Exploring Topology Optimization on Hierarchical Meshes by Scaled Boundary Finite Element Method

***Adrian Egger[1], Albert Saputra[2], Savvas Triantafyllou[3], and Eleni Chatzi[1]**

[1]Institute of Structural Engineering, ETH Zurich, Switzerland
[2]Centre for Infrastructure Engineering and Safety, University of New South Wales, Australia
[3]Centre for Structural Engineering and Informatics, The University of Nottingham, UK

*Presenting and corresponding author: egger@ibk.baug.ethz.ch

Abstract

In this contribution, we aim at accelerating topology optimization by recasting the forward problem into a form that directly interacts with the structural solver: The polytope nature of SBFEM elements is exploited on quad-/octree meshes to alleviate issues associated with hanging nodes. Furthermore, a balancing operation applied to the mesh results in a manageable number of precomputable element configurations, which significantly accelerates the forward analysis. The analysis mesh for each optimization iteration is obtained via automated image-based decomposition of the design variables.

A number of benefits arise from this combination of methods. The ease with which higher-order elements may be incorporated, coupled with the use of unstructured meshes, combats the formation of checker-boarding. Second, computational effort only arises where required by the problem definition, since adaptivity is automatically provided. Third, numerical examples in both 2D and 3D indicate that the amount of degrees of freedom present during analysis is reduced by more than an order or magnitude.

Keywords: Topology optimization, SBFEM, Hierarchical meshes.

Introduction

The adoption of Topology Optimization (TO) into engineering practice stems from the ever-increasing need for the sustainable, economical and accountable use of resources across diverse sectors, e.g., construction, aerospace and automotive. The solution of a computationally expensive forward problem is typically fundamental to TO of continuum structures [4], which aims at defining an optimal structural layout subject to constraints. The conventional approach employing a fine grid of design variables, with values of 0 and 1 corresponding to void and solid, respectively, and whose layout remains constant across all iterations, does not reflect the nature of the evolving topology and leads to an excessive computational toll. Several adaptive schemes have been proposed to alleviate this computational burden [8, 19, 27]. In [17] tree-based meshing strategies are explored. Unfortunately, therein the treatment of hanging nodes severely affects the computational efficiency, which the herein proposed scheme remedies.

This work is motivated by the adoption of numerical methods capable of treating polytope-elements to alleviate issues commonly associated with hanging nodes. The proposed scheme is suited for fusion with, but not limited to, the boundary element method [14], polygonal finite element method [26], conforming shape functions [7] or virtual element method [3]. Here, we employ the scaled boundary finite element method (SBFEM) [24], a semi-analytical numerical method that permits the treatment of star-convex polytopes by introducing a scaling center in every element, thereby retrieving an analytical solution in radial direction, while only necessitating discretization of element boundaries. This attribute permits the construction of hp-elements. The combination of higher order elements and unstructured meshes combats the formation of checker-boarding [4]. Although a mixed SBFEM has been employed for solving TO problems of incompressible materials [12], it follows the conventional fine-grid approach to TO and limits its treatment to linear elements and 2D applications.

The SBFEM has proven itself as a remarkably versatile tool in automatic image-based stress analysis [15, 21]. Such hierarchical meshes arising from tree-like image decompositions drastically reduce the amount of degrees of freedom (DOFs) present, which accelerates the solution of the forward problem by alleviating computational effort and memory requirements. Image decomposition techniques, within the context of TO, produce fewer DOFs when material transition zones are eliminated. Hence, bi-directional evolutionary structural optimization (BESO) [10], sequential element rejection and admission (SERA) [2], solid isotropic microstructure with penalization (SIMP) combined with grayscale filters [1, 13] and level-set methods [16], for example, represent suitable algorithms. Since image decomposition operates on the design variables to produce analysis-ready meshes at each iteration, this family of techniques only requires interchanging the solver of the forward problem. Hence, incorporation into existing code bases requires minimal modification. Furthermore, as tree-like image decomposition techniques [5] utilize the uniformity of an element as a criterion for subdivision, multi-material TO [20] may be incorporated through extension to color-aware tree-like decompositions [11].

This work is structured as follows: The pertinent theory is provided first. This entails the salient features of TO and automated image segmenting into analysis-ready hierarchical meshes. These meshes, consisting of significantly fewer DOFs than traditional fine-grid approaches, are employed during the solution of the forward problem and contribute significantly to the reduction in computational cost. Subsequently, the proposed scheme is introduced in detail. Its novelty is i) the fusion of TO with automated image segmenting schemes, ii) the use of polytope-elements constructed by SBFEM to alleviate issues associated with hanging nodes, iii) color-encoding of domain and analysis parameters into the input images and iv) the extension to SBFEM-powered 3D TO. Based on three numerical examples, the proposed scheme is then verified, showcased and discussed. Novelties i) and ii) are demonstrated to significantly reduce computational effort and memory requirements, when compared to traditional fine-grid approaches. Finally, we offer conclusions and directions for future work.

Theory

Topology Optimization for Compliance Problems

In this paper we consider TO with focus on compliance minimization when subjected to a volume constraint, i.e., maximizing the structural stiffness when only a fraction of the original volume is available:

$$\begin{aligned} \min_{\mathbf{x}} &: c(\mathbf{x}) = \mathbf{U^T K U} = \sum_{e=1}^{N} g(x_e)\mathbf{u_e^T k_0 u_e} \\ \text{subject to} &: V(\mathbf{x})/V_0 = f \\ &: \mathbf{KU} = \mathbf{F} \\ &: \mathbf{0} \leq \mathbf{x} \leq \mathbf{1} \end{aligned} \tag{1}$$

where the set of N design variables $\mathbf{x}$ spans the design domain and specifies the material distribution. Three conditions are imposed, constraining the optimal solution. First, each element of $\mathbf{x}$, denoted by x_e, must fall within limit values of 0 and 1, which correspond to void and solid regions respectively. Second, a user prescribed volume fraction f follows as the ratio of current volume $V(\mathbf{x})$ to initial volume V_0. Third, the displacement field $\mathbf{U}$, required to compute the compliance c, follows from the solution of the forward problem in 3D elastostatics. $\mathbf{K}$ denotes the corresponding stiffness matrix and $\mathbf{F}$ the load vector respectively. The compliance can either be calculated globally or locally, as a summation of element contributions. The elemental nodal displacements are denoted by $\mathbf{u_e}$ and the corresponding stiffness matrix, calculated with Young's modulus equal to one, i.e., a solid element, is given as $\mathbf{k_0}$. $g(x_e)$ is a function, which typically scales $\mathbf{k_0}$ according to the specific realization of its design variable x_e to account for intermediate material properties, i.e., Young's modulus.

Segmenting Images into Analysis-ready Hierarchical Meshes

Tree-based image decomposition techniques typically operate on gray-scale input then output regions, which fulfill a user specified homogeneity criterion. In this paper, we limit our focus to quadtrees (2D) and octrees (3D), which follow the same underlying principles: The region is bisected parallel to the Cartesian axes, if the spread in gray-scale values of any contained pixels exceeds a user-specified threshold. In 2D this results in splitting and replacing the region by four equally sized blocks, explaining the prefix "quad". If this process is repeated often enough, visually, a tree-like structure emerges (Fig. 1), which clarifies the suffix. For pixel-based applications, bisection requires the resolution of the input image to comprise a power of two. Since this is generally not the case, images are padded to the next power of 2, where applicable. Each homogeneous block is represented by a single element. Assuming nodes in each element corner, the image decomposition is transformed into a mesh representation (Fig. 2). Due to the irregularity of the mesh, challenges arise during analysis: Constructing appropriate elements and satisfying sufficient discretization. Traditionally, balancing such meshes, i.e., enforcing at maximum at 2:1 ratio of adjacent blocks, tempers most issues. However, hanging nodes (Fig. 2, red) still persist. One option to alleviate this issue is the adoption of a polytope-based finite element variant, such as the SBFEM. Since, for example, only 16 possible element realizations exist in 2D, precomputation may even be exploited in the process.

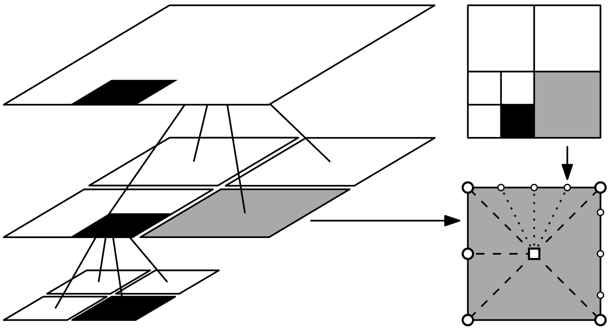

Figure 1: Example of image decomposition by quadtree algorithm with sample SBFEM polytope element in gray.

The Forward Problem for 3D Elastostatics

We consider a three dimensional domain Ω, whose boundary $\Gamma = \Gamma_0 \cup \Gamma_u \cup \Gamma_t$ comprises regions of free surface conditions (Γ_0), prescribed displacements $\bar{u}$ (Γ_u) and prescribed

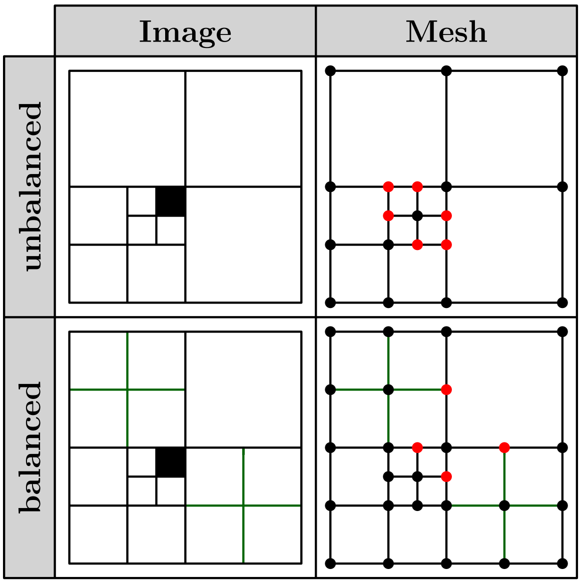

Figure 2: Segmentation of image with resulting mesh for unbalanced and balanced quadtree decompositions. Hanging nodes in red and additional elements introduced due to balancing in green.

traction $\bar{t}$ (Γ_t). The strong form with boundary conditions may be stated as:

$$\nabla \cdot \sigma + b = 0 \quad \text{in} \quad \Omega \tag{2a}$$
$$u = \bar{u} \quad \text{on} \quad \Gamma_u \tag{2b}$$
$$\sigma \cdot n = \bar{t} \quad \text{on} \quad \Gamma_t \tag{2c}$$

The Cauchy stress tensor, unit outward normal to the boundary and applied body force per unit volume are denoted by σ, n and b respectively. u is the displacement field and ∇ the linear gradient operator. Imposing small deformations and linear elastic material behaviour, the stress σ and strain ϵ fields are dependent on the modulus of elasticity E and Poisson's ratio ν:

$$\epsilon = \nabla_s u \quad \text{and} \quad \sigma = D\epsilon \tag{3}$$

where ∇_s is the symmetric gradient operator and D the 6x6 elasticity tensor.

The Scaled Boundary Finite Element Method in 3D Elastostatics

In this section, a brief summary of the scaled boundary finite element method in 3D is given. For a more elaborate derivation and detailed explanations, the readers may consult references [24, 25].

The salient features of an SBFEM analysis are illustrated on the problem domain described by the volume V depicted in Fig. 3. V comprises the volume spanned by the scaling center O and the 2D surface element, describing the boundary. One minor constraint is required: The domain must remain star-convex, i.e., the entire surface must be visible from the scaling center. The introduction of this scaling center is accompanied by a transition from a Cartesian reference system into one resembling polar coordinates. In radial direction the analytic variable ξ is introduced, while for each tangential direction, η respectively ζ represent the local coordinates on the boundary. Therefore, each surface element may be described by 2D interpolation shape functions $\mathbf{N}(\eta, \zeta)$ formulated in natural coordinates $-1 \leq \eta \leq 1$ and $-1 \leq \zeta \leq 1$. The interior of the domain is constructed by scaling the boundary (x, y, z) along the dimensionless radial coordinate $0 \leq \xi \leq 1$, which originates at the scaling center and ends on the boundary. The mapping of points employing the newly introduced scaled boundary coordinate system is therefore given as:

$$\hat{x}(\xi, \eta, \zeta) = \xi x(\eta, \zeta) = \xi \mathbf{N}(\eta, \zeta) \mathbf{x}, \tag{4a}$$

$$\hat{y}(\xi, \eta, \zeta) = \xi y(\eta, \zeta) = \xi \mathbf{N}(\eta, \zeta) \mathbf{y}, \tag{4b}$$

$$\hat{z}(\xi, \eta, \zeta) = \xi z(\eta, \zeta) = \xi \mathbf{N}(\eta, \zeta) \mathbf{z}. \tag{4c}$$

We denote the vectors of nodal coordinates of a surface element by $\mathbf{x}, \mathbf{y}, \mathbf{z}$ respectively. The set (ξ, η, ζ) is termed the scaled boundary coordinates in the three-dimensional domain.

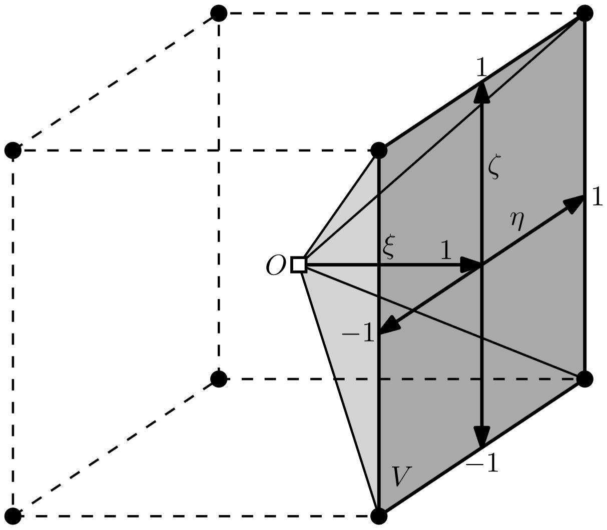

Figure 3: Three-dimensional coordinates for a scaled boundary finite element.

Similarly, the iso-parametric mapping of the displacements $\mathbf{u}(\xi, \eta, \zeta)$ at a point (ξ, η, ζ) comprises an analytic (ξ) and interpolatory (η, ζ) component:

$$\mathbf{u}(\xi, \eta, \zeta) = \left[u_x(\xi, \eta, \zeta), u_y(\xi, \eta, \zeta), u_z(\xi, \eta, \zeta)\right]^{\mathrm{T}} = \mathbf{N}^{\mathrm{u}}(\eta, \zeta) \mathbf{u}(\xi), \tag{5}$$

where $\mathbf{u}(\xi)$ represents an analytic displacement function along ξ, unique to each node on the boundary. These displacements functions are determined during the SBFEM solution. The interpolation shape function $\mathbf{N}^{\mathrm{u}}(\eta, \zeta)$ in Eq. (5) are defined analogously to the conventional FEM:

$$\mathbf{N}^{\mathrm{u}}(\eta, \zeta) = \left[N_1(\eta, \zeta) \mathbf{I}, N_2(\eta, \zeta) \mathbf{I}, \ldots, N_n(\eta, \zeta) \mathbf{I}\right], \tag{6}$$

where n denotes the amount of nodes of the surface element and $\mathbf{I}$ is the 3×3 identity matrix. Expressing the strains in scaled boundary coordinates requires splitting the linear differential operator into components $\mathbf{B}_1$ and $\mathbf{B}_1$, whose combined effect mimics the original transformation:

$$\boldsymbol{\varepsilon}(\xi, \eta, \zeta) = \mathbf{B}_1(\eta, \zeta) \mathbf{u}(\xi)_{,\xi} + \frac{1}{\xi} \mathbf{B}_2(\eta, \zeta) \mathbf{u}(\xi), \tag{7}$$

The stresses are obtained conventionally by pre-multiplying the strains by the constitute matrix:

$$\boldsymbol{\sigma}\left(\xi,\eta,\zeta\right)=\mathbf{D}\left(\mathbf{B}_1\left(\eta,\zeta\right)\mathbf{u}\left(\xi\right)_{,\xi}+\frac{1}{\xi}\mathbf{B}_2\left(\eta,\zeta\right)\mathbf{u}\left(\xi\right)\right). \tag{8}$$

The weak form for each subdomain may be derived by applying several methods [6,9,29]. Neglecting body loads and surface tractions, two equations arise (Eqs. 9 and 10):

$$\mathbf{E}_0\xi^2\mathbf{u}\left(\xi\right)_{,\xi\xi}+\left(2\mathbf{E}_0-\mathbf{E}_1+\mathbf{E}_1^{\mathrm{T}}\right)\xi\mathbf{u}\left(\xi\right)_{,\xi}+\left(\mathbf{E}_1^{\mathrm{T}}-\mathbf{E}_2\right)\mathbf{u}\left(\xi\right)=0, \tag{9}$$

The coefficient matrices $\mathbf{E}_0,\mathbf{E}_1,\mathbf{E}_2$ bare a striking similarity to conventional FEM stiffness matrices, both in their structure and in that they are calculated for each element individually, with subsequent assembly for each subdomain. The internal nodal forces modes at the boundary $\mathbf{q}\left(\xi\right)$ are derived as:

$$\mathbf{q}\left(\xi\right)=\xi\left(\mathbf{E}_0\xi\mathbf{u}\left(\xi\right)_{,\xi}+\mathbf{E}_1^{\mathrm{T}}\mathbf{u}\left(\xi\right)\right). \tag{10}$$

The scaled boundary finite element equation is solved by the matrix function solution proposed by Song [23]. Therein, the quadratic eigen-problem is recast into a system of first-order differential equations in ξ, at the expense of doubling the amount of unknowns:

$$\xi\left\{\begin{array}{c}\xi^{0.5}\mathbf{u}\left(\xi\right)\\ \xi^{-0.5}\mathbf{q}\left(\xi\right)\end{array}\right\}_{,\xi}=-\mathbf{Z}\left\{\begin{array}{c}\xi^{0.5}\mathbf{u}\left(\xi\right)\\ \xi^{-0.5}\mathbf{q}\left(\xi\right)\end{array}\right\}, \tag{11}$$

with the Hamiltonian coefficient matrix $\mathbf{Z}$ defined as:

$$\mathbf{Z}=\left[\begin{array}{cc}\mathbf{E}_0^{-1}\mathbf{E}_1^{\mathrm{T}}-0.5\mathbf{I} & -\mathbf{E}_0^{-1}\\ -\mathbf{E}_2+\mathbf{E}_1\mathbf{E}_0^{-1}\mathbf{E}_1^{\mathrm{T}} & -\left(\mathbf{E}_1\mathbf{E}_0^{-1}-0.5\mathbf{I}\right)\end{array}\right]. \tag{12}$$

The Schur decomposition with subsequent block-diagonalization is employed to avoid numerical deficiencies in the solution and decompose Z to satisfy:

$$\mathbf{ZV}=\mathbf{VS}, \tag{13}$$

where $\mathbf{S}$ and $\mathbf{V}$ are the real Schur form and the transformation matrix, respectively. In order to strip the bounded from the unbounded response, the diagonal blocks containing the eigen-values of $\mathbf{S}$ are sorted in ascending order and the columns of $\mathbf{V}$, which contain the associated eigen-modes, are reordered accordingly. The following partitioning is devised:

$$\mathbf{S}=\left[\begin{array}{cc}\mathbf{S}_{11} & 0\\ 0 & \mathbf{S}_{22}\end{array}\right], \tag{14a}$$

$$\mathbf{V}=\left[\begin{array}{cc}\mathbf{V}_{11} & \mathbf{V}_{12}\\ \mathbf{V}_{21} & \mathbf{V}_{22}\end{array}\right]. \tag{14b}$$

$\mathbf{S}_{11}$ contains all eigenvalues with negative sign, i.e., $\mathrm{Re}\left(\lambda(\mathbf{S}_{11})\right)<0$, which can be shown to correspond to the bounded domain solution. The general solutions for the displacements and internal nodal forces for the bounded domain is sought in the form of a power series and can be expressed as:

$$\mathbf{u}\left(\xi\right)=\mathbf{V}_{11}\xi^{-\mathbf{S}_{11}-0.5\mathbf{I}}\mathbf{c}, \tag{15a}$$

$$\mathbf{q}\left(\xi\right)=\mathbf{V}_{21}\xi^{-\mathbf{S}_{11}+0.5\mathbf{I}}\mathbf{c}. \tag{15b}$$

The integration constants $\mathbf{c}$, whose values are determined in post-processing, follow from the effective boundary conditions of the problem. Formulating Eq. (15) at the boundary ($\xi = 1$), the nodal displacements $\mathbf{u} = \mathbf{u}\,(\xi = 1)$ and the nodal forces $\mathbf{q} = \mathbf{q}\,(\xi = 1)$ can be expressed as:

$$\mathbf{u} = \mathbf{V}_{11}\mathbf{c}, \tag{16a}$$

$$\mathbf{q} = \mathbf{V}_{21}\mathbf{c},. \tag{16b}$$

Since the static stiffness matrix $\mathbf{K}$ is defined as $\mathbf{q} = \mathbf{K}\mathbf{u}$, $\mathbf{K}$ of a subdomain is determined by substituting Eq. (16):

$$\mathbf{K} = \mathbf{V}_{21}\mathbf{V}_{11}^{-1}. \tag{17}$$

Upon assembly of all individual subdomain stiffness matrices into a global stiffness matrix, the displacement solution is sought according to conventional FEM procedures.

Proposed Scheme

The proposed scheme primarily entails a drop-in replacement for calculating the displacement field. By exploiting hierarchical image decomposition techniques, regions with homogeneous material properties are identified and represented by a single element, such that coarser, adaptive meshes result with significantly fewer DOFs than with conventional grid approaches. Although remeshing is required for each iteration, the implemented decomposition techniques are economical and the resulting computational toll is easily recuperated by solving a forward problem with significantly fewer DOFs, even on modestly sized example domains. The steps differ slightly between 2D and 3D, since in 3D the precomputation procedure employed for 2D would result in 4096 unique element realizations, whose construction requires building a substantial library and accompanying algorithmic logic. Instead, the unique element realizations for each 3D mesh are determined, which are computed once and subsequently cloned for the remaining mesh. Empirically, only a fraction of possible element realizations exist simultaneously on a mesh.

The proposed scheme comprises the following steps:

1. **Precompute**
   For the 2D case, the 16 possible element realizations are precomputed with Young's modulus equal to one.

2. **Initialize and begin TO loop**
   Only the filter must be prepared. Conversely to conventional approaches, remeshing at each iteration is required, rendering the preparation of sparse stiffness matrix assembly vectors [1] obsolete.

3. **Calculate displacement field**
   The grid of design variables is fed as a gray-scale image to the decomposition algorithm, which outputs an analysis-ready hierarchical mesh. Color-encoded regions are automatically recognized and resolved. Their inscribed operations are then applied. Once the stiffness properties of each element are identified, the displacement field is calculated analogous to the conventional FEM.

4. **Determine compliance**
   The compliance at each iteration is calculated as the product of the system displacement field and the force vector.

5. **Determine sensitivities**
   The sensitivities of the design variables are evaluated element-wise, by iterating over each subdomain. Since the subdomains are of variable size, the calculated sensitivities must be normalized per unit volume.

6. **Filter sensitivities**
   Standard mesh-independency filtering techniques may be applied as necessary.

7. **Design variable pro-/demotion**
   Solving the optimization problem, for example by optimality criterion (OC) approach identifies, which design variables to promote or demote, i.e., assign or subtract material. While B/ESO, SERA and level-set approaches result in black-and-white outputs, SIMP-based approaches introduce intermediate material distributions. This is rectified by either employing a Heavyside projection [1] or a gray-scale filter [13, 22]. The updated design variables form the input for the subsequent iteration.

8. **Export of results**
   Upon completing the analysis, the hierarchical mesh is thresholded to yield the optimized system geometry. Simple methods permit exporting the hierarchical mesh to STL format for subsequent additive manufacturing.

In order to further accelerate the analysis procedure, which is constrained by the solution of the forward problem, so-called hard-kill variants have been proposed. These differ from standard soft-kill approaches in how they treat void elements: Soft-kill approaches assign a very small stiffness, typically 10^{-9}, which impacts the conditioning of the numerical problem, while hard-kill approaches disregard such elements entirely. Not all problems, however, are amenable to this approach, since multiple independent substructures potentially arise during analysis, leading to numerical instabilities.

Numerical Examples

Three numerical examples are examined in this paper:

1. A thick cantilever subject to a point load at mid-height.
2. An L-shaped bracket with prescribed material distributions and multiple load cases.
3. The 3D wheel.

The first example verifies the proposed method, while the second showcases the extended capabilities, by color-encoding system and analysis information directly into the input image. Having thoroughly discussed the 2D behaviour, we extend the analysis to the 3D wheel problem. For each of the numerical examples, we couple the proposed method with a different TO variant, e.g., B/ESO, SERA and SIMP. One could have equally chosen to employ a level-set based method as an alternative to the ones listed prior.

Thick Cantilever

A thick cantilever subject to a point load at mid-height (Fig. 4) is considered. The width and height are discretized by 512 and 256 pixels respectively. The prescribed volume fraction f is chosen as 0.4. The penalty exponent of both SIMP and BESO approaches is equal to 3. Following [1], a filter utilizing Matlab's built-in `conv2` function is implemented with a radius of 16 pixels. For the BESO, the evolutionary volume ratio parameter is set to 0.1. Elements arising during quadtree decompositon are limited in size to ≤ 32 pixels. For the first iteration the discretization of the conventional grid is adopted. This is necessary, since the initial homogeneous material distribution would lead to a too coarse discretization, biasing the calculated sensitivities.

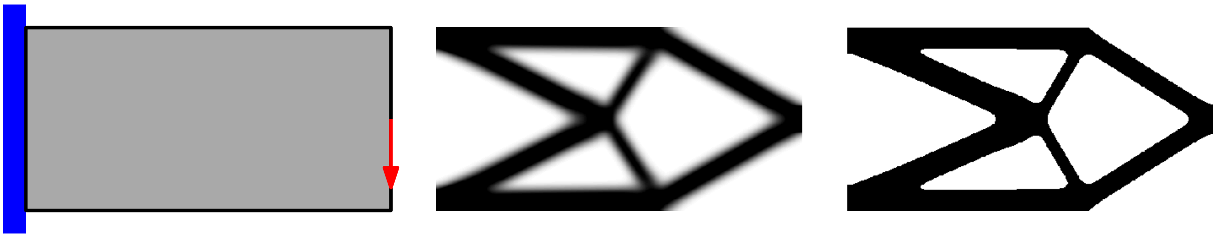

Figure 4: From left to right: Thick cantilever setup, SIMP reference solution and BESO baseline.

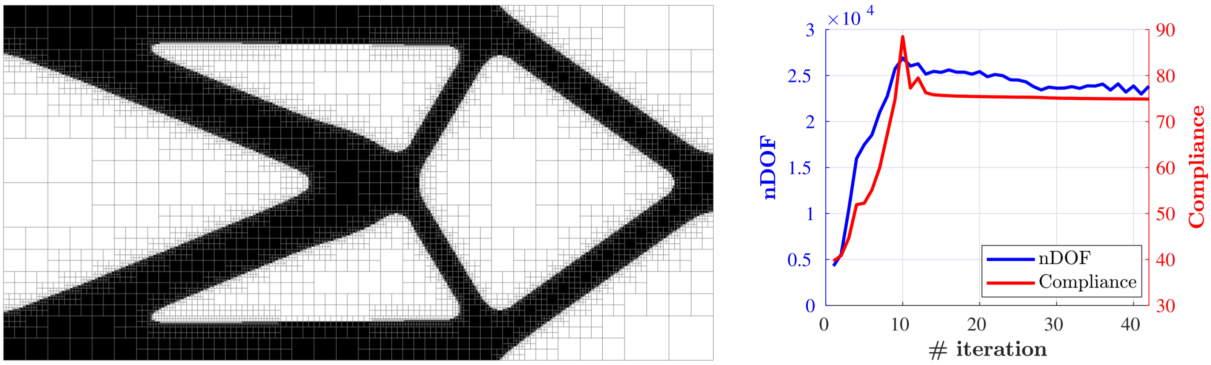

Figure 5: BESO+QT optimized topology with evolution of DOFs and compliance across iterations.

In order to verify the proposed scheme, we investigate an elementary case: TO schemes, which result in black-and-white output, i.e., do not posses noticeable transitions in material distributions, permit the adaptive meshing algorithm to develop its full potential and minimize the DOFs required for analysis. Therefore, we contrast our scheme, which we term BESO+QT, to a BESO baseline [10]. Since, BESO follows a heuristic approach to TO, and it can therefore be mislead to local minima under certain circumstances [18], we first supply a SIMP reference solution [1]. Both the reference solution and the baseline depict the same resulting topology (Fig. 4). The BESO+QT (Fig. 5) is indistinguishable from the BESO baseline. The difference in compliance (Tab. 1) resulting from the SIMP and BESO approaches stems from the presence of transition material in the SIMP, leading to a slightly more flexible structure and therefore higher compliance. Given a higher value for the penalization parameter, this difference diminishes. The discrepancy in compliance between BESO and BESO+QT is attributed to the discretization: A coarse discretization results in a stiffer structure and therefore lower compliance. This is evident, since the amount of DOFs present during analysis is reduced by more than one order of magnitude. This in turn significantly alleviates the computational burden and associated memory requirements (Fig. 5, DOFs evolution). The two additional iterations required to reach the stopping criterion, i.e., a 5% increase, represents a negligible difference to the base line case. In this contrived example, in which all three methods share a common implementation, differing only in the method of solving the forward problem, a glimpse of the computational potential of the proposed scheme is possible: For this specific numerical example, an analysis concludes almost 5x faster, when employing the proposed scheme.

Table 1: A comparison of results for SIMP reference solution, BESO baseline and proposed BESO+QT.

Method	nIt.	Compliance	DOFs	time [s]
SIMP	49	87.6	263'682	110
BESO	40	75.6	263'682	98
BESO+QT	42	74.8	23'846	20

Modified L-bracket

A modified L-bracket setup is considered (Fig. 6). In this example the SERA is employed, which remedies the drawbacks of the BESO scheme [2], while maintaining black-and-white solutions. For this analysis, only a color-encoded input image is provided. The proposed scheme automatically recognizes significant regions and their associated operations during the automated decomposition phase. The colors blue, red, green, white and gray correspond to boundary conditions, loadings, solid, void and domain pixels respectively. Input images are easily constructed by small scripts or obtained from, e.g., medical imaging applications. Especially from the user perspective, manipulating analysis parameters by color-encoding simplifies the overall process and permits direct visual verification of the input prior to analysis. Further, it facilitates testing of variants. In this example, the domain is discretized by 512 pixels in each direction. A volume fraction of 0.3 is specified. The `conv2`-type filter is employed with radius equal 12 pixels. The SERA parameters PR, SR and B are chosen as 0.03, 1.3 and 0.003 respectively. All quadtree elements are of size ≤ 128.

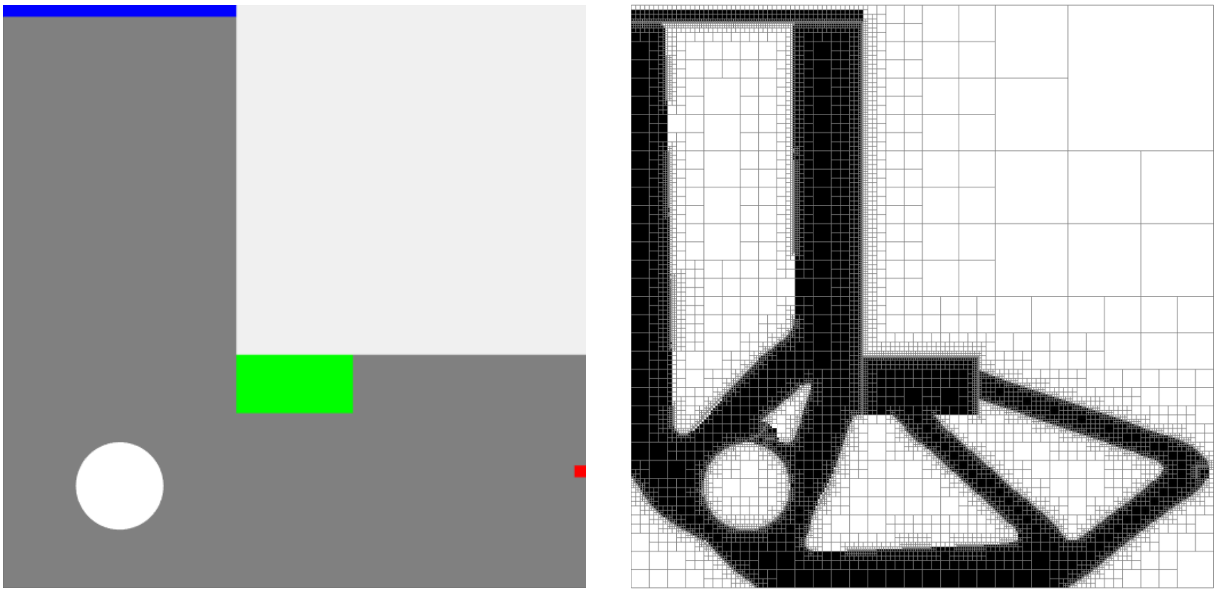

Figure 6: Topology optimization of an L-bracket via automated image-based analysis. Input image (left) and resulting topology (right).

For more involved geometries, gratuitous discretization may arise due to slight misalignment with respect to the optimal quadtree meshing strategy. This is apparent surrounding the blue and red regions (Fig. 6), where a pertubation by 1 pixel triggers excessive refinement to accurately capture the domain's geometry. To demonstrate this general case, the L-bracket is treated as is by the quadtree decomposition, which results in 32'694 DOFs, while a shifted, scaled and therefore better aligned domain geometry results in 26'554 DOFs respectively. The conventional method employing the fine grid discretization treats 526'338 DOFs. In this example, the required DOFs are reduced by over an order of magnitude. For such small examples, a typical forward analysis completes in ≤ 0.6 seconds on a modest desktop computer running in serial.

3D-Wheel

In this example we extend SBFEM-powered TO to 3D problems. The established 3D wheel problem is studied (Fig. 7): The width, depth and height are discretized by 80, 80 and 40 pixels respectively. All four corners at the bottom edge are fully restrained. A point load is applied in downward direction in the middle of the bottom surface. A volume fraction of 0.075 is sought. The penalty parameters for SIMP and gray-scale filter are chosen as 3 and 2 respectively. The filter radius is given as 3 pixels. All octree elements are ≤ 16 pixels in size. A sensitivity filter is employed. A Young's modulus of $E = 10^{-9}$ denotes material voids. The analysis concludes after 200 iterations.

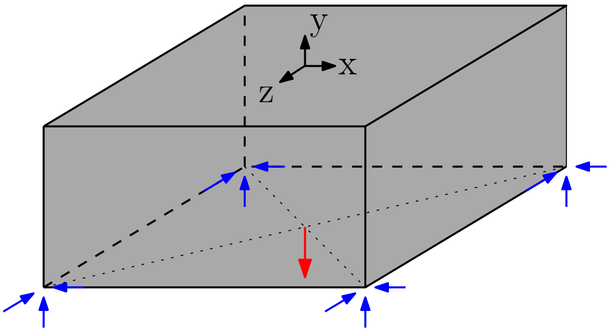

Figure 7: Problem domain of the 3D wheel benchmark.

The SIMP with gray-scale filter is utilized. The addition of the gray-scale filter is crucial to obtaining computational efficiency, since regions of transitional material distributions, which impact the proposed hierarchical meshing techniques, are minimized. The preconditioned conjugate gradients method (pcg) is employed to obtain the solution of the forward problem. Scaling is performed to combat the conditioning issues due to the treatment of void elements [28] with secondary preconditioning by incomplete Cholesky decomposition. This significantly reduced the amount of iterations required for convergence. A strict convergence tolerance of 10^{-9} was specified to contain any divergence from the reference solution to the adaptive discretization choice. In Fig. 8 the final topology is depicted along with a sample of the octree mesh.

It is demonstrated, that the amount of DOFs and number of non-zero entries in the stiffness matrix (nnz) is significantly reduced (Fig. 9) by employing the proposed scheme, especially after the initial phase in the TO process. Although up to 4096 unique element realization are possible on the octree, only a handful exist at each iteration (Fig. 9). Computational effort is alleviated by only computing the element stiffness matrices for the unique elements and cloning the remainder. However, the overwhelming computational burden still resides in the solution of the forward problem.

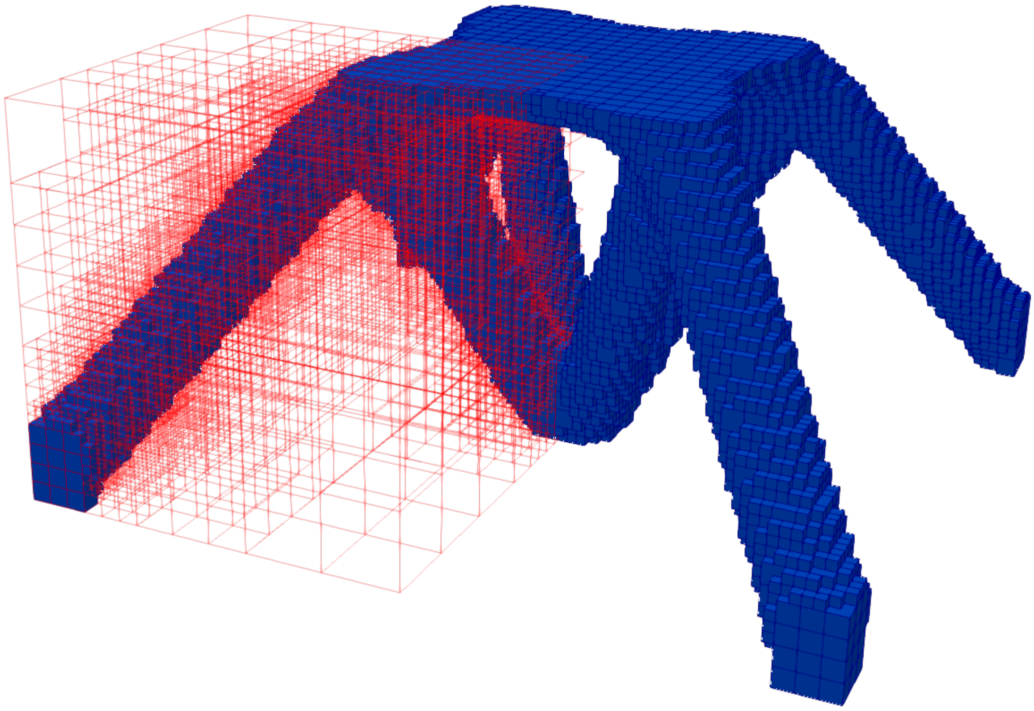

Figure 8: Resulting topology for 3D wheel benchmark in blue with sample octree mesh in red.

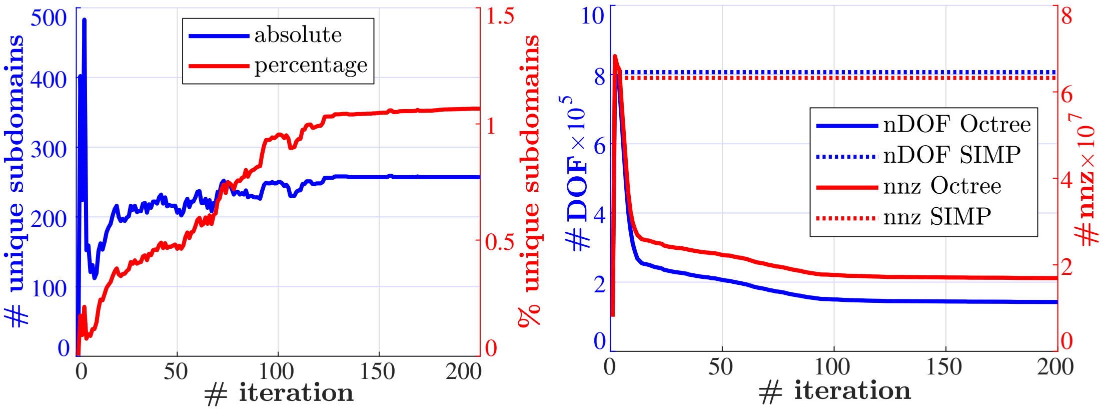

Figure 9: Evolution of hierarchical mesh quantities across iterations.

The final topology is readily exported to STL format by operating on the octree, greatly facilitating subsequent additive manufacturing procedures.

Conclusions

This paper advances the current state-of-the-art in TO by employing hierarchical meshes coupled with polytope-based numerical methods, which alleviate issues commonly associated with hanging nodes. Enforcing a balancing operation on the mesh further limits the amount of feasible element realizations, which is exploited through precomputation of element stiffness matrices and cloning of element properties. Moreover, we propose a novel and intuitive scheme for interacting with the analysis parameters by color-encoding input images, which the decomposition techniques automatically translate into analysis-ready meshes. Further, we extend SBFEM-powered TO to 3D problems and demonstrate an effective means for exporting results to STL format for subsequent additive manufacturing (3D printing).

TO relying on hierarchical meshes is shown to require only a fraction of the DOFs demanded by traditional grid approaches, significantly reducing the computational toll. The three investigated numerical examples consistently demonstrate a remarkable reduction in required DOFs and memory requirements, without incurring a perceivable loss of accuracy.

Considering the generality of this approach in handling color-coded input, an extension to multi-material TO presents an intriguing direction for future work.

Acknowledgements

This research was performed under the auspices of the Swiss National Science Foundation (SNSF), Grant # 200021_153379, A Multiscale Hysteretic XFEM Scheme for the Analysis of Composite Structures.

References

[1] Andreassen, E., Clausen, A., Schevenels, M., Lazarov, B. S., and Sigmund, O. Efficient topology optimization in MATLAB using 88 lines of code. *Structural and Multidisciplinary Optimization 43*, 1 (Jan. 2011), 1–16.

[2] Ansola Loyola, R., Querin, O. M., Garaigordobil Jimnez, A., and Alonso Gordoa, C. A sequential element rejection and admission (SERA) topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization 58*, 3 (Sept. 2018), 1297–1310.

[3] Beiro Da Veiga, L., Brezzi, F., Cangiani, A., Manzini, G., Marini, L. D., and Russo, A. BASIC PRINCIPLES OF VIRTUAL ELEMENT METHODS. *Mathematical Models and Methods in Applied Sciences 23*, 01 (Jan. 2013), 199–214.

[4] Bendse, M. P., and Sigmund, O. *Topology Optimization.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[5] de Berg, M., Cheong, O., van Kreveld, M., and Overmars, M. *Computational Geometry.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[6] Deeks, A. J., and Wolf, J. P. A virtual work derivation of the scaled boundary finite-element method for elastostatics. *Computational Mechanics 28*, 6 (jun 2002), 489–504.

[7] Gupta, A. K. A finite element for transition from a fine to a coarse grid. *International Journal for Numerical Methods in Engineering 12*, 1 (1978), 35–45.

[8] Hoshina, T. Y. S., Menezes, I. F. M., and Pereira, A. A simple adaptive mesh refinement scheme for topology optimization using polygonal meshes. *Journal of the Brazilian Society of Mechanical Sciences and Engineering 40*, 7 (July 2018), 348.

[9] Hu, Z., Lin, G., Wang, Y., and Liu, J. A Hamiltonian-based derivation of Scaled Boundary Finite Element Method for elasticity problems. *IOP Conference Series: Materials Science and Engineering 10* (jun 2010), 012213.

[10] Huang, X., and Xie, Y. M. *Evolutionary Topology Optimization of Continuum Structures: Methods and Applications.* John Wiley & Sons, Ltd, Chichester, UK, Apr. 2010.

[11] Lan, L. T. T., and Boucher, A. Simplified quadtree image segmentation for image annotation.

[12] Li, C., and Tong, L. Topology optimization of incompressible materials based on the mixed SBFEM. *Computers & Structures 165* (Mar. 2016), 24–33.

[13] Liu, K., and Tovar, A. An efficient 3d topology optimization code written in Matlab. *Structural and Multidisciplinary Optimization 50*, 6 (Dec. 2014), 1175–1196.

[14] Liu, Y. J., Mukherjee, S., Nishimura, N., Schanz, M., Ye, W., Sutradhar, A., Pan, E., Dumont, N. A., Frangi, A., and Saez, A. Recent Advances and Emerging Applications of the Boundary Element Method. *Applied Mechanics Reviews 64*, 3 (Mar. 2012), 030802.

[15] Man, H., Song, C., Natarajan, S., Ooi, E. T., and Birk, C. Towards Automatic Stress Analysis using Scaled Boundary Finite Element Method with Quadtree Mesh of High-order Elements. *arXiv:1402.5186 [math]* (Feb. 2014). arXiv: 1402.5186.

[16] Otomori, M., Yamada, T., Izui, K., and Nishiwaki, S. Matlab code for a level set-based topology optimization method using a reaction diffusion equation. *Structural and Multidisciplinary Optimization 51*, 5 (May 2015), 1159–1172.

[17] Panesar, A., Brackett, D., Ashcroft, I., Wildman, R., and Hague, R. Hierarchical remeshing strategies with mesh mapping for topology optimisation: Hierarchical remeshing strategies with mesh mapping for topology optimisation. *International Journal for Numerical Methods in Engineering 111*, 7 (Aug. 2017), 676–700.

[18] Rozvany, G. I. N. A critical review of established methods of structural topology optimization. *Structural and Multidisciplinary Optimization 37*, 3 (Jan. 2009), 217–237.

[19] Salazar de Troya, M. A., and Tortorelli, D. A. Adaptive mesh refinement in stress-constrained topology optimization. *Structural and Multidisciplinary Optimization 58*, 6 (Dec. 2018), 2369–2386.

[20] Sanders, E. D., Pereira, A., Aguil, M. A., and Paulino, G. H. PolyMat: an efficient Matlab code for multi-material topology optimization. *Structural and Multidisciplinary Optimization 58*, 6 (Dec. 2018), 2727–2759.

[21] Saputra, A., Talebi, H., Tran, D., Birk, C., and Song, C. Automatic image-based stress analysis by the scaled boundary finite element method: AUTOMATIC IMAGE-BASED STRESS ANALYSIS BY THE SCALED BOUNDARY FEM. *International Journal for Numerical Methods in Engineering 109*, 5 (feb 2017), 697–738.

[22] Sigmund, O. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization 33*, 4-5 (Feb. 2007), 401–424.

[23] Song, C. A matrix function solution for the scaled boundary finite-element equation in statics. *Computer Methods in Applied Mechanics and Engineering 193*, 23-26 (jun 2004), 2325–2356.

[24] Song, C. *The scaled boundary finite element method: intorduction to theory and implementation.* John Wiley & Sons, Hoboken, New Jersey, 2018.

[25] Song, C., and Wolf, J. P. The scaled boundary finite-element methodalias consistent infinitesimal finite-element cell methodfor elastodynamics. *Computer Methods in Applied Mechanics and Engineering 147*, 3-4 (aug 1997), 329–355.

[26] Tabarraei, A., and Sukumar, N. Adaptive computations on conforming quadtree meshes. *Finite Elements in Analysis and Design 41*, 7-8 (Apr. 2005), 686–702.

[27] Wang, S., de Sturler, E., and Paulino, G. H. Dynamic Adaptive Mesh Refinement for Topology Optimization. *arXiv:1009.4975 [cs, math]* (Sept. 2010). arXiv: 1009.4975.

[28] Wang, S., Sturler, E. d., and Paulino, G. H. Large-scale topology optimization using preconditioned Krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering 69*, 12 (Mar. 2007), 2441–2468.

[29] Wolf, J. P. *The scaled boundary finite element method.* J. Wiley, Chichester, West Sussex, England ; Hoboken, NJ, USA, 2003.