for the airplane bearing bracket, 0.5 objective volume fraction. Top: lateral view. Bottom: top and bottom views.

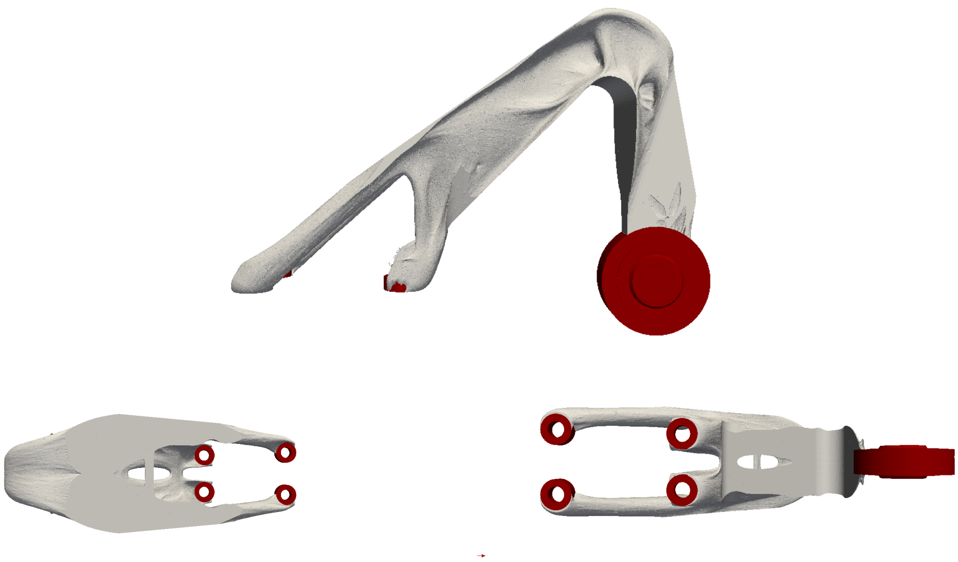


Figure 13: Results for the airplane bearing bracket, 0.2 objective volume fraction. Top: lateral view. Bottom: top and bottom views.

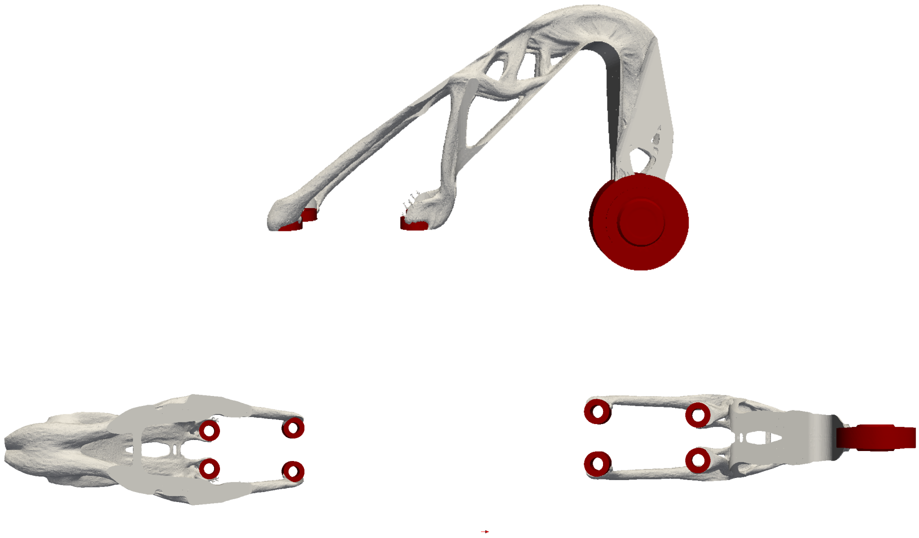


Figure 14: Results for the airplane bearing bracket, 0.1 objective volume fraction. Top: lateral view. Bottom: top and bottom views.

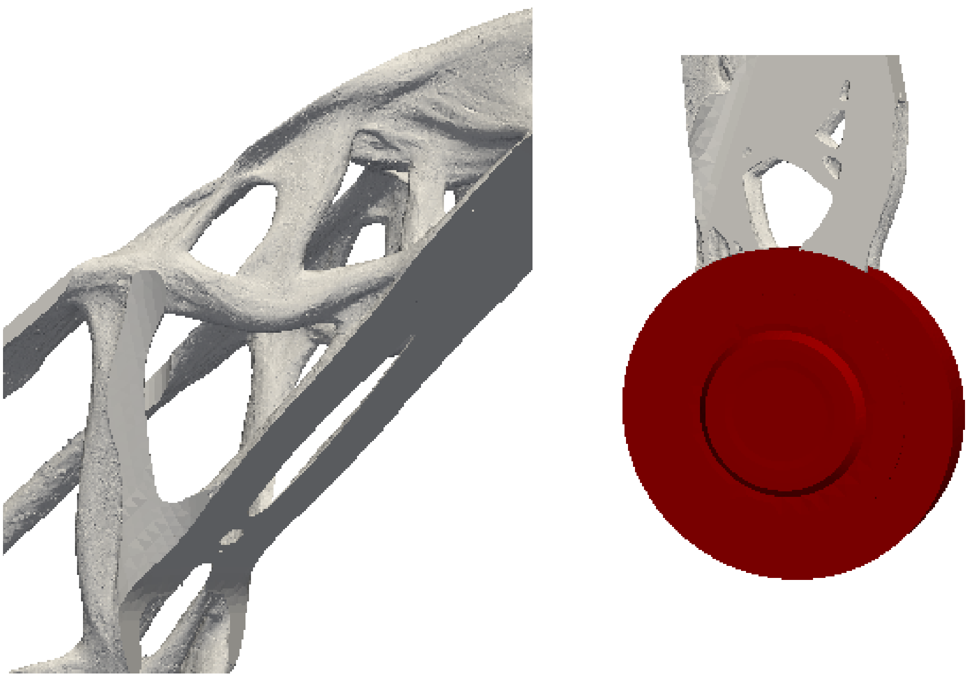


Figure 15: Results for the airplane bearing bracket, 0.1 objective volume fraction. Details of the obtained geometry.

6. Conclusion

In this work we have presented the strategy for stochastic topology optimization in high performance computing distributed memory clusters. The main idea is to use adaptive computational meshes and to exploit parallelism both at the domain decomposition level and at the stochastic collocation points level.

The topology optimization process consists in the iterative computation of the topological derivative coupled with a level set strategy for the definition of the stiff and soft materials, which allows to keep a sharp tracking of the interface.

For the adaptive mesh refinement strategy, a finite element method coupled with an adaptive refinement library has been used. The main advantage of this approach is that it allows to dynamically concentrate the refined elements in the area close to the material interface, thus saving computational effort. A domain decomposition strategy for the distribution of load amongst processors is also used for parallelism, which is coupled with a load rebalancing step when the adaptive mesh refinement starts. At the stochastic level, parallelism is exploited by evaluating separately each of the stochastic collocation points, with communications being done only for the computation of the topological derivative mean and standard deviations.

Several numerical examples illustrate the capability of the proposed methodology to save computational effort, and the computational scalability of the method is assessed in examples run in high performance computing clusters. The proposed algorithm is capable of efficiently exploiting these parallel distributed memory systems, with scalability being assessed in the LaPalma Supercomputer for up to 2,300 processors, and in the Curie Supercomputer for up to 21,000 processors. It is also capable of managing finite element meshes of the order of hundreds of millions of elements, with savings in wall-clock time of up to four orders of magnitude with respect to a naive serial, non-adaptive approach .

Acknowledgements

This research was partially supported by the AEI/FEDER and UE through the ELASTIC-FLOW (Ref. DPI2015-67857-R) and GPUSIM (Ref. DPI2016-77538-R) projects, and by the "Fundación Séneca-Agencia de Ciencia y Tecnología de la Región de Murcia" under the contract 19274/PI/14. J. Baiges acknowledges the support of the Spanish Government through the Ramón y Cajal grant RYC-2015-17367. The support of the "Red Española de Supercomputación" and the "European PRACE" computational network is acknowledged.

References

[1] M. P. Bendsøe, O. Sigmund, Topology Optimization – Theory, Methods, and Applications, second ed., Springer-Verlag Berlin Heidelberg, 2004.

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

[3] J. D. Deaton, R. V. Grandhi, A survey of structural and multidisciplinary continuum topology optimization: post 2000, Struct. Multidiscip. Optim. 49 (2013) 1–38.

[4] N. D. Lagaros, M. Papadrakakis, Robust seismic design optimization of steel structures, Struct. Multidiscip. Optim. 33 (2007) 457–69.

[5] B. D. Youn, K. K. Choi, R.-J. Yang, L. Gu, Reliability-based design optimization for crashworthiness of vehicle side impact, Struct. Multidiscip. Optim. 26 (2004) 272–83.

[6] J. H. Zhu, W. H. Zhang, L. Xia, Topology Optimization in Aircraft and Aerospace Structures Design, Arch. Comput. Meth. Eng. 23 (2016) 595–622.

[7] M. Schevenels, B. S. Lazarov, O. Sigmund, Robust topology optimization accounting for spatially varying manufacturing errors, Comput. Meth. Appl. Mech. Eng. 200 (2011) 3613–27.

[8] J. Martínez-Frutos, D. Herrero-Pérez, M. Kessler, F. Periago, Risk-averse structural topology optimization under random fields using stochastic expansion methods, Comput. Meth. Appl. Mech. Eng. 330 (2018) 180–206.

[9] S. Chen, W. Chen, S. Lee, Level set based robust shape and topology optimization under random field uncertainties, Struct. Multidisc. Optim. 41 (2010) 507–24.

[10] J. Martínez-Frutos, D. Herrero-Pérez, M. Kessler, F. Periago, Robust shape optimization of continuous structures via the level set method, Comput. Meth. Appl. Mech. Eng. 305 (2016) 271–91.

[11] D. W. Kim, B. Kwak, Reliability-based shape optimization of two-dimensional elastic problems using BEM, Comput. Struct. 60 (1996) 743–50.

[12] A. T. Beck, W. J. S. Gomes, R. H. Lopez, L. F. F. Miguel, A comparison between robust and risk-based optimization under uncertainty, Struct. Multidiscip. Optim. 52 (2015) 479–92.

[13] J. K. Guest, T. Igusa, Structural optimization under uncertain loads and nodal locations, Comput. Meth. Appl. Mech. Eng. 198 (2008) 116–24.

[14] M. Carrasco, B. Ivorra, A. M. Ramos, Stochastic topology design optimization for continuous elastic materials, Comput. Meth. Appl. Mech. Eng. 289 (2015) 131–54.

[15] N. Aage, E. Andreassen, B. S. Lazarov, O. Sigmund, Topology optimization using PETSc: An easy-to-use, fully parallel, open source topology optimization framework, Struct. Multidiscip. Optim. 51 (2015) 565–72.

[16] N. Aage, E. Andreassen, B. S. Lazarov, O. Sigmund, Giga-voxel computational morphogenesis for structural design, Nature 550 (2017) 84–6.

[17] J. Alexandersen, O. Sigmund, N. Aage, Large scale three-dimensional topology optimisation of heat sinks cooled by natural convection, Int. J. Heat Mass Transfer 100 (2016) 876–91.

[18] T. Borrvall, J. Petersson, Large-scale topology optimization in 3D using parallel computing, Comput. Meth. Appl. Mech. Eng. 190 (2001) 6201–29.

[19] A. Mahdavi, R. Balaji, M. Frecker, E. M. Mockensturm, Topology optimization of 2D continua for minimum compliance using parallel computing, Struct. Multidisc. Optim. 32 (2006) 121–32.

[20] J. Martínez-Frutos, P. Martínez-Castejón, D. Herrero-Pérez, Efficient topology optimization using GPU computing with multilevel granularity, Adv. Eng. Softw. 106 (2017) 47–62.

[21] J. Martínez-Frutos, D. Herrero-Pérez, GPU acceleration for evolutionary topology optimization of continuum structures using isosurfaces, Comput. Struct. 182 (2017) 119–36.

[22] M. Wlotzka, V. Heuveline, Energy-efficient multigrid smoothers and grid transfer operators on multi-core and GPU clusters, J. Parallel Distrib. Comput. 100 (2017) 181–92.

[23] B. Lazarov, M. Schevenels, O. Sigmund, Topology optimization considering material and geometric uncertainties using stochastic collocation methods, Struct. Multidisc. Optim. 46 (2012) 597–612.

[24] P. E. Hadjidoukas, P. Angelikopoulos, C. Papadimitriou, P. Koumoutsakos, π4U: A high performance computing framework for Bayesian uncertainty quantification of complex models, J. Comput. Phys. 284 (2015) 1–21.

[25] G. Stavroulakis, D. G. Giovanis, V. Papadopoulos, M. Papadrakakis, A GPU domain decomposition solution for spectral stochastic finite element method, Comput. Meth. Appl. Mech. Eng. 327 (2017) 392–410.

[26] R. Bellman, Adaptive control processes: a guided tour, Princeton University Press, 1961.

[27] R. Bellman, Dynamic Programming, Courier Dover Publications, 2003.

[28] E. D. Sturler, G. H. Paulino, S. Wang, Topology optimization with adaptive mesh refinement, in: Int. Conf. on Computation of Shell and Spatial Structures (IASS-IACM 2008), Ithaca, NY, USA, 2008, pp. 1–4.

[29] A. Lambe, A. Czekanski, Topology optimization using a continuous density field and adaptive mesh refinement, Int. J. Numer. Meth. Eng. 113 (2018) 357–73.

[30] N. Kikuchi, K. Y. Chung, T. Torigaki, J. E. Taylor, Adaptive finite element methods for shape optimization of linearly elastic structures, Comp. Meth. Appl. Mech. Eng. 57 (1986) 67–89.

[31] E. Ramm, K. Maute, S. Schwarz, Adaptive topology and shape optimization, in: S. Idelsohn, E. Oñate, E. Dvorkin (Eds.), Fourth World Congress on Computational Mechanics, volume Computational Mechanics: New Trends and Applications, CIMNE, Barcelona, Spain, Buenos Aires, Argentina, 1998.

[32] A. Nana, J.-C. Cuillière, V. Francois, Towards adaptive topology optimization, Adv. Eng. Soft. 100 (2016) 290–307.

[33] J. K. Guest, L. C. S. Genut, Reducing dimensionality in topology optimization using adaptive design variable fields, Int. J. Numer. Meth. Eng. 81 (2010) 1019–45.

[34] J. Baiges, C. Bayona, Refficientlib: An Efficient Load-Rebalanced Adaptive Mesh Refinement Algorithm for High-Performance Computational Physics Meshes, SIAM J. Sci. Comput. 39 (2017) 65–95.

[35] A. A. Novotny, R. A. Feijo, E. Taroco, C. Padra, Topological sensitivity analysis, Comp. Meth. Appl. Mech. Eng. 192 (2003) 803–29.

[36] S. M. Giustia, A. Ferrer, J. Oliver, Topological sensitivity analysis in heterogeneous anisotropic elasticity problem. Theoretical and computational aspects, Comp. Meth. Appl. Mech. Eng. 311 (2016) 134–50.

[37] A. J. Torii, A. A. Novotny, R. B. dos Santos, Robust compliance topology optimization based on the topological derivative concept, Int. J. Numer. Meth. Eng. 106 (2016) 889–903.

[38] A. A. Novotny, J. Sokołowski, Topological Derivatives in Shape Optimization, Interaction of Mechanics and Mathematics, Springer-Verlag Berlin Heidelberg, 2013.

[39] J. Martínez-Frutos, D. Herrero-Pérez, Large-scale robust topology optimization using multi-GPU systems, Comp. Meth. Appl. Mech. Eng. 311 (2016) 393–414.

[40] F. Nobile, R. Tempone, C. G. Webster, An Anisotropic Sparse Grid Stochastic Collocation Method for Partial Differential Equations with Random Input Data, SIAM Journal on Numerical Analysis 46 (2008) 2411–42.

[41] S. A. Smolyak, Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions, Sov. Math. Doklady 4 (1963) 240–3.

[42] S. Badia, J. Baiges, Adaptive Finite Element Simulation of Incompressible Flows by Hybrid Continuous-Discontinuous Galerkin Formulations, SIAM J. Sci. Comput. 35 (2013) 491–516.

[43] J. Baiges, R. Codina, Variational Multiscale error estimators for solid mechanics adaptive simulations: An Orthogonal Subgrid Scale approach, Comput. Meth. Appl. Mech. Eng. 325 (2017) 37–55.

[44] J. Baiges, R. Codina, A. Pont, E. Castillo, An adaptive Fixed-Mesh ALE method for free surface flows, Comput. Meth. Appl. Mech. Eng. 313 (2017) 159–88.

[45] F. D. Gournay, G. Allaire, F. Jouve, Shape and topology optimization of the robust compliance via the level set method, ESAIM: Control, Optimisation and Calculus of Variations 14 (2008) 43–70.

[46] A. H. Baker, R. D. Falgout, T. V. Kolev, U. M. Yang, Scaling Hypre's Multigrid Solvers to 100,000 Cores, Springer-Verlag, London, 2012, pp. 261–79.

[47] S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, PETSc Web page, Available from: http://www.mcs.anl.gov/petsc, 2015.