© 2019 by Miguel A. Salazar de Troya. All rights reserved.

ADAPTIVE MESH REFINEMENT IN TOPOLOGY OPTIMIZATION

BY

MIGUEL A. SALAZAR DE TROYA

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Mechanical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Doctoral Committee:

Professor Daniel A. Tortorelli, Chair and Director of Research
Professor Phillipe Geubelle
Professor Arif Masud
Assistant Professor Kai James

Abstract

This dissertation presents developments in stress constrained topology optimization with Adaptive Mesh Refinement (AMR).

Regions with stress concentrations dominate the optimized design. As such, we first present an approach to obtain designs with accurately computed stress fields within the context of topology optimization. To achieve this goal, we invoke threshold and AMR operations during the optimization. We do so in an optimal fashion, by applying AMR techniques that use error indicators to refine and coarsen the mesh as needed. In this way, we obtain accurate simulations and greater resolution of the design domain in a computationally efficient manner. We present results in two dimensions to demonstrate the efficacy of our method.

The topology optimization community has regularly employed optimization algorithms from the operations research community. However, these algorithms are implemented in the Euclidean space instead of the proper function space where the design, i.e. volume fraction, field resides. In this thesis, we show that, when discretizing the volume fraction field over a non-uniform mesh, algorithms in Euclidean space are mesh dependent. We do so by first explaining the functional analysis tools necessary to understand why convergence is affected by the mesh. Namely, the distinction between derivative and gradient definitions and the role of the mesh dependent inner product. These tools are subsequently used to make the Globally Convergent Method of Moving Asymptotes (GCMMA), a popular optimization algorithm in the topology optimization community, mesh independent. We then benchmark our algorithm with three common problems in topology optimization.

High resolution three-dimensional design models optimized for arbitrary cost and constraint functions are absolutely necessary ingredients for the solution of real–world engineering design problems. However, such requirements are non trivial to implement. In this thesis, we address this dilemma by developing a large scale topology optimization framework with AMR. We discuss the need for efficient parallelizable regularization methods that work across different mesh resolutions, iterative solvers and data structures. Furthermore, the optimization algorithm needs to be implemented with the same data structure that is used for the design field. To demonstrate the versatility of our framework, we optimize the designs of a three dimensional stress

constrained benchmark L-bracket and a stress-constrained compliant mechanism.

To Father and Mother.

Acknowledgments

I would like to thank my advisor Professor Daniel Tortorelli for his guidance and support. I thank him for giving me the freedom to work on the research topics that I enjoyed the most. Without his support, I would have never been able to work at the Lawrence Livermore National Laboratory (LLNL) to finish my thesis.

Likewise, I want to thanks the Livermore Graduate Scholarship Program to financially support this work and give me the freedom to collaborate with other projects within the LLNL. Special thanks go to the topology optimization team at LLNL: Geoffrey Oxberry, Cosmin Petra, Daniel White, Jun Kudo, Mark Stowell, Ryan Fellini, Andrew Barker, Bill Arrighi, Seth Watts and Boyan Lazarov who adopted me as one more of their group and provided invaluable feedback for my work. I would also like to thank my mentors at the LLNL, Kyle Sullivan and Todd Weisgrabber, who helped me with the paperwork to pursue my research. I cannot forget to express my gratitude to all the personnel at the LLNL with whom I interacted. Specially the IT support for the computer cluster, who guided me through the process to execute the large scale simulations needed in this thesis.

During my years in Urbana-Champaign, my office mates Felipe Fernández, Kazeem Alidoost and Stephanie Ott-Monsivais created a stimulating and enjoyable work enviroment which I am grateful for. Likewise, my roommates Carlos, Benjamin and Andrés were always there at the end of each day to share great conversations and laughs to help me wind down.

I would like to thank the libMesh team, Roy Stogner, Paul Bauman, John Peterson for answering my questions promptly and guiding me towards a better implementation design.

Lastly, I would like to thank my family, José Antonio and Isabel, and my sister Cristina for being always close to me despite the distance.

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Topology optimization is a growing field that has found applications in many arenas, from the aerospace and automotive industries to microfluidics. It is a design tool that finds the optimal distribution of material in a design domain to minimize a cost function subject to a set of constraint inequalities. It is a relatively new approach to design which obviates the need for trial-and-error design and lessens the need for engineering intuition. Notably, it can be applied to design systems where intuition is lacking.

The first applications of topology optimization were in solid mechanics where the goal was to minimize a structure's compliance subject to a constraint on its mass. In real world applications, however, structural components need to satisfy failure constraints related to maximum stress so as to prevent collapse. To satisfy these constraints, a redesign step of the compliance optimized design is performed, thereby reverting to the trial-and-error approach we want to avoid. Including these constraints directly in the optimization would render these redesign steps unnecessary. However, before this can be done, the accuracy of the computed stress field in the topology optimization must be increased; this thesis addresses this topic.

In topology optimization, the design geometry can be represented with volume fraction or level set methods. The volume fraction method discretizes the design domain into elements and assigns a volume fraction variable which ranges between 0 and 1 to each element. Penalization is used to enforce discrete values, i.e. either 0 or 1. Although this penalization method can nucleate holes, it is highly dependent of the problem and not always applicable. On the other hand, level set methods represent the design boundary implicitly using the isocontour of the scalar-valued level-set function. The design is discrete from the start hence penalization methods are unnecessary. However, the ability to change the topology is limited as new holes in the design are not created without external tools such as the topological derivative. In this thesis, we focus on the former method as it is well established in our field of application, i.e., linear elasticity.

The topology optimization problem is ill-posed, as the "optimal" designs consist of a non converging sequence of structures with highly oscillatory material-void regions. By far the most popular way to formulate a well-posed problem is by restriction whereby the oscillations are supressed by applying a filter operation to the volume fractions. This filter, however, blurs the boundary and renders the stress field inaccurate. To obtain a sharp boundary, a threshold function over the volume fractions is commonly employed, however, the stress field inaccuracy is still not resolved.

Adaptive Mesh Refinement (AMR) refines (coarsens) the mesh in the domain regions with the highest (lowest) error of the approximated response is. This strategy allows us to save computational resources while increasing accuracy. It is ideally suited for topology optimization in which regions devoid of material are meaningless and need not be accurately modeled. The AMR would coarsen these regions. By the same token, regions filled with material require a higher mesh resolution for more accurate geometry definition and stress prediction.

In Chapter 2, we combine an AMR strategy with a thresholding function to obtain sharp boundaries and accurately computed stress fields. First, we perform a validation study to demonstrate the accuracy of the computed stress field by our AMR approach that is based on an energy error indicator. Next, we replace the energy error indicator with a goal oriented error indicator that is more suitable for our computed optimization cost and constraint functions. In this way, we attain suitably accurate cost and constraint function values, which ensures the design prototypes will behave as expected.

Chapter 3 explains the mesh dependency in the optimized designs that can result when viewing the design volume fraction field as an element of an Euclidean space rather than its proper function space. Problems arise when this approach is applied over nonuniform meshes, such as those generated in AMR. To address this issue, we first present an in-depth mathematical formulation of optimization algorithms in function spaces. Then, we lay out the necessary concepts to formulate optimization algorithms independent of the mesh configuration and apply them to one of the most popular algorithms in topology optimization, the Globally Convergent Method of Moving Asymptotes (GCMMA).

Chapter 4 leverages the concepts introduced in Chapters 2 and 3 to extend the range of applications to high resolution three dimensional designs. We also mention additional issues that need to be addressed to solve such large scale problems. Namely, the use of an efficient technique to filter the volume fractions across several mesh refinement levels and efficient preconditioners for the iterative linear solvers for multiple systems of equations.

Recommendations for future work are provided in Chapter 5.

Chapter 2

Adaptive mesh refinement in stress-constrained topology optimization

2.1 Introduction

Topology optimization is a well established design tool that has found industrial applications in recent years. However, most developments focus on the "compliance problem", i.e. to minimize compliance subject to a mass constraint. This is in spite of the fact that in many cases, it is necessary to satisfy failure constraints such as on maximum yield stress. In this work, we investigate stress constrained topology optimization.

There are numerous challenges to solve stress-constrained problems. First, the optimal stress constrained solutions belong to degenerate lower dimension subspaces of the design domain. Gradient-based optimization algorithms cannot reach these optima; rather they get trapped in locally optimal solutions. This phenomenon, first studied in the optimal design of trusses [27, 52, 92] is known in the literature as the "singularity problem". It is resolved by relaxing the stress constraints, thereby regularizing the degenerate subspace [25, 77] . This is accomplished via $\varepsilon - relaxation$ [26] or $qp - relaxation$ [19] and the relaxed stress indicator [56]. In this work, we employ the last.

Second, the local stress constraints in the continuum setting lead to one constraint per finite element after discretization. This creates a scenario where the number of design variables is as large as the number of constraints, which is a computationally challenging problem. To overcome this, the constraints are agglomerated into a single measure that approximates the maximum stress value over the domain. These approximations include the p-norm and the Kreisselmeier-Steinhauser (KS) functions [73]. Alternatively, in [10] a ramp function is used to penalize all regions with stress constraint violations. We adopt this approach.

Third, the inaccuracy of the computed stress field leads to designs that do not perform in service as their simulations suggest. This is because topology optimization does not use a conforming mesh; rather, it projects the design onto a fixed grid akin to a fictitious domain finite element method. Although this approach is convenient for the optimization, it yields poor accuracy in the computed response. This issue is exasperated in density-based topology optimization because the domain geometry is not explicitly defined; rather it is defined by an interphase of "partially filled" finite elements which renders useless stress computations. In this work, we use Adaptive Mesh Refinement (AMR) and thresholding to obtain accurate stress values. Our goals are to obtain comparable optimal designs to those that are obtained on highly refined uniform meshes, compute accurate stress fields and reduce the computational cost. To achieve these goals we need a reliable and efficient error indicator to drive the AMR and a topology optimization strategy that accommodates the evolving discretization.

The first application of AMR in topology optimization appears in [63]; it uses separate design and analysis meshes which are related by a smoothing algorithm. The AMR is only performed on the analysis mesh; it is based on geometric criteria and no coarsening is performed. Coarsening is considered in [100] which uses a mesh hierarchy to solve the compliance problem. Again the refinement and coarsening criteria are solely based on the geometry and not on the error in the computed response. A similar approach proposed in [67] uses the distance function with respect to the domain boundary as the refinement and coarsening indicator. A study of the effects of various AMR parameters on the optimization performance appears in [71]. As in [63], they use different design and analysis meshes and as in [67], the AMR uses a geometric error indicator that is based on the proximity to the domain boundary. In [68], the AMR uses a tree data structure with polygonal elements; again the error indicator is solely based on geometrical considerations. Following this trend, [85] uses the phase-field method and refines regions adjacent to the design boundary.

Independently discretizing the design and analysis meshes, [102] refines each with their own error indicators to solve the compliance problem. Unfortunately, this approach is not scalable because it is necessary to perform a search over the design discretization for each finite element in the analysis mesh. On a positive note, a posteriori error estimators are used to obtain accurate displacement and stress fields. In [20] two different error estimators are invoked to control error in the geometry and compliance. However, their method is only applied to compliance problems. An energy based error indicator is used in [97] to refine the mesh in the compliance problem. In the context of stress-constrained problems, [29] utilizes h-adaptive mesh refinement by combining mesh quality and displacement based error indicators. However, they do not validate the accuracy of their stress fields. [49] solves a stress constrained compliance minimization problem using an anisotropic mesh adaptation scheme based on the metric tensors of the spatial Hessian of the cost and constraint functions. Using topological derivatives [10] considers local stress constraints and develops an AMR method that is based on residual error indicators of the energy norm of the displacement.

Finite element analysis in structural topology optimization is a delicate issue, especially in regard to the calculation of the stress fields. Indeed, stress fields are not accurately resolved with the density method due to the blurred boundary region. Because of this, topology optimized designs are post processed wherein the blurred boundary is thresholded to define an “exact” boundary, a conforming mesh is created from this boundary and shape optimization is used to obtain designs with desirable stress distributions. To eliminate the shape optimization task, [91] embeds stress field post-processing into the topology optimization wherein the interior stress values are extrapolated to the boundary region to compute more accurate stress fields. Another option to obtain accurate stress fields is to define an explicit domain boundary with the level set method and use either a conforming mesh [8, 59, 103] or eXtended Finite Element Method (XFEM) and similar approaches [41, 42, 83, 108] to capture the boundary. However, level-set methods cannot systematically nucleate holes and their designs are highly dependent on the initial design, which limits their effectiveness in comparison to density methods. It is possible to nucleate holes via topological derivatives; however, this approach is only practical for linear problems. It is also well known that level set methods suffer from poor convergence rates and numerical oscillations in the boundary [37]. They are also difficult to implement. However, for stress-constrained problems, they are immune to the singularity issue because there is virtually no blurred boundary region.

Our contribution is a topology optimization framework to obtain designs that satisfy pointwise stress constraints with a high degree of accuracy. We are able to achieve this goal by combining AMR and a thresholding function to obtain a sharp interphase that models the material boundary.

We implement our topology optimization framework on top of the parallel finite element library libMesh [51], which allows us to address large scale problems. This framework accomodates arbitrary cost and constraint functions and can be extended to other physics than elasticity. The high performance capability will be essential when solving problems in three dimensions, which is the scope of our future work.

2.2 Adaptive mesh refinement in stress-constrained topology optimization

Topology optimization finds the distribution of material within a hold-all domain D that minimizes a cost function and satisfies constraints. Its mathematical statement is as follows

$$\min_{\chi \in \{0,1\}} \theta_0(\chi) = \int_D \pi(\chi, \mathbf{u}) \; dV \,, \tag{2.1}$$

$$\text{s.t. } \mathbf{u} \in V \text{satisfies } a(\chi; \mathbf{u}, \mathbf{v}) = L(\mathbf{v}) \text{ for all } \mathbf{v} \in V \,, \tag{2.2}$$

$$\theta_i(\chi) = \int_D g_i(\chi, \mathbf{u}) dV \leq 0 \;\; i = 1, 2 ... \mathrm{n}_i \,, \tag{2.3}$$

where

$$a(\chi; \mathbf{u}, \mathbf{v}) = \int_D \chi \mathbb{C}[\nabla \mathbf{u}] \cdot \nabla \mathbf{v} \; dV \tag{2.4}$$

and

$$L(\mathbf{v}) = \int_{\Gamma_N} \mathbf{t} \cdot \mathbf{v} \; da \, . \tag{2.5}$$

In the above, χ is the material indicator function defined such that $\chi(\mathbf{x}) = 1$ and $\chi(\mathbf{x}) = 0$ represents the presence or lack of material at $\mathbf{x} \in D$, θ_0 is the cost function to be minimized, e.g. compliance, mean displacement, maximum stress, etc. and θ_i are the n_i inequality constraint functions that typically limit the total mass, maximum stress, etc. In addition, the design needs to satisfy the linear elasticity equation, given in its weak form in Equation (2.2) where $\mathbf{u}$ is the displacement field and $V = \left\{ \mathbf{v} \in H^1(D)^d \mid \mathbf{v} = 0 \text{ on } \Gamma_D \right\}$ is the function space of admissible displacements, where Γ_D is the boundary of D over which Dirichlet boundary conditions $\mathbf{u} = 0$ are applied. Γ_N is the complementary boundary of ∂D over which the non-homogeneous traction $\mathbf{t}$ is applied; both Γ_D and Γ_N are fixed surfaces. Finally, $\mathbb{C}$ is the symmetric elasticity tensor.

Our discrete optimization problem is not amenable to gradient-based algorithms. For this reason, we 1) convexify the design domain by replacing the binary-valued design field $\chi \in \{0, 1\}$ with the volume fraction field ν, which continuously varies from 0 to 1, i.e. $\nu \in [\epsilon_\nu, 1]$, and 2) penalize intermediate values so that ν best mimics the characteristic function χ. This ensures the elasticity problem is well-posed. In this work, we penalize the elasticity tensor by replacing χ in Equation (2.4) with the RAMP function [86]

$$r(\nu) = \frac{\nu}{1 + q(1 - \nu)} \, ,$$

where q is the penalization parameter. Figure 2.1 plots the RAMP function for several values of q. We observe better behavior using the RAMP rather than the traditional SIMP method [16]. We use the usual ersatz material to model regions with no material, i.e., we replace regions where $\chi(\mathbf{x}) = 0$ with $\nu(\mathbf{x}) = \epsilon_\nu$.

Unfortunately, the topology optimization problem is ill-posed. Compliance designs consist of a non converging sequence of structures with highly oscillatory material-void regions. There are two approaches to obtain a well-posed problem, relaxation and restriction. We use the restriction approach, wherein the design space is reduced by imposing a length scale constraint on the design's geometric features. This is accomplished by imposing a constraint on the perimeter [43], the slope of the volume fraction field [74] or as in this work, by filtering the volume fraction field [17]. In this work, we use the cone filter presented in [21].

$$\hat{\nu}(\mathbf{x}, \nu) = \int_{\mathcal{B}_\epsilon(\mathbf{x})} K(\mathbf{x} - \mathbf{y}) \nu(\mathbf{y}) dV \, , \tag{2.6}$$

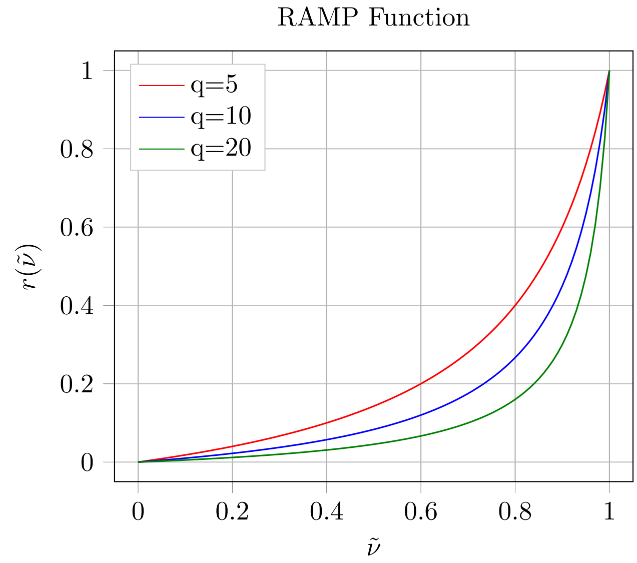


Figure 2.1: RAMP interpolation scheme.

where $\mathcal{B}_\epsilon(\mathbf{x})$ is a ball of radius ϵ centered at $\mathbf{x}$ and $K(\mathbf{x}-\mathbf{y})$ is the cone kernel function, given by

$$K(\mathbf{p}) = \frac{1}{\epsilon}(\epsilon - \|\mathbf{p}\|) \text{ if } \mathbf{p} \in \mathcal{B}_\epsilon(\mathbf{x})\,. \tag{2.7}$$

The filter radius ϵ is a parameter which defines the length scale such that the geometric complexity increases as $\epsilon \to 0$. It is important to consider the integral in Equation (2.6) instead of the weighted average of neighboring elements, as commonly seen in the literature, because the elements have different volume/areas when applying AMR. We considered using the PDE-based filter [55], but its solution with traditional lagrange elements requires a sufficiently refined mesh to avoid filtered volume fraction values above 1 and below 0. This is not possible at early stages of our AMR strategy. We will investigate in the future alternative methods that avoid this oscillation.

The filter operation brings an additional issue: the presence of blurred boundaries in the material interface. To lessen this effect, we use the threshold function of [98], cf. Equation (2.8),

$$\tilde{\nu}(\hat{\nu}) = \frac{\tanh(0.5\beta) + \tanh(\beta(\hat{\nu} - 0.5))}{\tanh(0.5\beta) + \tanh(\beta(1.0 - 0.5))}\,, \tag{2.8}$$

where β is a parameter defined such that

$$\lim_{\beta\to\infty} \tilde{\nu}(\hat{\nu}) = H(x - 0.5)\,, \tag{2.9}$$

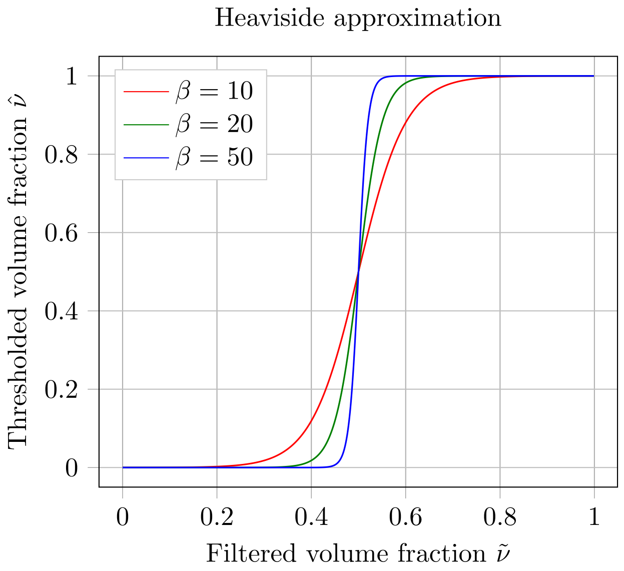


Figure 2.2: Threshold function.

where H is the unit step function. It is known that the threshold operation does not prevent the appearance of localized artifacts such as one-node-connected hinges in compliant problems, which can be alleaviated with a robust formulation [98] or with a stress constraint [31]. Therefore, we do not worry about features below the minimum length scale. The thresholding function makes the sensitivity zero in regions away from the $\tilde{\nu} = 0.5$ level set boundary, which makes it difficult to nucleate holes. For this reason, we use a continuation approach wherein β is selectively increased during the optimization.

Summarizing, we replace the binary material indicator field χ with the continuous volume fraction field ν to convexify the design space and make the optimization problem amenable to NLP. We compute the filtered volume fraction field $\hat{\nu}$ to impose a length scale constraint and hence obtain a well-posed optimization problem. Finally we compute the thresholded volume fraction field $\tilde{\nu}$ such that it mimics the originally sought material indicator function χ. Ultimately we solve the topology optimization problem of finding ν such that

$$\min_{\nu \in [\epsilon_\nu, 1]} \theta_0(\nu) = \int_D \pi(\tilde{\nu}, \mathbf{u}) \; dV \,, \tag{2.10}$$

$$\text{s.t. } \mathbf{u} \in V \text{satisfies } a(\nu; \mathbf{u}, \mathbf{v}) = L(\mathbf{v}) \text{ for all } \mathbf{v} \in V \,, \tag{2.11}$$

$$\theta_i(\nu) = \int_D g_i(\tilde{\nu}, \mathbf{u}) dV \leq 0 \;\, i = 1, 2..\mathrm{n}_i \,, \tag{2.12}$$

where

$$a(\nu; \mathbf{u}, \mathbf{v}) = \int_D r(\tilde{\nu}) \mathbb{C}[\nabla \mathbf{u}] \cdot \nabla \mathbf{v} \; dV \,. \tag{2.13}$$

2.3 Stress field accuracy in the density method

Convexifying the design space has the effect of blurring the domain boundaries, which creates inaccurate stress field computations, as demonstrated by [91]. In their work, sharper boundaries are obtained via a non-linear filter rather than the cone filter of Equation (2.6). Because of the mesh resolution, however, the boundary is jagged which gives rise to artificially high stress values. To resolve this jaggedness, the stress fields are post-processed by extrapolating the interior values onto the boundary.

In this work, we demonstrate that stress post-processing is not necessary if the mesh is adaptively refined leaving only a fine interphase boundary region. To validate our method, we perform the dog-bone study of [91] cf. Figure 2.3 in which $D = 1\ m$ and $r = 1/3\ m$. The material Young's modulus is $E = 10\ Pa$ and the Poisson ratio, $\nu = 0.3$. The volume fraction field that defines the plane stress structure is discretized as piecewise constant over the elements. The geometry is directly interpolated onto the mesh, i.e. only the elements whose centroids $\mathbf{x}_i$ are within the structure boundary take a $\nu(\mathbf{x}_i) = 1$ volume fraction. For all other void elements we assign $\nu(\mathbf{x}_i) = \epsilon_\nu = 10^{-5}$. We then filter the volume fraction with Equation (2.6) and $\epsilon = 0.0625\ m$ and thresholded with Equation (2.8) using $\beta = 100$ to leave a fine interphase region with intermediate volume fraction values. We apply uniaxial loading of 10 Pa significantly distant from the notches to avoid end effects in the notch region. Our calculated minimum principal stress is compared to those from empirical formulas [106]. This computation is repeated for several dog-bone orientations within the fixed mesh. In Figure 2.5, we plot the minimum principal stress across section A-A that passes through the center of the notch where it is seen that stress converges to the same distribution regardless of the orientation. The minimum principal stress value is -39.8, which is similar to the tabulated value -39.02 in [106].

To obtain these results, we use the residual based error indicator of Equation (2.35) and the Dörfler marking strategy [33] which is proven to decrease the error in elliptic problems. Unlike the marking strategy that we will employ in the optimization, the Dörfler strategy does not allow for coarsening. In our Dörfler implementation we select a subset $\mathcal{M}$ of elements with minimum cardinality such that

$$\eta_{\mathcal{M}} = \left(\sum_{K \in \mathcal{M}} \eta_K^2 \right)^{1/2} \geq \theta \eta \quad \theta \in [0, 1]\,, \tag{2.14}$$

where

$$\eta = \left(\sum_{K \in \mathcal{T}_h} \eta_K^2 \right)^{1/2} \tag{2.15}$$

and η_K is the element error of Equation (2.35) and $\theta = 0.3$. The mesh is iteratively refined until the total energy error η of Equation (2.38) is below 0.5. Figure 2.4 shows the refined mesh for an orientation of 0.4.

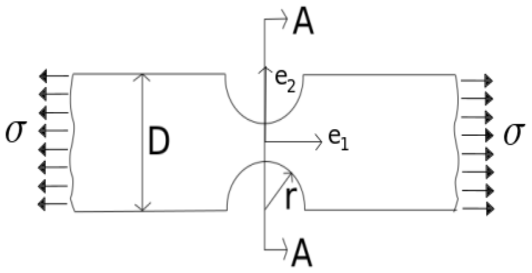


Figure 2.3: Dog-bone structure from [106].

2.4 Stress constrained topology optimization

Now we are in a position to formulate our topology optimization problem. We intend to minimize the volume of the structure subject to a constraint on the von Mises stress

$$\sigma_{VM} = \sqrt{\frac{3}{2}\boldsymbol{\sigma}^{\boldsymbol{dev}} \cdot \boldsymbol{\sigma}^{\boldsymbol{dev}}}\,, \tag{2.16}$$

where

$$\boldsymbol{\sigma}^{\boldsymbol{dev}} = \boldsymbol{\sigma} - \frac{1}{3}(\mathrm{tr}\boldsymbol{\sigma})\mathbf{I} \tag{2.17}$$

is the deviatoric stress. As such the topology optimization problem now reads

$$\min_{\nu\in[\epsilon_\nu,1]} \theta_0(\nu) = \int_D \tilde{\nu}\ dV\,, \tag{2.18}$$

$$\text{s.t. } \mathbf{u} \in V\text{satisfies } a(\nu;\mathbf{u},\mathbf{v}) = L(\mathbf{v}) \text{ for all } \mathbf{v} \in V\,, \tag{2.19}$$

$$\text{and } \theta_1(\nu) = \sigma_{VM} \leq \sigma_y \ \text{ for all } \mathbf{x} \in D\,, \tag{2.20}$$

where σ_y is the maximum allowed stress.

A consistent numerical approach would use the same elasticity tensor for the displacement solution and the stress evaluation. However, [34] showed that the stress field in a porous micro structure tends to a non-zero value even when the volume fraction tends to zero. To be physically accurate, the stress field in regions with intermediate volume fraction $\nu \in [\epsilon_\nu, 1]$ should therefore tend to non-zero values as ν tends

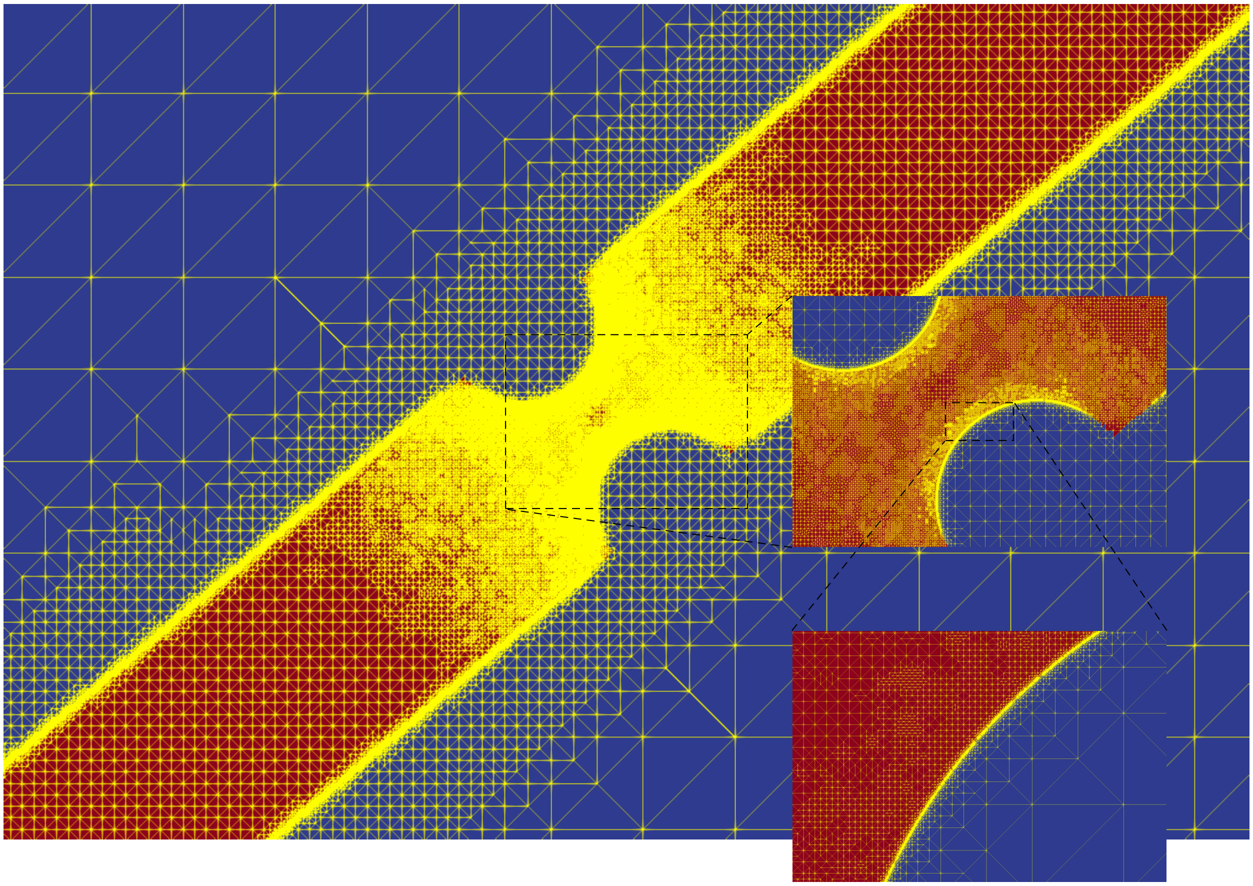


Figure 2.4: Dog-bone structure and the mesh used in the analysis for an orientation of 0.4 radians.

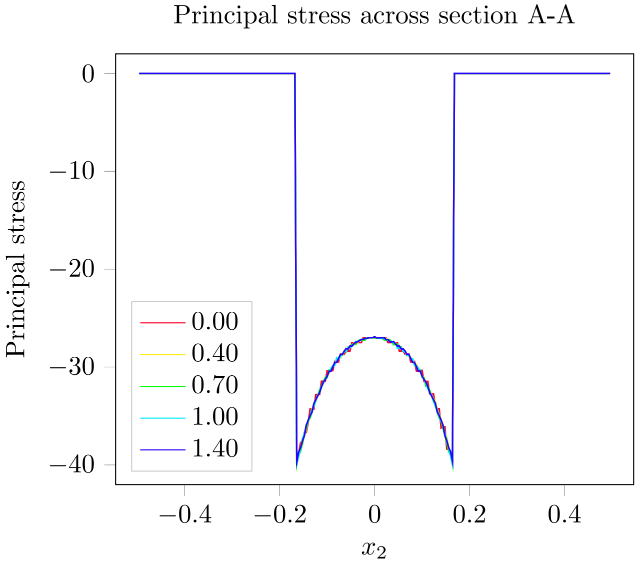


Figure 2.5: Minimum principal stress over cross section A-A for orientations 0.0, 0.4, 0.7, 1.0 and 1.4 rad.

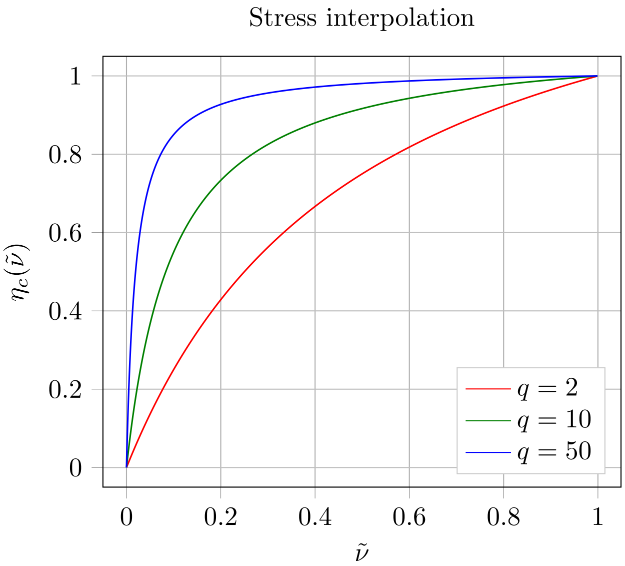


Figure 2.6: Stress penalization function.

to zero. However, our focus is in obtaining black-and-white designs so we are not interested in accurately calculating the stress in regions with intermediate volume fraction. Therefore, when computing the stress we use a relaxed stress formulation similar to [56] wherein

$$\mathbb{C}(\tilde{\nu}) = \eta_c(\tilde{\nu})\mathbb{C}_0 \,, \tag{2.21}$$

where η_c is the inverse RAMP function

$$\eta_c(\tilde{\nu}) = \tilde{\nu}\frac{1+q}{1+q\tilde{\nu}} \,, \tag{2.22}$$

cf. Figure 2.6. Basically, when computing the displacement in regions with intermediate volume fraction, we penalize $\mathbb{C}$ such that it is more compliant which generates a displacement field which is artificially "large". We now combine this large displacement field with a stiff $\mathbb{C}$ to compute an artificially large stress field. Because of the volume minimization, the optimizer deems such regions inefficient and hence reduces their size.

The stress constraint Equation (2.20) is a pointwise constraint. When solving the elasticity equation using the finite element method, the pointwise constraint translates into one constraint per finite element; it is evaluated at the element centroid $\mathbf{x}_i$. This results in a computationally challenging optimization problem. To resolve this burden, the pointwise constraints $\sigma_{VM}(\mathbf{x}_i) \leq \sigma_y$ are replaced by the single global maximum constraint $\max\limits_{\mathbf{x}_i} \sigma_{VM}(\mathbf{x}_i) \leq \sigma_y$. However, this constraint is not differentiable, which hinders the optimization algorithm. As such aggregation strategies in [34, 104] approximate the maximum value of the Von Mises stress field with a p-norm or a Kreisselmeier-Steinhauser (KS) function [40]. Obviously the single global measure cannot represent the pointwise field values. To address this shortcoming, several strategies employ regional measures [47, 56, 72], where the single agglomerated constraint over the entire domain is replaced by several such constraints over subdomains. Another shortcoming with these aggregated measures is that they only approximate the maximum field value. A renormalization strategy was presented in [56] to improve the approximation. However, this approach results in a non-differentiable constraint although the effect of the non-differentiability lessens as the optimization converges.

We apply a different aggregation strategy that is commonly used in PDE-constrained optimization [46] and in [10] for stress constrained optimization via the topological derivative. It requires neither renormalization nor regional clustering techniques. We replace the pointwise constraint Equation (2.20) with the global constraint $\|\sigma_{VM} - \sigma_y\|_+ = \int_D R(\frac{\sigma_{VM}}{\sigma_y})dV$ where R is the shifted ramp function: $R(x) = x - 1$ if $x > 1$ and $R(x) = 0$ otherwise. For the optimization, we replace the ramp function with a smooth approximation R_p such that

$$R_p(x) = (1 + (x)^p)^{\frac{1}{p}} - 1\,, \tag{2.23}$$

cf. Figure 2.7. In our calculations, we use $p = 8$ and leave it constant throughout the optimization.

The constraint of Equation (2.23) is enforced via a penalty method with a penalty parameter γ so that the topology optimization problem now reads

$$\min_{\nu \in (0,1]} \theta_0(\nu) = \int_D \tilde{\nu}\; dV + \gamma \left\|\sigma_{VM} - \sigma_y\right\|_+ \,, \tag{2.24}$$

$$\text{s.t. } \mathbf{u} \in V \text{satisfies } a(\nu; \mathbf{u}, \mathbf{v}) = L(\mathbf{v}) \text{ for all } \mathbf{v} \in V\,. \tag{2.25}$$

The parameter γ is increased throughout the optimization, starting from a small value. This helps alleviate the sharp gradient in Equation (2.23).

We use an adjoint sensitivity analysis to evaluate the variation of a general functional $\theta(\nu)$. Although this derivation is well-known, it behooves us to present it as it also plays a role in our AMR.

We use a reduced space optimization formulation, hence we must consider dependency of $\mathbf{u}$ on the design variable ν, i.e. $\mathbf{u} \rightarrow \mathbf{u}(\nu)$ so that $\hat{\theta}(\nu) = \theta(\nu, \mathbf{u}(\nu))$. In this way

$$\delta\hat{\theta}(\nu; \delta\nu) = \delta_\nu \theta(\nu, \mathbf{u}(\nu); \delta\nu) + \delta_{\mathbf{u}} \theta(\nu, \mathbf{u}(\nu); \delta\mathbf{u}(\nu; \delta\nu))\,, \tag{2.26}$$

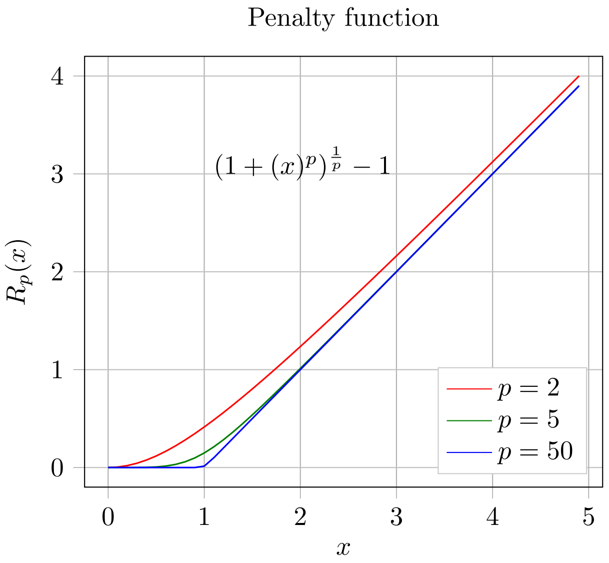


Figure 2.7: Smoothed shifted function.

where $\delta\hat{\theta}(\nu;\delta\nu)$ is the variation of $\hat{\theta}$ at ν acting on $\delta\nu$; $\delta_{\mathbf{u}}\theta$ is the variation of θ at $\mathbf{u}$ acting on $\delta\mathbf{u}$, $\delta_\nu\theta$ is the variation of θ at ν acting on $\delta\nu$ and $\delta\mathbf{u}$ is the variation of $\mathbf{u}$ at ν acting on $\delta\nu$. In the adjoint method, we annihilate the implicit variation $\delta\mathbf{u}(\nu;\delta\nu)$. To do this we take the variation of the state Equation (2.25) with respect to the design field ν and augment this zero term to $\delta\hat{\theta}$.

$$\delta\hat{\theta}(\nu;\delta\nu) = \delta_\nu\theta(\nu,\mathbf{u}(\nu);\delta\nu) \tag{2.27}$$

$$+\,\delta_{\mathbf{u}}\theta(\nu,\mathbf{u}(\nu);\delta\mathbf{u}(\nu;\delta\nu)) \tag{2.28}$$

$$-\,\delta_\nu a(\mathbf{u},\mathbf{v};\delta\nu) \tag{2.29}$$

$$-\,\delta_{\mathbf{u}} a(\mathbf{u},\mathbf{v};\delta\mathbf{u}(\nu;\delta\nu))\,, \tag{2.30}$$

where $\delta_\nu a$ and $\delta_{\mathbf{u}}a$ are the variations of the bilinear form a with respect to ν and $\mathbf{u}$. In our linear elasticity formulation

$$\delta_{\mathbf{u}}a(\mathbf{u},\mathbf{v};\delta\mathbf{u}(\nu;\delta\nu)) = a(\mathbf{v},\delta\mathbf{u}(\nu;\delta\nu))\,, \tag{2.31}$$

and

$$\delta_\nu a(\mathbf{u},\mathbf{v};\delta\mathbf{u}(\nu;\delta\nu)) = \int_D \delta r(\tilde{\nu};\delta\tilde{\nu}(\nu;\delta\nu))\mathbb{C}[\nabla\mathbf{u}]\cdot\nabla\mathbf{v}\;dV\,. \tag{2.32}$$

To annihilate $\delta\mathbf{u}(\nu;\delta\nu)$ in Equation (2.30), we define the arbitrary $\mathbf{v} \in V$ such that

$$a(\mathbf{v}, \delta\mathbf{u}) = \delta_{\mathbf{u}}\theta(\nu, \mathbf{u}(\nu); \delta\mathbf{u}) \text{ for all } \delta\mathbf{u} \in V\,, \tag{2.33}$$

where we use the symmetry of the bilinear from a. Having calculated $\mathbf{v}$, $\delta\hat{\theta}$ reduces to.

$$\delta\hat{\theta}(\nu;\delta\nu) = \delta_{\nu}\theta(\nu, \mathbf{u}(\nu); \delta\nu) - \delta_{\nu}a(\mathbf{u}, \mathbf{v}, \delta\nu)\,. \tag{2.34}$$

Summarizing, to calculate $\delta\hat{\theta}(\nu)$, we first solve the primal problem of Equation (2.25) to obtain $\mathbf{u}$, we compute the partial derivative information $\delta_{\mathbf{u}}\theta(\nu, \mathbf{u}(\nu); \delta\mathbf{u})$ and $\delta_{\nu}\theta(\nu, \mathbf{u}(\nu); \delta\nu)$, we solve the adjoint problem of Equation (2.33) for $\mathbf{v}$ and finally we compute the variation $\delta\hat{\theta}$ from Equation (2.34).

2.4.1 Adaptive mesh refinement

To produce meaningful designs, we must compute the stress field with as much accuracy and efficiency as possible, hence the motivation to use AMR. More importantly, in optimization, AMR also provides an accurate and efficient means to compute the cost and constraint functions. Indeed, the engineer is basing the entire design on the values of these functions so their computed values must be accurate.

The accuracy of our finite element computation is directly dependent on the spatial discretization. To gauge this accuracy we compute an error measure which allows us to best allocate the spatial discretization parameters while simultaneously achieving the desired accuracy. In the context of finite element methods, these measures are called a posteriori error estimates. They bound the error in terms of the current solution approximation and are computed by summing the element error indicators over the mesh. In our h-refinement AMR strategy, a marking strategy identifies the elements with largest and smallest errors for refinement and coarsening. The elements to be refined are divided in four children elements, which belong to a different refinement level, cf. Figure 2.8. Those elements to be coarsened are removed along with the corresponding children from the same parent, cf. Figure 2.9. The refinement creates so called “hanging nodes” that break the continuity of the displacement field. To ensure continuity, constraints are imposed that relate these nodes’ displacements to their parents’. We only allow for hanging nodes between children and their parents neighbors; i.e. not between children and their grandparents neighbors. Such situations are corrected by adding more elements, cf. Figure 2.10.

As just mentioned, the AMR strategy relies on element error indicators. The theory of a posteriori finite element error estimation for elliptic problems is well established cf. the monographs [6, 95]. The a posteriori error estimates are categorized into three types: explicit residual-based error estimates, implicit residual-

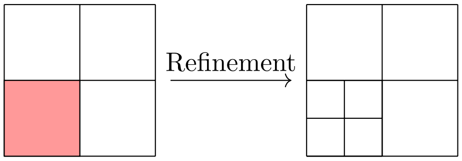


Figure 2.8: Element marked for refinement in red and the resulting refinement with now two levels.

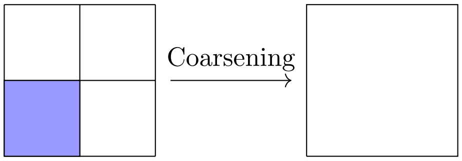
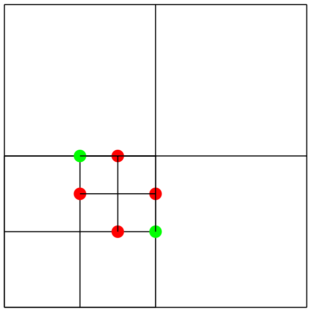


Figure 2.9: Element marked for coarsening in blue and the coarsening result.

based error estimates and gradient or flux recovery based error estimates. The explicit and implicit methods approximate the error with the current finite element solution. The explicit estimates are easy to implement, however they render bounds with problem dependent constants that can be difficult to assign. The implicit estimates require the solution of multiple regional boundary value problems over single elements or small element patches. The problem dependent constants are avoided at the behest of additional computational expense and onerous implementations. Error estimates based on gradient or flux recovery post-process the finite element displacement field $\mathbf{u}_h$ to obtain an improved approximation of the stress field $\hat{\boldsymbol{\sigma}}(\mathbf{u}_h)$ which is smooth and has better convergence properties versus the nominally computed discontinuous stress field $\boldsymbol{\sigma}(\mathbf{u}_h)$. The element error indicator is based on the difference between $\hat{\boldsymbol{\sigma}}(\mathbf{u}_h)$ and $\boldsymbol{\sigma}(\mathbf{u}_h)$. The most popular flux recovery method is the Zienkiewicz-Zhu [111, 112], which locally projects $\boldsymbol{\sigma}(\mathbf{u}_h)$ onto a higher-order polynomial approximation space over a patch of neighboring elements. Its popularity is due to its easy implementation, generality and accuracy. However, it is well known that adaptive mesh refinement algorithms using this estimator are not effective for interface problems, e.g. such as those encountered in topology optimization problems [24, 70]. This is because the tangential components of the stress field $\mathbf{t} \cdot \boldsymbol{\sigma}\mathbf{t}$, where $\mathbf{t}$ is any vector perpendicular to the interface normal vector, are discontinuous at the interface. Smoothing these components results in unnecessary overrefinement. Knowing that the stress field exists in the $H(\text{div}, D) = \{v \in L^2(D, \mathbb{R}^n) : \text{div } v \in L^2(D)\}$ space, [24] projects the stress field onto finite elements in this space. However, we do not follow this approach because $H(\text{div}, D)$ finite elements are notoriously difficult to implement, especially in parallel computing environments.

Based on all these considerations, we use explicit residual-based a posteriori error estimates and follow



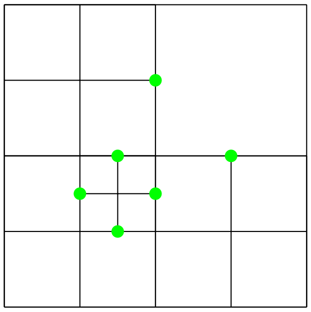


Figure 2.10: Allowed hanging nodes green and disallowed hanging nodes in red (top) and additional refinement to resolve disallowed hanging nodes (bottom).

[94] wherein the element error indicator for an element $K \in \mathcal{T}_h$, where $\mathcal{T}_h$ is the finite element mesh, is

$$\begin{aligned}\eta_K = \{ & h_K^2 \left\| \nabla \cdot \boldsymbol{\sigma}(\mathbf{u}_h) + \mathbf{f} \right\|^2_{L^2(K)} \\ & + \frac{1}{2} \sum_{E \in \mathcal{E}(K)} h_E \left\| j(\boldsymbol{\sigma}(\mathbf{u}_h) \cdot \mathbf{n}) \right\|^2_{L^2(E)} \}^{1/2} .\end{aligned} \tag{2.35}$$

In the above, the stress field is computed as

$$\boldsymbol{\sigma}(\mathbf{u}_h) = r(\tilde{\nu}) \mathbb{C}[\nabla \mathbf{u}_h] \tag{2.36}$$

and

$$j(\boldsymbol{\sigma} \cdot \mathbf{n})_E = \begin{cases} [\![\boldsymbol{\sigma}(\mathbf{u}_h) \cdot \mathbf{n}]\!] & E \notin \mathcal{E}_D, \mathcal{E}_N \\ \mathbf{t}_N - \boldsymbol{\sigma}(\mathbf{u}_h) \cdot \mathbf{n} & E \in \mathcal{E}_N \end{cases} \tag{2.37}$$

is the traction jump across the element face E, which belongs to the set of faces of element K, $\mathcal{E}(K)$. $\mathcal{E}_D \subset \Gamma_D$ is the set of element faces with prescribed Dirichlet boundary condition and $\mathcal{E}_N \subset \Gamma_N$ is the set of element faces with prescribed $\mathbf{t}_N$ Neumann boundary condition. h_K is the element diameter and h_E is the element face area.

Summing the element error η_K gives the global error

$$\eta = \left(\sum_{K \in \mathcal{T}_h} \eta_K^2 \right)^{1/2} . \tag{2.38}$$

This in turn is used to compute upper and lower bounds on the energy norm of the error, i.e.

$$|||\mathbf{u} - \mathbf{u}_h||| = \int_D r(\hat{\nu}) \mathbb{C}[\nabla(\mathbf{u} - \mathbf{u}_h)] \cdot \nabla(\mathbf{u} - \mathbf{u}_h) \; \mathrm{d}V \, . \tag{2.39}$$

The upper bound $C\eta$ where C is a constant ensures that the error estimate is conservative, i.e.

$$|||\mathbf{u} - \mathbf{u}_h||| \leq C\eta \, . \tag{2.40}$$

On the other hand, the lower bound $c\eta$, where c is also a constant, ensures the error estimate is not excessively conservative, i.e.

$$c\eta \leq |||\mathbf{u} - \mathbf{u}_h||| \, . \tag{2.41}$$

To refine the mesh, we use the element error indicator η_K, cf. Equation (2.35). Refinement (coarsening) is done in those elements whose error is above the 70-th percentile (below the 5-th percentile) of the error indicator distribution.

Conventional error estimates are based on global measures such as the energy norm as in Equation (2.39) or the L^2-norm $\|\mathbf{u} - \mathbf{u}_h\|_{L^2}$. However these measures are not effective as we are only interested in accuracy of our optimization cost and constraint functions. For this purpose, goal-oriented error estimates were developed [14, 69]. As alluded to above, these estimates require the solution of the adjoint problem, which within our optimization context we have already solved to obtain the sensitivities.

To present the goal estimate, we discretize $\mathbf{u}$, i.e. we now find $\mathbf{u}_h \in V_h$ such that

$$a(\mathbf{u}_h, \mathbf{v}_h) = L(\mathbf{v}_h) \text{ for all } \mathbf{v}_h \in V_h \, , \tag{2.42}$$

where V_h is the finite element discretization of the function space V. Substracting Equation (2.11) from this and noting that $V_h \subseteq V$ yields the Galerkin orthogonality condition.

$$a(\mathbf{u} - \mathbf{u}_h, \mathbf{v}_h) = 0 \; \forall \mathbf{v}_h \in V_h \, , \tag{2.43}$$

i.e. the error $\mathbf{u} - \mathbf{u}_h$ is a-orthogonal to V_h.

We are interested in defining the error $\theta(\mathbf{u}) - \theta(\mathbf{u}_h)$ in a goal functional $\theta(\mathbf{u}) : V \to \mathbb{R}$, i.e. a cost or constraint function. We start by restating the adjoint problem in Equation (2.33). Dropping the argument ν for conciseness and replacing $\delta\mathbf{u}$ with $\mathbf{w}$ we find $\mathbf{v} \in V$ such that

$$a(\mathbf{v}, \mathbf{w}) = \delta_{\mathbf{u}}\theta(\mathbf{u}; \mathbf{w}) \text{ for all } \mathbf{w} \in V\,. \tag{2.44}$$

Expanding $\theta(\mathbf{u}) - \theta(\mathbf{u}_h)$ in first-order Taylor series and applying Equation (2.44) we find

$$\begin{aligned}
\theta(\mathbf{u}) - \theta(\mathbf{u}_h) &= \theta(\mathbf{u}) - \theta(\mathbf{u} + \mathbf{u}_h - \mathbf{u})\,, && (2.45)\\
&= \theta(\mathbf{u}) - \theta(\mathbf{u}) - \delta_{\mathbf{u}}\theta(\mathbf{u}; \mathbf{u}_h - \mathbf{u}) \\
&\qquad - o\left\|\mathbf{u}_h - \mathbf{u}\right\|^2\,, && (2.46)\\
&= \delta_{\mathbf{u}}\theta(\mathbf{u}; \mathbf{u}_h - \mathbf{u})\,, && (2.47)\\
&= a(\mathbf{v}, \mathbf{u} - \mathbf{u}_h)\,, && (2.48)\\
&= a(\mathbf{u} - \mathbf{u}_h, \mathbf{v})\,. && (2.49)
\end{aligned}$$

Equation (2.49) [1] gives us the error estimate for the quantity of interest θ, however, the exact adjoint solution $\mathbf{v}$ is unknown, rather we compute $\mathbf{v}_h$ using the discretization that is used to compute $\mathbf{u}_h$, but this yields a zero error estimate by Galerkin orthogonality of (2.43). A common technique to resolve this issue is to calculate $\mathbf{v}_h$ over a more refined mesh or to post-process $\mathbf{v}_h$ to obtain a higher order approximation. We instead follow [38] and add the zero Galerkin orthogonality Equation (2.43) to Equation (2.49) and apply Cauchy-Schwarz inequality to obtain bounds on the error.

$$\begin{aligned}
\theta(\mathbf{u}) - \theta(\mathbf{u}_h) &= a(\mathbf{u} - \mathbf{u}_h, \mathbf{v} - \mathbf{v}_h)\,, && (2.50)\\
&= \int_D r(\hat{\nu})\mathbb{C}[\nabla(\mathbf{u} - \mathbf{u}_h)] \\
&\qquad \cdot [\nabla(\mathbf{v} - \mathbf{v}_h)]\; dV\,, && (2.51)\\
&\leq \left\|\sqrt{r(\hat{\nu})\mathbb{C}}[\nabla(\mathbf{u} - \mathbf{u}_h)]\right\|_{L^2} \\
&\qquad \times \left\|\sqrt{r(\hat{\nu})\mathbb{C}}[\nabla(\mathbf{v} - \mathbf{v}_h)]\right\|_{L^2}\,, && (2.52)\\
&= |||\mathbf{u} - \mathbf{u}_h||| \, |||\mathbf{v} - \mathbf{v}_h|||\,. && (2.53)
\end{aligned}$$

[1] In Equation (2.47), we have neglected the higher order terms. For details on higher order nonlinear functionals, we refer to [14].

With these bounds, we can reformulate the error estimate in terms of the energy norms:

$$\theta(\mathbf{u}) - \theta(\mathbf{u}_h) \leq |||\mathbf{u} - \mathbf{u}_h||| |||\mathbf{v} - \mathbf{v}_h||| \,, \tag{2.54}$$

$$\leq \sum_{K \in \mathcal{T}_h} |||\mathbf{u} - \mathbf{u}_h|||_K |||\mathbf{v} - \mathbf{v}_h|||_K \,, \tag{2.55}$$

$$e_K = |||\mathbf{u} - \mathbf{u}_h|||_K |||\mathbf{v} - \mathbf{v}_h|||_K \,. \tag{2.56}$$

The expression inside the summation is the K element error indicator Equation (2.56) which is used to mark cells for refinement and coarsening throughout the optimization. We compute the error indicator $|||\mathbf{u} - \mathbf{u}_h|||_K = \eta_K$ from Equations (2.35)–(2.36). The estimate for $|||\mathbf{v} - \mathbf{v}_h|||_K$ is similarly computed using Equations (2.35)–(2.36), but we replace $\mathbf{u}_h$ with $\mathbf{v}_h$ and $\mathbf{f}$ and $\mathbf{t}_N$ with their corresponding body load and traction from the adjoint linear term $\delta_{\mathbf{u}}\theta$. With the error indicator e_K for each element, we proceed as in the energy error approach and refine the elements whose error is above the 70-th percentile of the error indicators distribution and coarsen those below the 5th percentile. We only use the element error indicator (2.56) and not the goal estimate (2.55), as it is not accurate [38].

2.4.2 Optimization algorithm and refinement strategy

Our optimization framework uses the C++ MMA [89] implementation by [3]. We implement a continuation strategy for the β parameter in the threshold function Equation (2.8) and for the penalty parameter γ in Equation (2.25). We do not wait for MMA to converge to change these parameters, rather we increase them following a heuristic strategy.

For each optimization iteration k, we use the current design estimate ν_k to calculate the state $\mathbf{u}_h$ and adjoint $\mathbf{v}_h$ states, which are needed to calculate the cost θ_0 and constraint θ_i functions and their gradients $\nabla\theta_0$ and $\nabla\theta_i$ for $(i = 1, ..., n_i)$. These are fed into the optimizer which then updates the design to ν_{k+1}, cf. Algorithm 1. During each optimization iteration, we determine if the relative change in the cost function between iterations is smaller than tol_{AMR}. If so, we invoke our AMR strategy described earlier. The MMA mesh dependent information, i.e. the design field and Lagrangian multiplier variables over the last three iterations [89], are interpolated onto the new mesh. Next, the filter kernel is reconstructed. We update the threshold and penalty parameters β and γ following a heuristic strategy explained in the numerical examples. Finally, tol_{AMR} is updated such that its value is halved after each refinement. We reset tol_{AMR} to its starting value once the threshold parameter β exceeds 200 so as to reinitiate the refinement. The termination criteria for the optimization is based on the change in the design or the maximum number of iterations, set to 500.

Algorithm 1 Algorithm outline.

```
Build filter.
while $\|\nu_k - \nu_{k+1}\| > tol$ or $it < maxit$ do
    Solve state problem to obtain $\mathbf{u}$
    Solve adjoint problem to obtain $\mathbf{v}$
    Calculate $\theta_0(\nu_k)$, $\nabla\theta_0(\nu_k)$
    Update $\nu_{k+1}$ according to the MMA algorithm.
    if $\frac{\theta_0(\nu_k)-\theta_0(\nu_{k+1})}{\theta_0(\nu)} \leq tol_{AMR}$ then
        Calculate $\eta_K$ for $K = 1...nel$
        Sort element errors in decreasing order.
        Refine of 30 % of the elements with highest errors.
        Coarsen of 5 % of the elements with lowest errors.
        Update filter with the new mesh.
        Project MMA data into the new discretization.
        Update $tol_{AMR}$
    end if
    if $\beta$ satisfies $\beta$-criteria then
        Update $\beta$
    end if
    if $\gamma$ satisfies $\gamma$-criteria then
        Update $\gamma$
    end if
end while
```

2.4.3 Finite Element Implementation

The finite element computations are performed in parallel thanks to the libMesh finite element library [51]. For solvers we use PETSc [12, 13], and the HYPRE [60] preconditioner. As both the adjoint and primal problems share the same stiffness matrix, we recycle the preconditioner to avoid the overhead of multiple builds. libMesh uses a quadtree/octree data structure in their adaptive mesh refinement implementation. We use bilinear Lagrangian elements and accomodate the hanging nodes via constraint equations. Our code design is similar to that in [38] wherein the users need only write the optimization cost and constraint functions and their derivatives with respect to $\mathbf{u}$ and ν. All the optimizations were run on a 8-core 2.60 GHz Intel Xeon E5-2670 processor.

2.4.4 Numerical examples

We first benchmark our algorithm with the eyebar example from [10]. The design domain in Figure 2.11 is discretized with the Gmsh library [28] using second order triangles to capture the circumference with more precision and avoid sharp angles when refining the mesh. The largest triangle has an inradius of 0.15 mm and the smallest, 0.10 mm, placed near the hole. The load applied on the left hand side of the hole follows the distribution $\mathbf{t}(x, y) = (-((y-4)^2 - 1.5^2), 0.0)$ and Dirichlet boundary conditions are applied to the 2 mm segment in the right side. The isotropic material has a Young's modulus $E = 1.0$ MPa, a Poisson's ratio of $\nu = 0.3$ and a yield criteria is $\sigma_y = 5\ MPa$. The filter radius is $\epsilon = 0.6\ mm$. The initial mesh contains 681 elements. We limit the maximum refinement level to four because the relatively large filter radius compared to the mesh dimensions causes the filter kernel to exhaust the computer's memory if the refinement is too high.

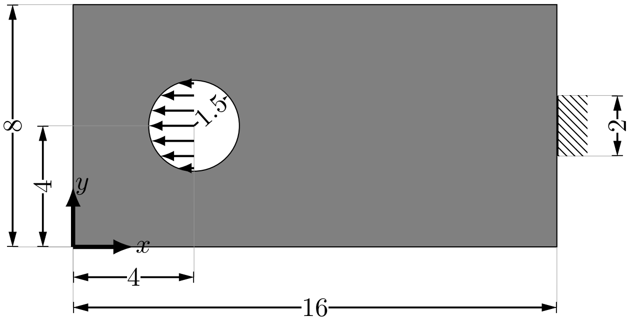


Figure 2.11: Gray design domain, loads and boundary conditions. All dimensions are in mm.

Our continuation strategy to obtain optimized designs is as follows. First, the threshold parameter β starts with a value of 1 and is increased by 5 every 20 iterations after iteration 50 to avoid having the design fall into a local minimum. Second, the penalty parameter γ starts with a value of 10 and is increased by 50 every 50 iterations after iteration 50. The tolerance to trigger refinement tol_{AMR} is 1e-2.

We plot the optimized design in Figure 2.12. As we can see, part of the structure is stuck to the right side of the hole. The reason being that, thanks to the threshold function of Equation (2.8), the sensitivities are only significant in the material-void transition regions, i.e. at the design boundaries. If the hold-all domain boundary ∂D coincides with the intended design domain boundary, it is difficult to move the latter, as the sensitivity is zero there because the design boundary interface cannot fully resolve itself. For this reason we expand the domain into the hole, cf. Figure 2.13. To recover the intended design domain, we impose a zero volume constraint for the thresholded volume fractions $\tilde{\nu}$ in the extended, i.e. red, region.

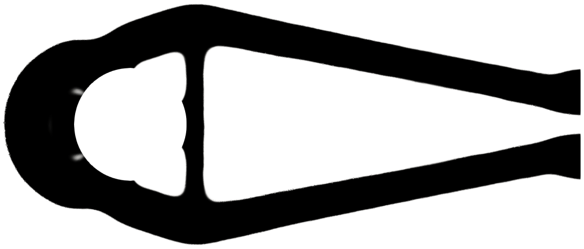


Figure 2.12: Optimized design for the eyebar geometry.

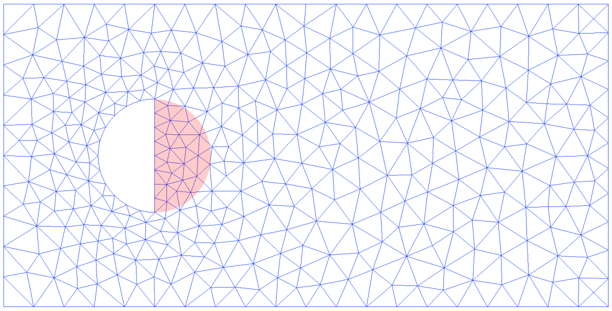


Figure 2.13: Initial mesh with extended simulation domain in red.

We plot the new design in Figure 2.14. The nonlocal nature of the filter causes the zero volume constraint to extend its influence beyond the actual hole. To resolve this, in the extended domain, we equate $\nu = 0$ and $\epsilon = 0$ to avoid nonzero filtered volume fractions within the hole.

We first run an optimization using the global energy norm error indicator described in Section 2.4.1. The result with this approach cf. Figure 2.15, shows how the algorithm fails to obtain a complete 0/1 design. Indeed, the lack of a global convergence mechanism in MMA causes an oscillatory behavior in the boundary region near the hole. We also plot the Von Mises Stress field in Figure 2.16. The optimization time was 1900.8 seconds, out of which 339.5 seconds were devoted to the filter kernel construction.

Using the same parameters as the previous case, we run an optimization using the goal-oriented error indicator with respect to the cost function, cf. Equation (2.24). The new optimized design in Figure 2.17 is almost identical to that obtained with the global energy error indicator, cf. Figure 2.15. The optimization took 1893.6 seconds, with 313.91 devoted to the filter kernel. The evolution of the cost function is similar as well, but the refinement process is not, cf. Figure 2.18. The goal error indicator initially refines fewer elements than the energy error where after it refines more until the difference stagnates due to the ceiling on the maximum level of refinement. The upticks in the value of the cost function correspond to mesh refinements, which resulted in higher stress values.

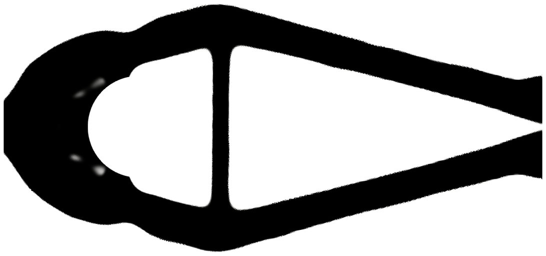


Figure 2.14: Optimized design with a zero volume constraint in the extended region.

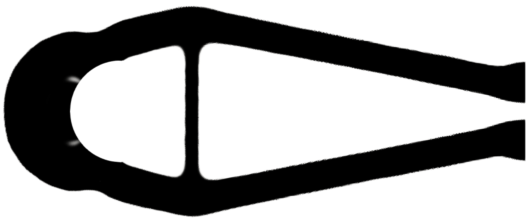


Figure 2.15: Optimized design for the eyebar geometry with the global energy error indicator and $\nu = 0$ and $\epsilon = 0$ in the extended simulation domain.

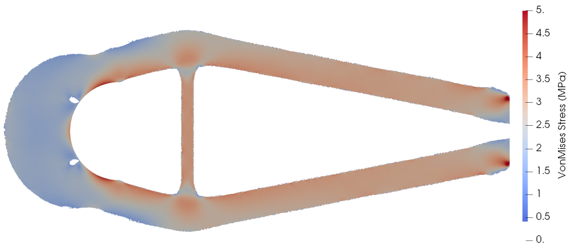


Figure 2.16: Von Mises stress field for the eyebar design in Figure 2.15.

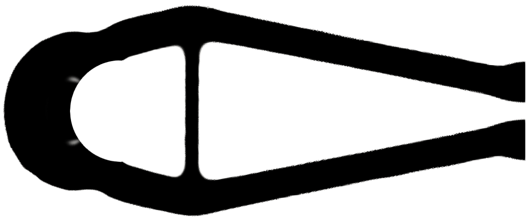


Figure 2.17: Optimized design for the eyebar geometry with the goal-oriented error indicator and $\nu = 0$ and $\epsilon = 0$ in the extended simulation domain.

For the sake of comparison, we study the "L-bracket" problem described in [56], cf. Figure 2.19. We apply 0.08 N/mm traction over the load region, the isotropic material model has again a Young's modulus of $E = 1.0$ MPa , Poisson's ratio of $\nu = 0.3$ and $\epsilon_\nu = 10^{-4}$; the filter radius is $\epsilon = 2.0$ mm and the maximum allowable Von Mises stress is $\sigma_y = 2$ MPa. The initial design uses the mesh in Figure 2.20 which defines refinement level 0.

Again, there are several knobs that need to be adjusted to obtain the optimized design. First, the threshold parameter β in Equation (2.8) starts with a value of 1 and is not increased until iteration number

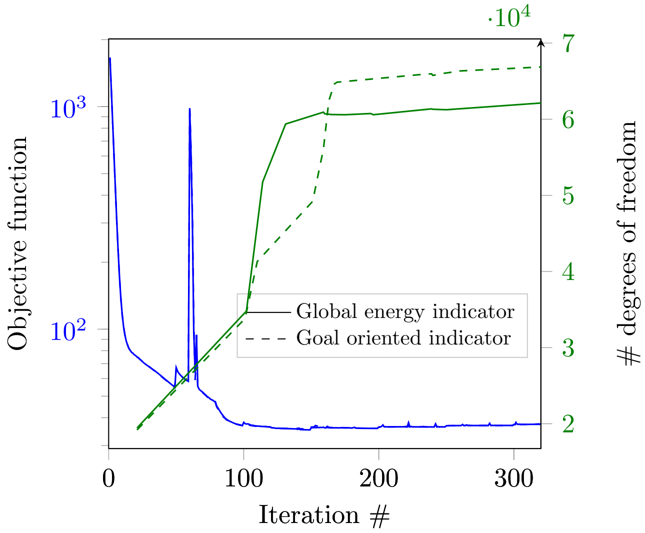


Figure 2.18: Cost function and discretization degrees of freedom histories for the eyebar.

100. Thereafter then increased by 5 every 20 iterations up to a maximum value of 250. This is necessary to obtain a sharp interface and also prevents the design from falling into local minima. Second, the penalty parameter γ starts with an initial value of 1 and increases by 2 every 20 iterations after iteration 50. Third, the maximum refinement level is fixed at 3 until iteration 200, 5 until iteration 400, and 8 thereafter. These strategies give us the best results. For example, it prevents areas with intermediate volume fraction values from being overly refined given that they are eliminated with the threshold function later in the optimization.

The tolerance to trigger the refinement tol_{AMR} is initially 1e-2; it is halved after each refinement. As such tol_{AMR} can become very small early in the design process, i.e. while the primary topology is still evolving. This is attributed to the threshold function, which makes the problem similar to a shape optimization. Therefore, we reset tol_{AMR} =1e-2 every 80 iterations after iteration 100.

As in the previous example, we extend the simulation domain beyond the design domain, cf. the red region in Figure 2.20. We likewise equate $\nu = 0$ and $\epsilon = 0$ in these red areas. This approach however, does not work well for the L-bracket problem. As seen in Figure 2.21, there is still part of the material stuck at the L-bracket corner. The corner creates a high stress concentration, but the gradient is zero there due to the threshold operation in (2.8), as explained earlier. For this reason, we revert to our previous approach and place a zero constraint on thresholded volume fractions $\tilde{\nu}$ in the extended region; we no longer have $\nu = 0$ and $\epsilon = 0$. As in Figure 2.20 design, we run an optimization using the global energy norm error indicator described in Section 2.4.1. Our result in Figure 2.22 shows an optimized design with a volume of 1228.49 mm^2. The final mesh contains 524,842 elements. The same mesh with a uniform refinement corresponding to the smallest element would contain 6,160,384 elements.

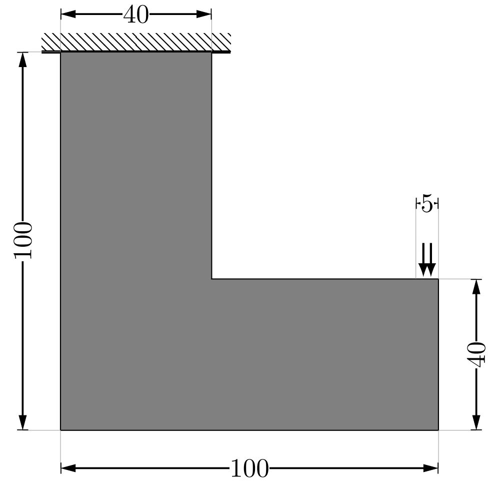


Figure 2.19: Gray design domain, loads and boundary conditions.. All dimensions are in mm.

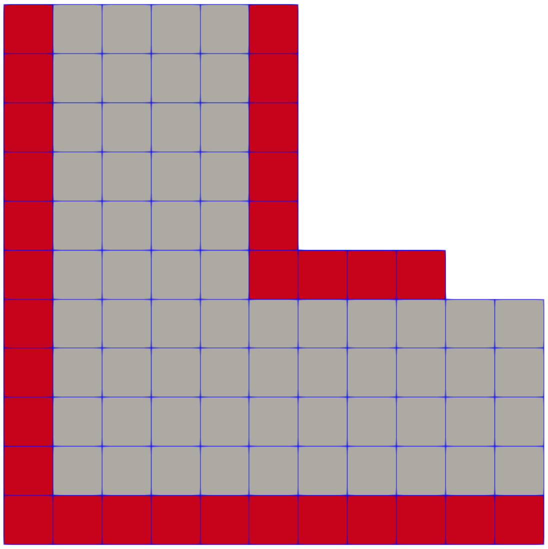


Figure 2.20: Initial mesh for the optimization and actual design domain shown in gray. A zero volume constraint is imposed in the extended domain shown in red.

We plot the Von Mises stress distribution calculated using the SIMP stress formula given in the weak form in Equation (2.13), i.e. $\boldsymbol{\sigma} = r(\tilde{\nu})\mathbb{C}[\nabla \mathbf{u}]$, cf. Figure 2.23. The distribution is considerably more uniform, which indicates a more optimized design. We plot the mesh corresponding to the optimized design with different zoom levels in Figure 2.26. Note the high refinement level that it is necessary to accurately compute the stress field.

The total computational cost was of 2181.6 seconds. Out of this time, 797.2 seconds were devoted on building the cone filter in Equation (2.6). This expensive operation is done every time the mesh is refined

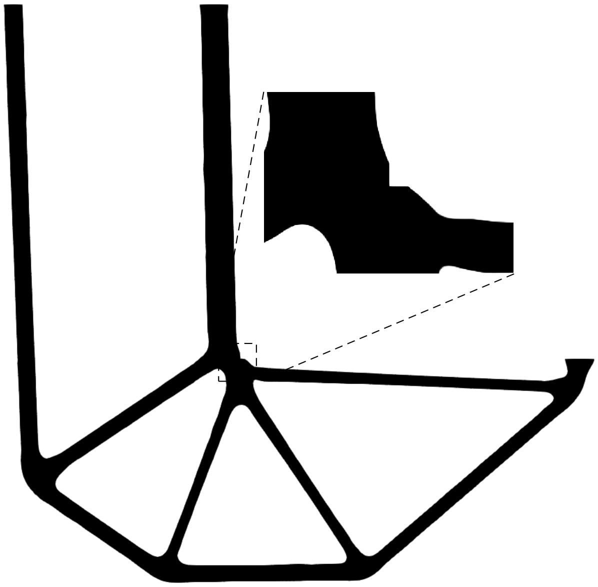


Figure 2.21: Optimized design using $\nu = 0$ and $\epsilon = 0$ in the extended region.

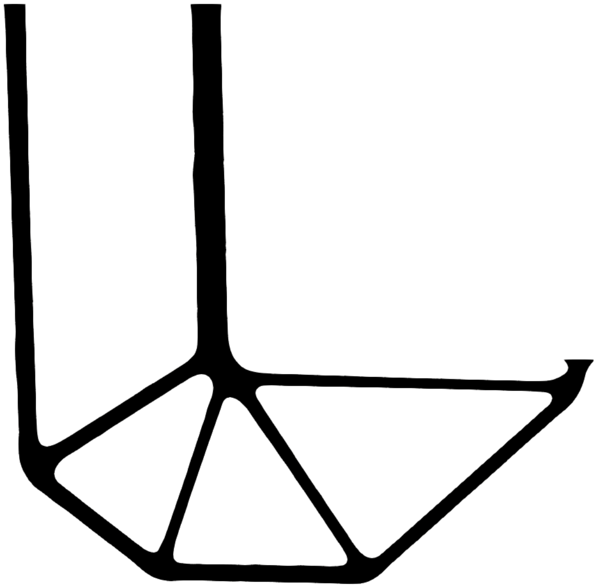


Figure 2.22: Optimized design using zero volume constraint on the thresholded volume fractions in the extended region.

and has an algorithmic complexity that is polynomial with respect to the number of elements. Based on this, it is even more compeling to find out a way to use [55] in the context of AMR.

Using the same parameters as the previous case, we run another optimization but we replace the global energy error indicator for the goal-oriented error indicator, cf. Equation (2.24) to obtain the best possible accuracy. The optimized design is similar than that obtained with the global energy error indicator, cf.

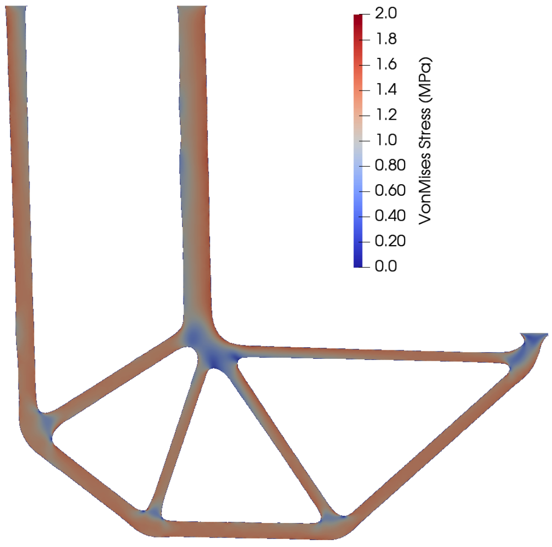


Figure 2.23: Von Mises stress for the design of Figure 2.22.

Figures 2.22 and 2.24. The total volume is 1233.01 mm^2 and the mesh contains 516,346 elements which is less than those obtained via the global energy error indicator.

The goal-oriented optimization took 2120.4 seconds. Figure 2.25 shows the evolution of the cost function and the number of degrees of freedom for the displacement field. The goal oriented error indicator marks fewer elements for refinement than the energy error indicator which translates into a slight computational saving for our case. The global nature of the goal error, Equation (2.24), explains the similar number of elements versus the energy error. A cost function localized in a small region would have resulted in a more localized mesh refinement because the adjoint response would only be significant in the small region.

2.4.5 Design validation with explicit geometry finite element simulation

To validate the Figure 2.22 design, we use the $\tilde{\nu} = 0.5$ level set of the thresholded volume fraction field to define the optimized geometry. A body-fitted mesh is genererated over this domain and a finite element analysis is performed to compute the displacement and stress fields using the finite element library FEniCS [9]. We plot the difference between the fitted and topology optimized (topopt) Von Mises stress fields $\sigma_{VM}^{\text{fitted}} - \sigma_{VM}^{\text{topopt}}$, cf. Figure 2.27. As we can see, most of the difference is within 10% of the maximum von mises stress value of 2 MPa, with $\sigma_{VM}^{\text{topopt}}$ slightly greater than $\sigma_{VM}^{\text{fitted}}$.

For comparison purposes, we run a similar optimization with a single uniform mesh containing 528,384 elements, which is similar to the number of elements in the optimal design of Figure 2.22. This result is

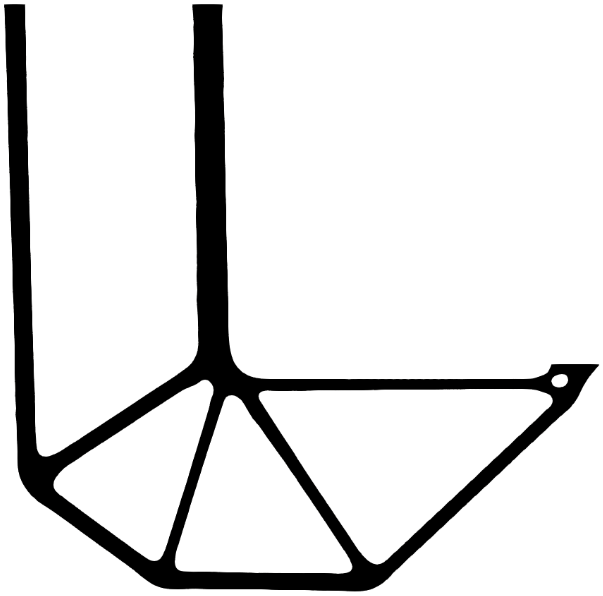


Figure 2.24: Optimized design using the goal oriented error indicator.

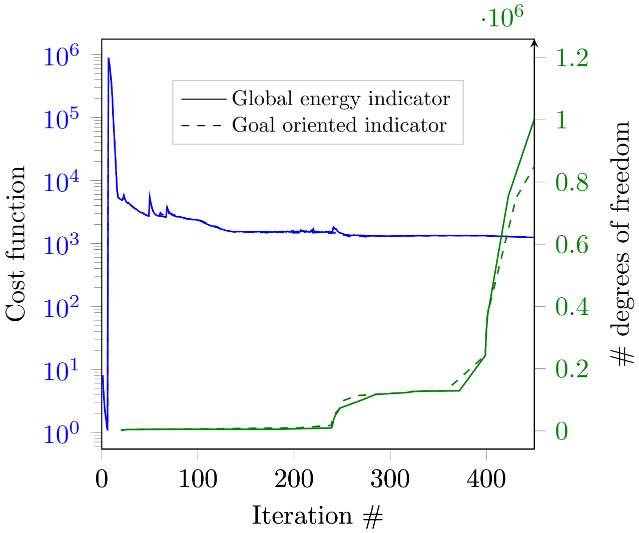


Figure 2.25: Cost function and discretization degrees of freedom histories for the L-bracket.

plotted in Figure 2.28 with a total volume of 1381.9 mm^2. We perform a similar validation study with a body fitted mesh and plot the result in Figure 2.29, as expected the lack of refinement creates a greater difference in the Von Mises stress field over the design boundaries and in the thin structural members. Increasing the refinement with two additional levels to match the most refined elements in our AMR mesh will improve the stress field, but this uniform mesh would result in 8,454,144 elements, versus the 524,842 elements in our AMR mesh.

2.5 Conclusion and future work

We presented a new topology optimization methodology to obtain designs in structural mechanics with accurate stress fields that satisfy yield criteria. We are able to do so by combining a threshold function that sharpens the otherwise blurred boundaries from the filter operation and adaptive mesh refinement that increases the mesh resolution in the boundary region. We validated the stress accuracy by using geometries whose stress fields are known and tabulated. The eye-bar and L-bracket example problems are used to validate our approach. We foresee that it is possible to further improve our results by using a globally convergence optimization algorithm such as the GCMMA [90] and the PDE-based filter [55] to reduce memory requirements.

Future work will address a more rigurous mesh refinement strategies. One that considers the inexactness of the finite element approximation and still ensures global convergence to a local minima [109]. This will avoid our current heuristic and case dependent strategy, e.g. for the assignment of β, tol_{AMR}, etc. Indeed for this approach to be succesful, it is necessary to employ error estimates and error indicators with tighter bounds than the currently used residual based approaches. To save computational resources, uncoupling the discretizations of the volume fraction field and the displacement field could be incorporated.

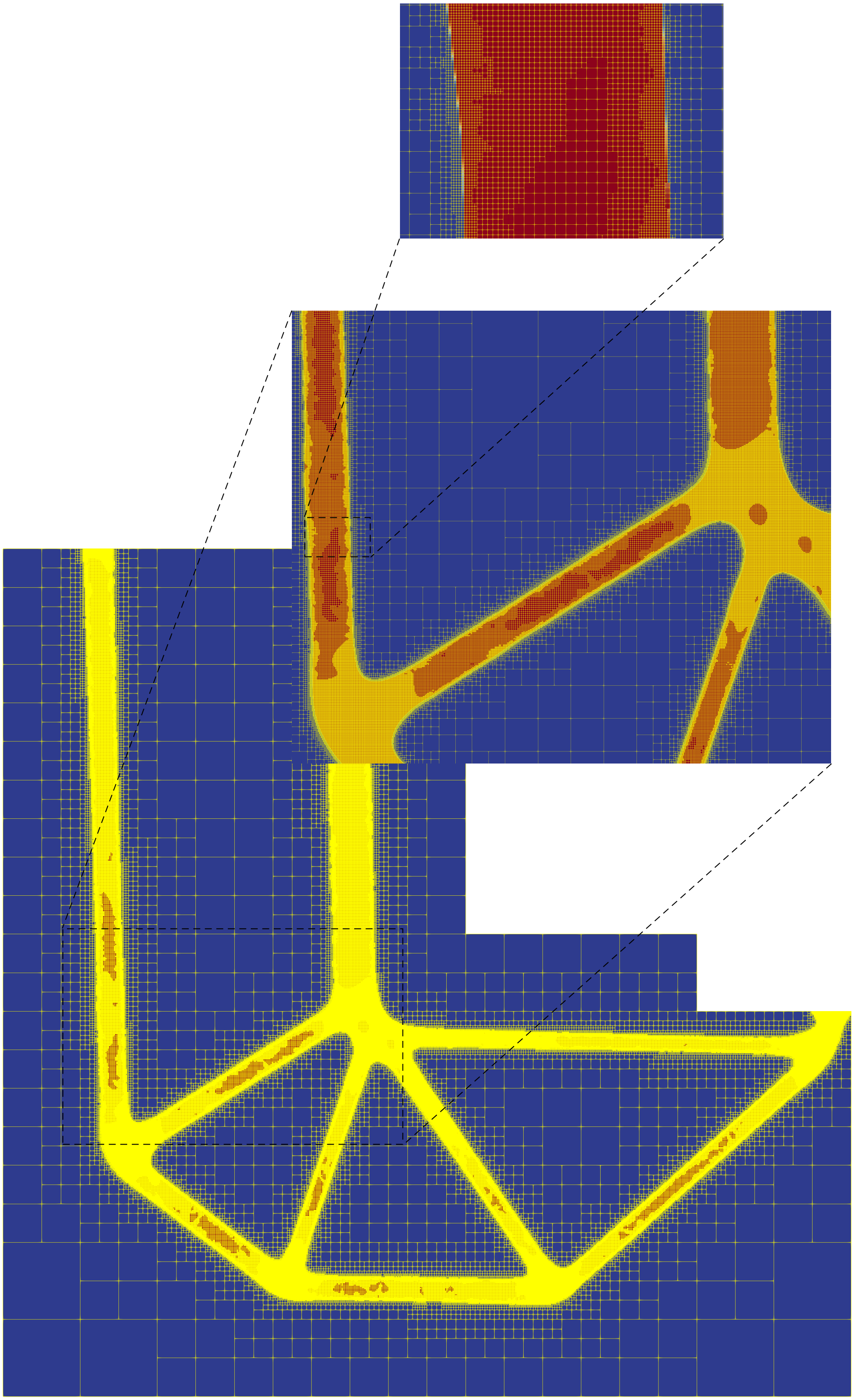


Figure 2.26: Final mesh for the design in Figure 2.22.

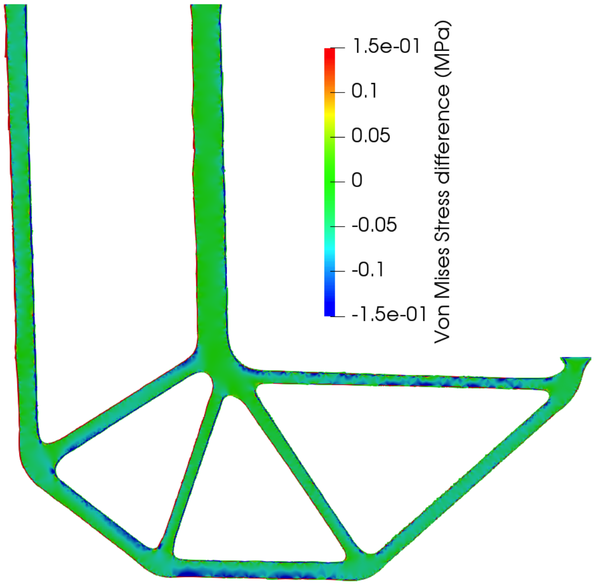


Figure 2.27: Von Mises stress field verification.

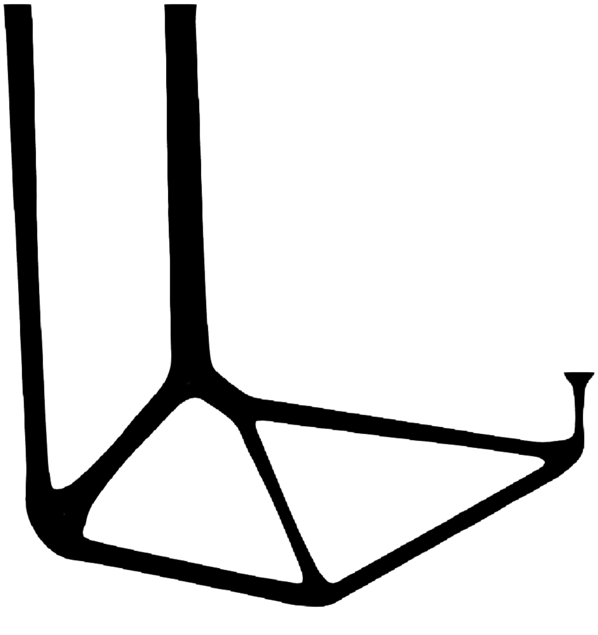


Figure 2.28: Optimized design with a uniform mesh.

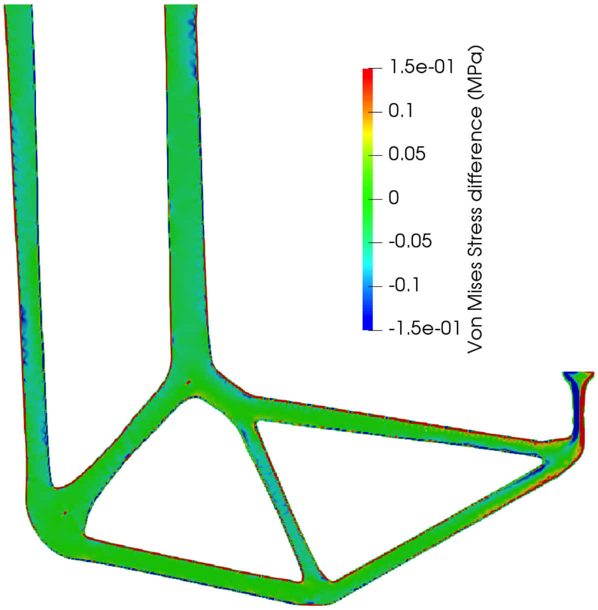


Figure 2.29: Von Mises stress field difference for the optimized design with $\sigma_y = 2$ MPa and a uniform mesh

Chapter 3

Mesh independency in topology optimization

3.1 Introduction

Topology optimization finds the optimal distribution of material in a given design domain D to minimize a cost function and satisfy constraint function inequalities. On a background mesh that represents the design domain, the optimization algorithm places material in individual elements to define the geometry of the optimal design. Regions devoid of material are meaningless, hence the motivation to coarsen the mesh in these regions. By the same token, regions which contain material require a higher mesh resolution. This cost saving strategy that distributes elements with different sizes within the mesh is known as the Adaptive Mesh Refinement (AMR).

In the usual topology optimization scenario, the design is defined by an element-wise material volume fraction. This piecewise uniform function, i.e. design field, belongs to the Hilbert space L_2 defined on D, i.e. $L_2(D)$. If the elements have differente sizes, as in AMR, it is intuitively wrong to think that all design variables have the same contribution to the sensitivities. Here we explain why it is also necessary to accomodate the element size when calculating the inner products involving the design variables within the optimization algorithms. However, this is not done within most optimization algorithms used in the topology optimization community as they assume the design is a vector in the Euclidean space ℓ_2, i.e., simply a vector of length equal to the number of elements in the mesh, cf. IPOPT [96], SNOPT [39], MMA [89], FMINCON [62] and Optimality Criteria [44]. On the other hand, the optimization libraries Optizelle [105], Moola [84], ROL [76] and TAO [32] contain to various extents the capacity of treating design fields as elements of their underlying function spaces. Related work by [81] compares mesh-independent and dependent versions of the steepest descent algorithms for an unconstrained problem and estimates their rates of convergence. It is also necessary to mention that in the PDE-constrained optimization community, optimization algorithms are inherently mesh-independent as they are implemented in the corresponding function space. For instance, [93] implements an infinite-dimensional (inf-dim) primal-dual interior-point method with a Newton solver and [110] implements an inexact sequential quadratic programming method with an adaptive multilevel mesh refinement scheme.

The inconsistency between the L_2 and ℓ_2 spaces is generally not a problem because most of the topology optimization studies use uniform meshes. However, when using meshes with different element sizes, as is the case in AMR, the ℓ_2 viewpoint yields mesh dependent designs, whereas the L_2 approach does not.

This work is laid out as follows: Section 3.2 presents the mathematical tools we need to implement mesh independent optimization algorithms. We use these tools in Section 3.3 to make one of the most popular optimization algorithms in the topology optimization community, the Globally Convergent Method of Moving Asymptotes (GCMMA/MMA) algorithm, mesh independent. In Section 3.4, we validate our theory by solving three common problems in topology optimization. Section 3.5 briefly summarizes our findings and presents conclusions.

3.2 Mathematical Preliminaries

Our topology optimization algorithm becomes mesh independent by being formulated in the inf-dim L_2 space, using concepts from functional analysis. It is then discretized in a consistent manner using the finite element space of the discretized design field, i.e. the element-wise uniform volume fraction field. Notably, the norms in the optimization algorithm that check for convergence are discretized in this finite element space.

To illustrate the proper discretization, consider the unconstrained minimization problem

$$\min_{\nu \in V} \quad \theta(\nu)\,, \tag{3.1}$$

with the functional

$$\theta : V \mapsto \mathbb{R}\,, \tag{3.2}$$

where $\nu : V \mapsto \mathbb{R}$ is our volume fraction design field that belongs to V, a Hilbert space on domain D, equipped with an inner product $(\cdot,\cdot)_V$, which induces the primal norm $\|\cdot\|_V$. For our topology optimization, $V = L_2(D)$; it is equipped with the norm

$$\|\nu\|_{L_2} = \sqrt{(\nu,\nu)_{L_2}} = \left(\int_D \nu^2 \; dV\right)^{1/2} . \tag{3.3}$$

The space of all bounded linear functionals that maps V to $\mathbb{R}$ is the dual spaces V^*, it is a subset of $\mathscr{L}(V,\mathbb{R})$, the space of linear operators from V to $\mathbb{R}$, i.e. $V^* \subset \mathscr{L}(V, \mathbb{R})$. Both the primal $\|\cdot\|_V$ and dual $\|\cdot\|_{V^*}$ norms can be used to check for convergence in optimization algorithms.

To formulate optimization algorithms on the function space V, we need the Riesz map from the Riesz representation theorem: Let V be a Hilbert space with inner product $(\cdot,\cdot)_V$ and dual space V^*. For every $\varphi \in V^*$ there is a unique element $u \in V$ such that $\varphi(v) = (u, v)_V$ for all $v \in V$. This one-to-one map is the Riesz map $\Phi : V \mapsto V^*$ defined such that $\Phi(u) = \varphi$.

The discretization of the primal and dual spaces follows from [82]. We approximate the volume fraction field $\nu \in V$ with $\nu_h \in V_h$, where V_h is the span of basis functions $\mathscr{P} = \{\phi_1, ..., \phi_n\}$, $\phi_i \in V$ and n is the dimension of V_h. Our approximation now reads

$$\nu_h(\mathbf{x}) = \sum_{i=1}^{n} \nu_i \phi_i(\mathbf{x}) = \boldsymbol{\nu}^T \boldsymbol{\phi}(\mathbf{x}) \,. \tag{3.4}$$

Replacing ν with ν_h and similarly $\iota \in V$ with $\iota_h \in V_h$ in the inner product definition yields

$$\begin{aligned}(\nu_h, \iota_h)_{V_h} &= \int_D (\boldsymbol{\nu}^T \boldsymbol{\phi})(\boldsymbol{\iota}^T \boldsymbol{\phi}) dV \\ &= \boldsymbol{\nu}^T \int_D \boldsymbol{\phi}\boldsymbol{\phi}^T \; dV \; \boldsymbol{\iota} \\ &= \boldsymbol{\nu}^T \mathbf{M} \boldsymbol{\iota} \,,\end{aligned} \tag{3.5}$$

where

$$\mathbf{M} = \int_D \boldsymbol{\phi}\boldsymbol{\phi}^T \; dV \tag{3.6}$$

is the mass matrix that reflects the mesh discretization. By construction, $\mathbf{M}$ is symmetric and invertible.

The discretized design field ν_h is in the function space $V_h = (\mathbb{R}^n, (\cdot,\cdot)_{\mathbf{M}})$, i.e. vectors of dimension n with an $\mathbf{M}$ inner product. This inner product induces the norm $\|\nu_h\|_{V_h} = \|\boldsymbol{\nu}\|_{\mathbf{M}} = (\boldsymbol{\nu}^T \mathbf{M} \boldsymbol{\nu})^{1/2}$.

Clearly the L_2 norm $\|\nu_h\|_{L_2} = (\boldsymbol{\nu}^T \mathbf{M} \boldsymbol{\nu})^{1/2}$ differs from the ℓ_2 norm $\|\nu_h\|_{\ell_2} = (\boldsymbol{\nu}^T \boldsymbol{\nu})^{1/2}$. In topology optimization, ν_h is usually discretized via piecewise uniform functions over the individual elements so $\mathbf{M}$ is a diagonal matrix whose entries are the volumes (areas) in 3D (2D) of each element in the mesh. So if the mesh is uniform, $\mathbf{M}$ is simply the identity matrix times the element volume (area) V_e and hence $\|\nu_h\|_{L_2} = V_e \|\nu_h\|_{\ell_2}$.

The basis $\mathscr{P}$ induces a unique dual basis $\mathscr{P}^* = \{\phi_1^*, ..., \phi_n^*\}$ for V_h^* defined such that $\phi_i^* \in V^*$ and $\phi_i^*(\phi_j) = \delta_{ij} \; \forall \; i, j = 1, ...n$. This $\mathscr{P}^*$ basis is used to discretize any $F \in V^*$ as $F_h \in V_h^*$ such that for all

$\iota_h \in V_h$

$$F_h(\iota_h) = \sum_{i=1}^{n} F_i \phi_i^*(\iota_h)\,. \tag{3.7}$$

In this way, $F_h(\iota_h)$ is computed as

$$\begin{aligned} F_h(\iota_h) &= \sum_{i=1}^{n} F_i \phi_i^* \left(\sum_{j=1}^{n} \iota_j \phi_j \right) \,, \\ &= \sum_{i=1}^{n} F_j \sum_{j=1}^{n} \iota_j \phi_i^*(\phi_j)\,, \\ &= \sum_{i=1}^{n} F_i \iota_i\,, \\ &= \mathbf{F}^T \boldsymbol{\iota}\,, \end{aligned} \tag{3.8}$$

where we used the orthonormal property between the bases $\mathscr{P}$ and $\mathscr{P}^*$ and the linearity of F. From the Riesz representation theorem, there exists $\nu_h \in V_h$ such that $\Phi(\nu_h) = F_h$ or $\Phi^{-1}(F_h) = \nu_h$ assuming Φ is invertible, where $\Phi^{-1} : V^* \mapsto V$. Therefore,

$$\begin{aligned} F_h(\iota_h) &= (\overbrace{\Phi^{-1}(F_h)}^{\nu_h}, \iota_h)_V \,, \\ &= (\nu_h, \iota_h)_V \,, \\ &= \boldsymbol{\nu}^T \mathbf{M} \boldsymbol{\iota}\,. \end{aligned} \tag{3.9}$$

From Equations (3.8) and (3.9) we can see that

$$\mathbf{F}^T \boldsymbol{\iota} = \boldsymbol{\nu}^T \mathbf{M} \boldsymbol{\iota}\,. \tag{3.10}$$

Therefore,

$$\mathbf{F} = \mathbf{M} \boldsymbol{\nu}\,, \tag{3.11}$$

and the discrete Riesz map and its inverse are defined such that $\Phi_h(\boldsymbol{\nu}) = \mathbf{M}\boldsymbol{\nu}$ and

$$\Phi_h^{-1}(\mathbf{F}) = \mathbf{M}^{-1}\mathbf{F}\,. \tag{3.12}$$

Considering the isometry between the spaces V_h and V_h^* and the Riesz map, we can now define and calculate the norm of an object $F_h \in V_h^*$ as

$$\begin{aligned} \|F_h\|_{V_h^*} &= \left\|\Phi^{-1}(F_h)\right\|_{V_h} , \\ &= \left\|\mathbf{M}^{-1}\mathbf{F}\right\|_{\mathbf{M}} , \\ &= \sqrt{\mathbf{F}^T\mathbf{M}^{-1}\mathbf{M}\mathbf{M}^{-1}\mathbf{F}} , \\ &= \|\mathbf{F}\|_{\mathbf{M}^{-1}} , \end{aligned} \tag{3.13}$$

where we used the definition of the discrete inner product in Equation (3.5).

In this work, we use the Fréchet derivative $D\theta(\nu) \in V^*$ of the function $\theta : V \mapsto \mathbb{R}$ at ν, which if it exists, is defined such that

$$\theta(\nu + h) - \theta(\nu) - D\theta(\nu)[h] = \mathcal{O}(\|h\|_V) \tag{3.14}$$

for all $h \in V$. By definition $D\theta(\nu) \in V^*$, and hence the Riesz representation theorem tells us there is an object in V that we will denominate $\nabla\theta(\nu) \in V$, i.e. the gradient of θ at ν such that $D\theta(\nu)[h] = (\nabla\theta(\nu), h)_V$ for all $h \in V$. Using the Riesz map, $\Phi(\nabla\theta(\nu)) = D\theta(\nu)$ and because the Riesz map depends on the inner product $(\cdot,\cdot)_V$, so does $\nabla\theta(\nu) \in V$. This is crucial in our optimization algorithm as the inner product depends on the mesh discretization, notably $\boldsymbol{\nabla\theta} = \mathbf{M}^{-1}\boldsymbol{D\theta}$, where $\boldsymbol{\nabla\theta}$ and $\boldsymbol{D\theta}$ are the discrete counterparts of $\nabla\theta(\nu)$ and $D\theta(\nu)$.

We are now in position to show how these functional analysis concepts apply to optimization algorithms in the inf-dim space. We start by examining the most basic optimization algorithm for the solution of the simple unconstrained minimization problem of (3.1). i.e, the steepest descent algorithm, for which the iterate $\nu^{(k)}$ is updated as

$$\nu^{(k+1)} = \nu^{(k)} - \gamma\nabla\theta(\nu^{(k)}) . \tag{3.15}$$

where $\gamma \geq 0$ is the step length. It seems intuitive that ν and $\nabla\theta(\nu)$ must be in the same function space since they are added together. This motivates us to use the gradient $\nabla\theta(\nu)$ and not the derivative $D\theta(\nu)$ in (3.15), which is contrary to most topology optimization algorithms. For a uniform mesh $\mathbf{M} = V_e\mathbf{I}$ so $\boldsymbol{\nabla\theta} = \frac{1}{V_e}\mathbf{I}\boldsymbol{D\theta}$ are parallel and there is no difference in the search direction. In general, however, the discretized, Equation (3.15) becomes

$$\begin{aligned} \boldsymbol{\nu}^{(k+1)} &= \boldsymbol{\nu}^{(k)} - \gamma\boldsymbol{\nabla\theta} . \\ &= \boldsymbol{\nu}^{(k)} - \gamma\mathbf{M}^{-1}\boldsymbol{D\theta} . \end{aligned} \tag{3.16}$$

When calculating the norm to check for convergence, we use $\|\nabla\theta(\nu)\|_V$, which upon discretization is $\|\boldsymbol{\nabla\theta}\|_{\mathbf{M}}$.

The second algorithm uses Newton's method, wherein we iterate to find ν such that

$$D\theta(\nu)[\delta\nu] = 0 \;\forall\delta\nu \in V. \tag{3.17}$$

To do so, we linearize around $\nu^{(k)}$ and solve for the update $\Delta\nu^{(k)}$ via

$$D^2\theta(\nu^{(k)})[\Delta\nu^{(k)}, \delta\nu] = -D\theta(\nu^{(k)})[\delta\nu] \quad \forall\delta\nu \in V\,, \tag{3.18}$$

where $D^2\theta(\nu^{(k)})[\cdot,\cdot] : V \mapsto \mathscr{L}(V \times V, \mathbb{R})$ is the Hessian, i.e. second derivative of θ at $\nu^{(k)}$; it is a bilinear map from $(V \times V)$ to $\mathbb{R}$. The difference here is that we need to feed the algorithm with the derivative $D\theta(\nu)$ (and the Hessian $D^2\theta(\nu)$) and not the gradient $\nabla\theta(\nu)$ as in the steepest descent algorithm. Upon discretization, Equation (3.18) becomes

$$\boldsymbol{D^2\theta\Delta\nu}^{(k)} = -\boldsymbol{D\theta} \tag{3.19}$$

When calculating the norm to check for convergence, we use $\|D\theta(\nu)\|_{V^*}$, whose discretization is $\|\boldsymbol{D\theta}\|_{\mathbf{M}^{-1}}$

We showcase the difference between $\nabla\theta(\nu)$ and $D\theta(\nu)$ with the following convex unconstrained optimization problem

$$\min_{\nu\in V} \theta(\nu) = \frac{1}{2}\langle\nu,\nu\rangle_V - \langle\nu, b\rangle_V \tag{3.20}$$

where b is a given function on V. The solution $\nu^* = b$ is trivially calculated by the stationarity condition $D\theta(\nu)$

$$\begin{aligned} D\theta(\nu)[v] = 0 &= \langle\nabla\theta, v\rangle_V \\ &= \langle\nu - b, v\rangle_V \end{aligned} \tag{3.21}$$

We proceed to discretize the function θ with piecewise uniform elements, resulting in the expression

$$\theta(\boldsymbol{\nu}) = \frac{1}{2}\boldsymbol{\nu}^T\mathbf{M}\boldsymbol{\nu} - \boldsymbol{\nu}^T\mathbf{M}\mathbf{b} \tag{3.22}$$

whose derivative is

$$\boldsymbol{D\theta} = \mathbf{M}(\boldsymbol{\nu} - \mathbf{b})\,. \tag{3.23}$$

Note that Equation (3.23) is the discretization of $D\theta(\nu)$ in Equation (3.21). Applying the steepest descent algorithm yields

$$\begin{aligned}\boldsymbol{\nu}^{(k+1)} &= \boldsymbol{\nu}^{(k)} - \gamma \boldsymbol{D\theta} \\ &= \boldsymbol{\nu}^{(k)} - \gamma \mathbf{M}(\boldsymbol{\nu}^{(k)} - \mathbf{b})\end{aligned} \tag{3.24}$$

The optimal step size γ is calculated with the closed-form expression

$$\gamma = \frac{(\boldsymbol{\nu}^{(k)} - \mathbf{b})^T \mathbf{M}\mathbf{M}(\boldsymbol{\nu}^{(k)} - \mathbf{b})}{(\boldsymbol{\nu}^{(k)} - \mathbf{b})^T \mathbf{M}^T\mathbf{M}\mathbf{M}(\boldsymbol{\nu}^{(k)} - \mathbf{b})} \tag{3.25}$$

We use this expression in Equation (3.24) to obtain the fixed point iteration

$$\boldsymbol{\nu}^{(k+1)} = \boldsymbol{\nu}^{(k)} - \frac{(\boldsymbol{\nu}^{(k)} - \mathbf{b})^T \mathbf{M}\mathbf{M}(\boldsymbol{\nu}^{(k)} - \mathbf{b})}{(\boldsymbol{\nu}^{(k)} - \mathbf{b})^T \mathbf{M}^T\mathbf{M}\mathbf{M}(\boldsymbol{\nu}^{(k)} - \mathbf{b})} \mathbf{M}(\boldsymbol{\nu}^{(k)} - \mathbf{b}) \tag{3.26}$$

If $\mathbf{M}$ is a positive definite diagonal matrix whose entries are not equal, as is common in AMR, we do not obtain convergence. Indeed, Table 3.1 denotes the iteration history of Equation (3.26) with ten design variables over a one dimensional mesh with nodes at positions $x = (0.0, 1.0, 3.0, 6.0, 10.0, 15.0, 21.0, 28.0, 36.0, 45.0, 50.0)$ using elementwise uniform interpolation for ν, $b = 9x + 1$ and the initial estimate $\nu^{(0)} = 0$. The lack of convergence is attributed to the fact that we are adding members from different spaces ($\nu \in V$ and $D\theta(\nu) \in V^*$). If we replace $\boldsymbol{D\theta}$ with the gradient $\boldsymbol{\nabla\theta} = \mathbf{M}^{-1}\boldsymbol{D\theta}(\boldsymbol{\nu}) = \boldsymbol{\nu} - \mathbf{b}$ in Equation (3.24) we find $\gamma = 1$ and we converge in one iteration.

Iteration #	Error $\|\boldsymbol{\nu} - \mathbf{b}\|_{\mathbf{M}}$
1	4.933
2	2.165
3	1.3056
30	0.003

Table 3.1: Iteration history for the discrete steepest descent in Equation (3.26).

3.3 GCMMA in function space

We are now motivated to formulate the first-order GCMMA algorithm [90] in the L_2 space. GCMMA [90] and its non globally-convergent version MMA [89] are widely used algorithms in the topology optimization community. Following the implementation given in [88], we highlight the necessary details to obtain a mesh-independent algorithm. To begin, we consider the optimization problem

$$\begin{aligned}
\min_{\nu \in V} \quad & \theta_0(\nu)\,, \\
\text{s.t.} \quad & \theta_i(\nu) \leq 0,\ i = 1, \ldots, m\,, \\
& \nu_{\min} \leq \nu \leq \nu_{\max}\,.
\end{aligned} \tag{3.27}$$

First, the artificial optimization variables $\mathbf{y} = (y_1, ..., y_m)$ and z are added to make certain subclasses of problems, like least squares or minmax problems, easier to formulate, i.e.

$$\begin{aligned}
\min_{\substack{\nu \in V \\ \boldsymbol{y} \in \mathbb{R}^m \\ z \in \mathbb{R}}} \quad & \theta_0(\nu) + a_0 z + \sum_{i=1}^{m} \left(c_i y_i + \frac{1}{2} d_i y_i^2 \right)\,, \\
\text{s.t.} \quad & \theta_i(\nu) - a_i z - y_i \leq 0,\ i = 1, \ldots, m\,, \\
& \nu_{\min} \leq \nu \leq \nu_{\max}\,, \\
& \boldsymbol{y} \geq 0\,, \\
& z \geq 0\,.
\end{aligned} \tag{3.28}$$

where a_0, a_i, c_i and d_i are real numbers which satisfy $a_0 > 0, a_i \geq 0, c_i \geq 0, d_i \geq 0$ and $c_i + d_i > 0$ for all i, and also $a_i c_i > a_0$ for all i [90]. Note that we recover the original algorithm for $z = 0$ and $\boldsymbol{y} = 0$.

For each optimization iteration k, we solve the following convex approximate subproblem that is based on Equation (3.28), the current iterate $(\nu^{(k)}, \boldsymbol{y}^{(k)}, z^{(k)})$, the cost and constraint functions values and their derivatives. Ultimately, we iterate by solving

$$\begin{aligned}
(\nu^{(k+1)}, \boldsymbol{y}^{(k+1)}, z^{(k+1)}) = \operatorname*{arg\,min}_{\substack{\nu \in V \\ \boldsymbol{y} \in \mathbb{R}^m \\ z \in \mathbb{R}}} \quad & \tilde{\theta}_0(\nu) + a_0 z + \sum_{i=1}^{m} \left(c_i y_i + \frac{1}{2} d_i y_i^2 \right)\,, \\
\text{s.t.} \quad & \tilde{\theta}_i(\nu) - a_i z - y_i \leq 0,\ i = 1, \ldots, m.\,, \\
& \alpha \leq \nu \leq \beta\,, \\
& \boldsymbol{y} \geq 0\,, \\
& z \geq 0\,.
\end{aligned} \tag{3.29}$$

where the newly introduced functions $\tilde{\theta}_0$ and $\tilde{\theta}_i$ and bounds α and β are defined momentarily.

In our formulation of the above subproblem, we replace the summations of the approximating functionals

$\tilde{\theta}_i$ in [88] with integrals over the domain.

$$\tilde{\theta}_i(\nu) = \int_D \left(\frac{p_i}{U^{(k)} - \nu} + \frac{q_i}{\nu - L^{(k)}} \right) \, dV + r_i, \; i = 0, 1, \ldots, m \,, \tag{3.30}$$

$$r_i = \theta_i(\nu^{(k)}) - \int_D \left(\frac{p_i}{U^{(k)} - \nu^{(k)}} + \frac{q_i}{\nu^{(k)} - L^{(k)}} \right) \, dV \,, \tag{3.31}$$

where

$$p_i = (U^{(k)} - \nu^{(k)})^2 \left(1.001 \left(\nabla\theta_i(\nu^{(k)}) \right)^+ + 0.001 \left(\nabla\theta_i(\nu^{(k)}) \right)^- + \frac{\rho_i^{(k,j)}}{\nu_{\max} - \nu_{\min}} \right) \,, \tag{3.32}$$

$$q_i = (\nu^{(k)} - L^{(k)})^2 \left(0.001 \left(\nabla\theta_i(\nu^{(k)}) \right)^+ + 1.001 \left(\nabla\theta_i(\nu^{(k)}) \right)^- + \frac{\rho_i^{(k,j)}}{\nu_{\max} - \nu_{\min}} \right) \,. \tag{3.33}$$

and $U^{(k)}$ and $L^{(k)}$ are the soon to be defined moving upper and lower asymptotes. To ensure the subproblem is convex, we use the ramp like functions $(a)^+ = \max(0, a)$, and $(a)^- = \max(0, -a)$. The bounds (now fields in V) α and β are taken as

$$\begin{aligned} \alpha &= \max\{\nu_{\min}, L^{(k)} + 0.1(\nu^{(k)} - L^{(k)}), \nu^{(k)} - 0.5(\nu_{\max} - \nu_{\min})\} \\ \beta &= \min\{\nu_{\max}, U^{(k)} - 0.1(U^{(k)} - \nu^{(k)}), \nu^{(k)} + 0.5(\nu_{\max} - \nu_{\min})\} \end{aligned} \tag{3.34}$$

The GCMMA differs from the MMA in its attempt to achieve global convergence by controlling the parameter $\rho_i^{(k,j)}$ (which in the MMA is a fixed small value, usually lower than 10^{-5}) in the convex approximation of Equations (3.32) and (3.33). Here, the added superscript j corresponds to the inner iteration within the GCMMA. For the initial $j = 0$ inner iteration, the solution $(\nu^{(k,0)}, \boldsymbol{y}^{(k,0)}, z^{(k,0)})$ of the Equation (3.29) subproblem is accepted if

$$\tilde{\theta}_i(\nu^{(k,j)}) \geq \theta_i(\nu^{(k,j)}) \; \forall i = 0, ..., m \,, \tag{3.35}$$

whereupon the outer iteration $k{+}1$ commences with the initial iterate $(\nu^{(k+1)}, \boldsymbol{y}^{(k+1)}, z^{(k+1)}) = (\nu^{(k,0)}, \boldsymbol{y}^{(k,0)}, z^{(k,0)})$. Otherwise, the $j+1$ subproblem (3.29) is solved with a more conservative convex approximation by replacing $\rho_i^{(k,j)}$ with $\rho_i^{(k,j+1)} > \rho_i^{(k,j)}$ and the Equation (3.35) inequality is rexamined. If Equation (3.35) is satisfied, we begin the outer iteration $k{+}1$ with the initial iterate $(\nu^{(k+1,0)}, \boldsymbol{y}^{(k+1,0)}, z^{(k+1,0)}) = (\nu^{(k,j+1)}, \boldsymbol{y}^{(k,j+1)}, z^{(k,j+1)})$, otherwise, the inner $j + 2$ subproblem is solved and so on, cf. Figure 3.1.

The parameters $\rho_i^{(k,j)}$ are calculated following [90], but with integrals replacing summations. For sub-

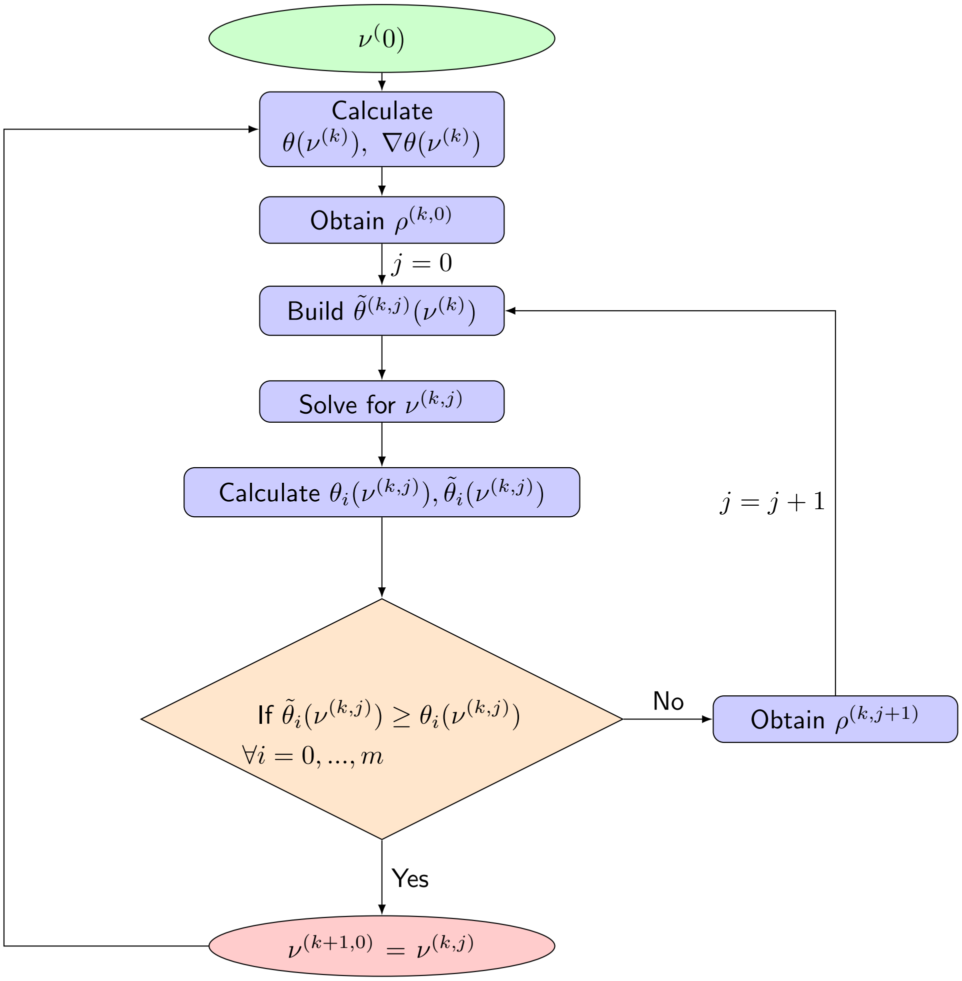

$\nu^(0)$
Calculate
$\theta(\nu^{(k)}),\ \nabla\theta(\nu^{(k)})$
Obtain $\rho^{(k,0)}$
$j = 0$
Build $\tilde{\theta}^{(k,j)}(\nu^{(k)})$
Solve for $\nu^{(k,j)}$
Calculate $\theta_i(\nu^{(k,j)}), \tilde{\theta}_i(\nu^{(k,j)})$
$j = j + 1$
If $\tilde{\theta}_i(\nu^{(k,j)}) \geq \theta_i(\nu^{(k,j)})$
$\forall i = 0, ..., m$
No
Obtain $\rho^{(k,j+1)}$
Yes
$\nu^{(k+1,0)} = \nu^{(k,j)}$

Figure 3.1: GCMMA algorithm.

problem $(k,0)$

$$\rho_i^{(k,0)} = \frac{0.1}{\hat{V}} \int_D |\nabla \theta_i(\nu^{(k,0)})| \, (\nu_{\max} - \nu_{\min}) \ dV \text{ for } i = 0, 1, \ldots, m \,, \tag{3.36}$$

where $\hat{V}$ is the volume (area) of D. For the subsequent $(k, j+1)$ subproblems

$$\begin{aligned} \rho_i^{(k,j+1)} &= \min\left\{1.1\left(\rho_i^{(k,j)} + \delta_i^{(k,j)}\right), 10\rho_i^{(k,j)}\right\} & \text{if } \delta_i^{(k,j)} > 0 \,, \\ \rho_i^{(k,j+1)} &= \rho_i^{(k,j)} & \text{if } \delta_i^{(k,j)} \le 0 \,, \end{aligned} \tag{3.37}$$

where

$$\delta_i^{(k,j)} = \frac{\theta_i\left(\nu^{(k,j)}\right) - \tilde{\theta}_i\left(\nu^{(k,j)}\right)}{d\left(\nu^{(k,j)}\right)} \,. \tag{3.38}$$

with

$$d(\nu) = \int_D \frac{\left(U^{(k)} - L^{(k)}\right)\left(\nu - \nu^{(k)}\right)^2}{\left(U^{(k)} - \nu\right)\left(\nu - L^{(k)}\right)(\nu_{\max} - \nu_{\min})} \ dV \,, \tag{3.39}$$

The moving asymptotes fields $L \in V$ and $U \in V$ are updated via heuristic rules. For iterations $k = 1$ and $k = 2$

$$\begin{aligned} L^{(k)} &= \nu^{(k)} - 0.5(\nu_{\max} - \nu_{\min}) \,, \\ U^{(k)} &= \nu^{(k)} + 0.5(\nu_{\max} - \nu_{\min}) \,. \end{aligned} \tag{3.40}$$

For iterations $k \ge 3$,

$$\begin{aligned} L^{(k)} &= \nu^{(k)} - \gamma^{(k)}(\nu^{(k-1)} - L^{(k-1)}) \,, \\ U^{(k)} &= \nu^{(k)} + \gamma^{(k)}(U^{(k-1)} - \nu^{(k-1)}) \,. \end{aligned} \tag{3.41}$$

The field γ that appears in Equations (3.41) is determined by the variation of ν in the last three outer iterations. When there is no oscillation in ν, we reduce the convexity by pushing the asymptotes further apart by choosing a larger γ to accelerate convergence. Otherwise, we use a smaller value to move the asymptotes closer together. Specifically, we assign

$$\gamma^{(k)} = \begin{cases} 0.7 & \text{if } (\nu^{(k)} - \nu^{(k-1)})(\nu^{(k-1)} - \nu^{(k-2)}) < 0 \,, \\ 1.2 & \text{if } (\nu^{(k)} - \nu^{(k-1)})(\nu^{(k-1)} - \nu^{(k-2)}) > 0 \,, \\ 1 & \text{if } (\nu^{(k)} - \nu^{(k-1)})(\nu^{(k-1)} - \nu^{(k-2)}) = 0 \,, \end{cases} \tag{3.42}$$

subject to the inequalities

$$\begin{aligned}
L^{(k)} &\leq \nu^{(k)} - 0.01(\nu_{\max} - \nu_{\min})\,,\\
L^{(k)} &\geq \nu^{(k)} - 10(\nu_{\max} - \nu_{\min})\,,\\
U^{(k)} &\geq \nu^{(k)} + 0.01(\nu_{\max} - \nu_{\min})\,,\\
U^{(k)} &\leq \nu^{(k)} + 10(\nu_{\max} - \nu_{\min})\,.
\end{aligned} \tag{3.43}$$

The MMA subproblem (3.29) is solved using a primal-dual interior-point algorithm as in [88], where a sequence of relaxed KKT conditions are solved using the Newton-Raphson method. Omitting the iteration indices (k, j) for clarity we solve

$$\begin{aligned}
\min_{\substack{\nu\in V\\ \boldsymbol{y}\in\mathbb{R}^m\\ z\in\mathbb{R}\\ s\in\mathbb{R}^m}} \quad & \int_D \left(\frac{p_o}{U-\nu} + \frac{q_o}{\nu - L}\right) dV + a_0 z + \sum_{i=1}^{m}\left(c_i y_i + \frac{1}{2} d_i y_i^2\right) + r_0\,,\\
\text{s.t.} \quad & \int_D \left(\frac{p_i}{U-\nu} + \frac{q_i}{\nu - L}\right) dV - a_i z - y_i + r_i + s_i = 0,\ i = 1, \ldots, m\,,\\
& \alpha \leq \nu \leq \beta\,,\\
& \boldsymbol{y} \geq 0\,,\\
& z \geq 0\,,\\
& \boldsymbol{s} \geq 0\,,
\end{aligned} \tag{3.44}$$

where we have introduced the slack variable $\boldsymbol{s} \in \mathbb{R}^m$ to transform the $\theta_i(\nu) \leq 0$ inequality constraints into m equality constraints.

To solve the above, the Lagrangian corresponding to Equations (3.44) is formed by introducing the log-barrier terms for the inequality constraints.

$$\begin{aligned}
\mathcal{L}(\nu, \mathbf{y}, z, \lambda, \mathbf{s}) = & \int_D \left(\frac{p_o}{U-\nu} + \frac{q_o}{\nu - L}\right) dV + a_0 z + \sum_{i=1}^{m}\left(c_i y_i + \frac{1}{2} d_i y_i^2\right) + r_0\\
& + \sum_{i=1}^{m} \lambda_i \left(\int_D \left(\frac{p_i}{U-\nu} + \frac{q_i}{\nu - L}\right) dV - a_i z - y_i + r_i + s_i\right)\\
& - \epsilon \int_D \ln(\nu - \alpha)\ dV - \epsilon \int_D \ln(\beta - \nu) dV - \epsilon \sum_{i=1}^{m} \ln y_i - \epsilon \ln(z) - \epsilon \sum_{i=1}^{m} \ln s_i\,.
\end{aligned} \tag{3.45}$$

Stationarity of the Lagrangian is then solved sequentially for different values of the barrier parameter ϵ, starting at $\epsilon = 1$ and decreasing by a factor of 0.1 until $\epsilon \leq 10^{-5}$ Convergence for a given ϵ is deemed when the residual of the derivative of the Equation (3.45) Lagrangian is less than 0.9ϵ. Note that the bound constraints on the volume fraction field $\nu \in V$ are enforced via integrals, as it is known that the pointwise bounds $\alpha \leq \nu \leq \beta$ for all $\mathbf{x} \in D$ will be satisfied by the integral expressions provided we start from a feasible design.

To keep the notation clean, we group the variables as $\chi = (\nu, \mathbf{y}, z, \boldsymbol{\lambda}, \mathbf{s})$ and express the function $\psi_j(\nu) : V \mapsto V,\ j = 0, \ldots, m$ as

$$\psi_j(\nu) = \frac{p_j}{U - \nu} + \frac{q_j}{\nu - L},\ j = 0, \ldots, m\,, \tag{3.46}$$

In this way, the stationarity conditions on the Lagrangian become

$$\begin{aligned}
D_\nu \mathcal{L}(\chi)[\delta\nu] &= \int_D \left(D\psi_0(\nu)[\delta\nu] + \sum_{i=1}^{m} \lambda_i D\psi_i(\nu)[\delta\nu] - \frac{\epsilon\delta\nu}{\nu - \alpha} + \frac{\epsilon\delta\nu}{\beta - \nu} \right)\, dV &&= 0\ \ \forall\delta\nu \in V\,,\\
D_{y_i} \mathcal{L}(\chi)[\delta y_i] &= \left(c_i + d_i y_i - \lambda_i - \frac{\epsilon}{y_i} \right) \delta y_i &&= 0\ \ \forall\delta y_i\ i = 1...m\,,\\
D_z \mathcal{L}(\chi)[\delta z] &= (a_0 - \sum_{i=1}^{m} \lambda_i a_i - \frac{\epsilon}{z})\delta z &&= 0\ \ \forall\delta z\,,\\
D_{\lambda_i} \mathcal{L}(\chi)[\delta\lambda_i] &= \left(\int_D \psi_i(\nu)\ dV - a_i z - y_i + r_i + s_i \right) \delta\lambda_i &&= 0\ \ \forall\delta\lambda_i\ i = 1...m\,,\\
D_{s_i} \mathcal{L}(\chi)[\delta s_i] &= (\lambda_i - \frac{\epsilon}{s_i})\delta s_i &&= 0\ \ \forall\delta s_i\ i = 1...m\,,
\end{aligned} \tag{3.47}$$

Equation (3.47).2 is fulfilled for any δy_i, therefore the term in the parenthesis equals zero and similarly for Equations (3.47).3-(3.47).5. From the last Equation (3.47).5, we obtain $\lambda_i s_i = \epsilon$ for $i = 1...m$.

We next introduce the dual variables

$$\begin{aligned}
\varepsilon &= \frac{\epsilon}{\nu - \alpha}\,,\\
\eta &= \frac{\epsilon}{\beta - \nu}\,,\\
\mu_i &= \frac{\epsilon}{y_i}\,,\ i = 1...m\,,\\
\zeta &= \frac{\epsilon}{z}\,,
\end{aligned} \tag{3.48}$$

where $\varepsilon, \eta \in V$ and $\mu_i, \zeta \in \mathbb{R}$ to generate the following system of nonlinear equations

$$\begin{aligned}
D_\nu \mathcal{L}(\chi)[\delta\nu] &= \int_D \left(D\psi_0(\nu)[\delta\nu] + \sum_{i=1}^{m} \lambda_i D\psi_i(\nu)[\delta\nu] - \varepsilon\delta\nu + \eta\delta\nu \right) \, dV && = 0 \ \ \forall \delta\nu \in V \,, \\
D_{y_i} \mathcal{L}(\chi) &= c_i + d_i y_i - \lambda_i - \mu_i && = 0 \ i = 1...m \,, \\
D_z \mathcal{L}(\chi) &= a_0 - \sum_{i=1}^{m} \lambda_i a_i - \zeta && = 0 \,, \\
D_{\lambda_i} \mathcal{L}(\chi) &= \int_D \psi_i(\nu) \, dV - a_i z - y_i + r_i + s_i && = 0 \ i = 1...m \,, \\
\lambda_i s_i &= \epsilon && \quad i = 1...m \,, \\
(\nu - \alpha)\varepsilon &= \epsilon \,, \\
(\beta - \nu)\eta &= \epsilon \,, \\
\mu_i y_i &= \epsilon && \quad i = 1...m \,, \\
\zeta z &= \epsilon \,.
\end{aligned} \tag{3.49}$$

We express the above in a more compact form $F(\upsilon) = 0$, where $\upsilon = (\nu, \mathbf{y}, z, \boldsymbol{\lambda}, \mathbf{s}, \varepsilon, \eta, \boldsymbol{\mu}, \zeta)$ and $F : V \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^m \times V \times V \times \mathbb{R}^m \times \mathbb{R} \mapsto V^* \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^m \times V \times V \times \mathbb{R}^m \times \mathbb{R}$ is defined as

$$F(\upsilon) = \begin{bmatrix} \int_D \left(D\psi_0(\nu)[\delta\nu] + \sum_{i=1}^{m} \lambda_i D\psi_i(\nu)[\delta\nu] - \varepsilon\delta\nu + \eta\delta\nu \right) \, dV \\ c_i + d_i y_i - \lambda_i - \mu_i \\ a_0 - \sum_{i=1}^{m} \lambda_i a_i - \zeta \\ \int_D \psi_i(\nu) \, dV - a_i z - y_i + r_i + s_i \\ \lambda_i s_i - \epsilon \\ \int_D ((\nu - \alpha)\varepsilon - \epsilon)\delta\varepsilon \, dV \\ \int_D ((\beta - \nu)\eta - \epsilon)\delta\eta \, dV \\ \mu_i y_i - \epsilon \\ \zeta z - \epsilon \end{bmatrix} = \begin{bmatrix} \delta_\nu \\ \delta_{y_i} \\ \delta_z \\ \delta_{\lambda_i} \\ \delta_{s_i} \\ \delta_\varepsilon \\ \delta_\eta \\ \delta_{\mu_i} \\ \delta_\zeta \end{bmatrix} \tag{3.50}$$

for all $\delta\nu \in V$, $\delta\varepsilon \in V$ and $\delta\eta \in V$. As seen above, for discretization purposes, we enforce equations (3.49).6 and (3.49).7 weakly.

We solve $F(\upsilon) = 0$ (for all $\delta\upsilon$) via Newton-Raphson by linearizing around $\upsilon^{(k)}$ and requiring $\Delta\upsilon^{(k)}$ to satisfy

$$DF(\upsilon^{(k)})\Delta\upsilon^{(k)} = -F(\upsilon^{(k)}) \,, \tag{3.51}$$

which we expand into

$$\begin{bmatrix} \int_D \left(D^2\psi_0(\nu)[\delta\nu, \Delta\nu] + \sum_{i=1}^m \lambda_i D^2\psi_i(\nu)[\delta\nu, \Delta\nu] + \sum_{i=1}^m D\psi_i(\nu)[\delta\nu]\Delta\lambda_i - \delta\nu\Delta\varepsilon + \delta\nu\Delta\eta\right)\ dV \\ d_i\Delta y_i - \Delta\lambda_i - \Delta\mu_i \\ -\sum_{i=1}^m a_i\Delta\lambda_i - \Delta\zeta \\ \int_D D\psi_i(\nu)[\Delta\nu]\ dV - a_i\Delta z - \Delta y_i + \Delta s_i \\ \Delta\lambda_i s_i + \lambda_i\Delta s_i \\ \int_D \left(\Delta\nu\varepsilon + (\nu-\alpha)\Delta\varepsilon\right)\delta\varepsilon\ dV \\ \int_D \left(-\Delta\nu\eta + (\beta-\nu)\Delta\eta\right)\delta\eta\ dV \\ \Delta\mu_i y_i + \mu_i\Delta y_i \\ \Delta\zeta z + \zeta\Delta z \end{bmatrix} = \begin{bmatrix} -\delta_\nu \\ -\delta_{y_i} \\ -\delta_z \\ -\delta_{\lambda_i} \\ -\delta_{s_i} \\ -\delta_\varepsilon \\ -\delta_\eta \\ -\delta_{\mu_i} \\ -\delta_\zeta \end{bmatrix} \tag{3.52}$$

for all $\delta\nu, \delta\varepsilon$ and $\delta\eta$ in V.

We proceed to discretize the fields ν, ε and η in V to be piecewise uniform over the finite elements as in Equation (3.4). So that, e.g.

$$\begin{aligned} \Delta\nu(\mathbf{x}) &= \boldsymbol{\Delta\nu}^T\boldsymbol{\phi}(\mathbf{x})\,, \\ \delta\nu(\mathbf{x}) &= \boldsymbol{\delta\nu}^T\boldsymbol{\phi}(\mathbf{x})\,. \end{aligned} \tag{3.53}$$

Substituting these expressions in Equations (3.50) and (3.52) and using the arbitrariness of $\delta\nu, \delta\varepsilon$ and $\delta\eta$ yields the discretized residuals

$$\begin{aligned} \boldsymbol{\delta_\nu} &= \int_D \left(D\psi_0(\nu) + \textstyle\sum_{i=1}^m \lambda_i D\psi_i(\nu) - \varepsilon + \eta\right)\boldsymbol{\phi}\ dV\,, \\ \boldsymbol{\delta_\varepsilon} &= (\boldsymbol{\nu} - \boldsymbol{\alpha})\circ\boldsymbol{\varepsilon} - \epsilon\mathbf{1}\,,^{1} \\ \boldsymbol{\delta_\eta} &= (\boldsymbol{\beta} - \boldsymbol{\nu})\circ\boldsymbol{\eta} - \epsilon\mathbf{1}\,, \end{aligned} \tag{3.54}$$

where the operator $\circ$ denotes the Hadamard product, i.e. the component wise multiplication between two

[1] Note that for the choice of $\boldsymbol{\phi}$ being piecewise uniform over the element $\int_D \left(\varepsilon(\nu-\alpha) - \epsilon\right)\boldsymbol{\phi}\ dV = \mathrm{diag}(V_e^1, V_e^2, ...,)\left(\boldsymbol{\varepsilon}\circ(\boldsymbol{\nu} - \boldsymbol{\alpha}) - \epsilon\mathbf{1}\right)$. Equation (3.54).2 follows since $\mathrm{diag}(V_e^1, V_e^2, ...,)$ is invertible.

vectors and $\mathbf{1} = (1, 1, ..., 1)$. We also define

$$
\begin{aligned}
\boldsymbol{\delta_y} &= \boldsymbol{c} + \boldsymbol{d} \circ \boldsymbol{y} - \boldsymbol{\lambda} - \boldsymbol{\mu} \,, \\
\delta_z &= a_0 - \boldsymbol{\lambda^T a} - \zeta \,, \\
\boldsymbol{\delta_\lambda} &= \left[\int_D \psi_1(\nu) \; dV, ..., \int_D \psi_m(\nu) \; dV \right]^T - \boldsymbol{a}z - \boldsymbol{y} + \boldsymbol{r} + \boldsymbol{s} \,, \\
\boldsymbol{\delta_s} &= \boldsymbol{\lambda} \circ \boldsymbol{s} - \epsilon \mathbf{1} \,, \\
\boldsymbol{\delta_\mu} &= \boldsymbol{\mu} \circ \boldsymbol{y} - \epsilon \mathbf{1} \,, \\
\delta_\zeta &= \zeta z - \epsilon \,.
\end{aligned}
\tag{3.55}
$$

In this way, the discretized update equations read

$$
\begin{aligned}
\boldsymbol{\Psi \Delta \nu} + \mathbf{G} \boldsymbol{\Delta\lambda} - \mathrm{M}(\boldsymbol{\Delta\varepsilon} - \boldsymbol{\Delta\eta}) &= -\boldsymbol{\delta_\nu} \,, \\
\langle \boldsymbol{d} \rangle \boldsymbol{\Delta y} - \boldsymbol{\Delta\lambda} - \boldsymbol{\Delta\mu} &= -\boldsymbol{\delta_y} \,, \\
-\mathbf{a}^T \boldsymbol{\Delta\lambda} - \Delta\zeta &= -\delta_z \,, \\
\mathbf{G}^T \boldsymbol{\Delta\nu} - \boldsymbol{\Delta y} - \boldsymbol{a}\Delta z + \boldsymbol{\Delta s} &= -\boldsymbol{\delta_\lambda} \,, \\
\langle \boldsymbol{s} \rangle \Delta\boldsymbol{\lambda} + \langle \boldsymbol{\lambda} \rangle \Delta\boldsymbol{s} &= -\boldsymbol{\delta_s} \,. \\
\langle \boldsymbol{\varepsilon} \rangle \boldsymbol{\Delta\nu} + \langle \boldsymbol{\nu} - \boldsymbol{\alpha} \rangle \boldsymbol{\Delta\varepsilon} &= -\boldsymbol{\delta_\varepsilon} \,, \\
-\langle \boldsymbol{\eta} \rangle \boldsymbol{\Delta\nu} + \langle \boldsymbol{\beta} - \boldsymbol{\nu} \rangle \boldsymbol{\Delta\eta} &= -\boldsymbol{\delta_\eta} \,, \\
\langle \boldsymbol{\mu} \rangle \Delta\boldsymbol{y} + \langle \boldsymbol{y} \rangle \Delta\boldsymbol{\mu} &= -\boldsymbol{\delta_\mu} \,, \\
\zeta \Delta z + z \Delta\zeta &= -\delta_\zeta \,,
\end{aligned}
\tag{3.56}
$$

where the operator $\langle\cdot\rangle$ is a diagonal matrix, e.g. $\langle\boldsymbol{\nu}\rangle = \mathrm{diag}(\nu_1, \nu_2, ..., \nu_n)$ and $\boldsymbol{\Psi}$ and $\mathbf{G}$ are the block matrices

$$
\begin{aligned}
\boldsymbol{\Psi} &= \int_D \left(D^2\psi_0(\nu) + \sum_{i=1}^{m} \lambda_i D^2 \psi_i(\nu) \right) \boldsymbol{\phi\phi}^T \; dV \,, \\
\mathbf{G} &= \left[\int_D D\psi_1(\nu)\boldsymbol{\phi} \; dV, \int_D D\psi_2(\nu)\boldsymbol{\phi} \; dV, ... \,, \int_D D\psi_m(\nu)\boldsymbol{\phi} \; dV \right] \,.
\end{aligned}
\tag{3.57}
$$

Ultimately, we are left with the update equation

$$\begin{pmatrix} \boldsymbol{\Psi} & & & \mathbf{G} & & -\mathbf{M} & \mathbf{M} & & \\ & \langle \mathbf{d} \rangle & & -\mathbf{I}_m & & & & -\mathbf{I}_m & \\ & & & -\boldsymbol{a}^T & & & & & -1 \\ \mathbf{G}^T & -\mathbf{I}_m & -\boldsymbol{a} & & \mathbf{I}_m & & & & \\ & & & \langle \boldsymbol{s} \rangle & \langle \boldsymbol{\lambda} \rangle & & & & \\ \langle \boldsymbol{\varepsilon} \rangle & & & & & \langle \boldsymbol{\nu} - \boldsymbol{\alpha} \rangle & & & \\ -\langle \boldsymbol{\eta} \rangle & & & & & & \langle \boldsymbol{\beta} - \boldsymbol{\nu} \rangle & & \\ & \langle \boldsymbol{\mu} \rangle & & & & & & \langle \mathbf{y} \rangle & \\ & & \zeta & & & & & & z \end{pmatrix} \begin{pmatrix} \boldsymbol{\Delta\nu} \\ \boldsymbol{\Delta y} \\ \Delta z \\ \boldsymbol{\Delta\lambda} \\ \boldsymbol{\Delta s} \\ \boldsymbol{\Delta\varepsilon} \\ \boldsymbol{\Delta\eta} \\ \boldsymbol{\Delta\mu} \\ \Delta\zeta \end{pmatrix} = \begin{pmatrix} -\boldsymbol{\delta_\nu} \\ -\boldsymbol{\delta_y} \\ -\delta_z \\ -\boldsymbol{\delta_\lambda} \\ -\boldsymbol{\delta_s} \\ -\boldsymbol{\delta_\varepsilon} \\ -\boldsymbol{\delta_\eta} \\ -\boldsymbol{\delta_\mu} \\ -\delta_\zeta \end{pmatrix} , \tag{3.58}$$

where $\mathbf{I}_m$ is the $m \times m$ identity matrix. The pointwise operations in Equations (3.34), (3.40) - (3.43) are carried out on the vector components of their corresponding discretized fields.

As in [88], the linear system of equations in (3.58) can be solved for $\boldsymbol{\Delta\upsilon} = (\boldsymbol{\Delta\nu}, \boldsymbol{\Delta y}, \Delta z, \boldsymbol{\Delta\lambda}, \boldsymbol{\Delta s}, \boldsymbol{\Delta\varepsilon}, \boldsymbol{\Delta\eta}, \boldsymbol{\Delta\mu}, \Delta\zeta)$ quickly given that all the block matrices are diagonal, although this only happens for the piecewise uniform discretization.

We limit the Newton step from violating the inequalities in Equations (3.44) as in [88]. To do this, we first maximize the parameter t such that

$$\begin{aligned} t &\leq 1\,, \\ \nu_l + t\Delta\nu_l - \alpha_l &\geq 0.01\,(\nu_l - \alpha_l) \text{ for all } l = 1, ..., n\,, \\ \beta_l - (\nu_l + t\Delta\nu_l) &\geq 0.01\,(\beta_l - \nu_l) \text{ for all } l = 1, ..., n\,, \\ (\mathbf{y}, z, \boldsymbol{\lambda}, \boldsymbol{s}, \boldsymbol{\varepsilon}, \boldsymbol{\eta}, \boldsymbol{\mu}, \zeta,) + t\,(\boldsymbol{\Delta}\mathbf{y}, \Delta z, \boldsymbol{\Delta\lambda}, \boldsymbol{\Delta s}, \boldsymbol{\Delta\varepsilon}, \boldsymbol{\Delta\eta}, \boldsymbol{\Delta\mu}, \Delta\zeta) &\geq 0.01\,(\mathbf{y}, z, \boldsymbol{\lambda}, \boldsymbol{s}, \boldsymbol{\varepsilon}, \boldsymbol{\eta}, \boldsymbol{\mu}, \zeta)\ . \end{aligned} \tag{3.59}$$

where all inequalities are performed on the vector components and n is the number of mesh elements. As in [88], we assign $\boldsymbol{\upsilon} \leftarrow \boldsymbol{\upsilon} + \tau\boldsymbol{\Delta\upsilon}$ where τ is the largest of $t,\ t/2,\ t/4, ...$ such that

$$\|F(\upsilon + \tau\Delta\upsilon)\|_Q < \|F(\upsilon)\|_Q\ . \tag{3.60}$$

We emphasize that $\|\cdot\|_Q$ is not the ordinary Euclidean norm, but the norm induced by the function space $Q = V^* \times \mathbb{R}^m \times \mathbb{R} \times \mathbb{R}^m \times \mathbb{R}^m \times V \times V \times \mathbb{R}^m \times \mathbb{R}$, i.e.

$$\|F(\upsilon)\|_Q = \sqrt{\|\delta_\nu\|^2_{V_h^*} + \|\delta_y\|^2_{\ell_2} + \|\delta_z\|^2 + \|\delta_\lambda\|^2_{\ell_2} + \|\delta_s\|^2_{\ell_2} + \|\delta_\varepsilon\|^2_{V_h} + \|\delta_\eta\|^2_{V_h} + \|\delta_\mu\|^2_{\ell_2} + \|\delta_\zeta\|^2}\,, \tag{3.61}$$

where, e.g.

$$\begin{aligned} \|\delta_\nu\|_{V_h^*} &= \boldsymbol{\delta}_\nu^T \mathbf{M}^{-1} \boldsymbol{\delta}_\nu\,, \\ \|\delta_\varepsilon\|_{V_h} &= \boldsymbol{\delta}_\varepsilon^T \mathbf{M} \boldsymbol{\delta}_\varepsilon\,. \end{aligned} \tag{3.62}$$

To recover the original optimization algorithm by [88] posed in the Euclidean space ℓ_2, we merely replace $\mathbf{M}$ with the identity matrix in the norm calculations and Equation (3.58).

The discretized $\mathbf{D}\boldsymbol{\theta}$ needs to be passed to the optimization algorithm [78]. However, the GCMMA requires the discretized gradient $\boldsymbol{\nabla}\boldsymbol{\theta}$ cf. Equations (3.32), (3.33) and (3.66).1 which is calculated using the Riesz map (3.12) as

$$\begin{aligned} \boldsymbol{\nabla}\boldsymbol{\theta} &= \Phi_h^{-1}(\mathbf{D}\boldsymbol{\theta})\,, \\ &= \mathbf{M}^{-1}\mathbf{D}\boldsymbol{\theta}\,. \end{aligned} \tag{3.63}$$

The gradient components are used to evaluate Equations (3.32) and (3.33), i.e.

$$\begin{aligned} p_{i,l}^{(k,j)} &= (U_l^{(k)} - \nu_l^{(k,j)})^2 \left(1.001\left(\nabla\theta_{i,l}(\nu^{(k,j)})\right)^+ + 0.001\left(\nabla\theta_{i,l}(\nu^{(k,j)})\right)^- + \frac{\rho_i^{(k,j)}}{\nu_{\max} - \nu_{\min}}\right)\,, \\ q_{i,l}^{(k,j)} &= (\nu_l^{(k)} - L_l^{(k,j)})^2 \left(0.001\left(\nabla\theta_{i,l}(\nu^{(k,j)})\right)^+ + 1.001\left(\nabla\theta_{i,l}(\nu^{(k,j)})\right)^- + \frac{\rho_i^{(k,j)}}{\nu_{\max} - \nu_{\min}}\right)\,. \end{aligned} \tag{3.64}$$

where the subscript l in $p_{i,l}^{(k,j)}$ and $q_{i,l}^{(k,j)}$ corresponds to the mesh element l and in $\nabla\theta_{i,l}$ it corresponds to the component of the gradient $\boldsymbol{\nabla}\boldsymbol{\theta}_i$.

The termination criteria of the original problem (3.27), i.e. not the subproblem (3.29), is derived from its KKT conditions. Notably, here we follow the stopping criteria of [90] and monitor the gradient of the Lagrangian

$$\mathcal{L}(\nu, \lambda, \tau^+, \tau^-) = \theta_0(\nu) + \sum_{i=1}^m \lambda_i \theta_i(\nu) + \int_D \tau^-(\nu_{\min} - \nu)\; dV + \int_D \tau^+(\nu - \nu_{\max})\; dV\,. \tag{3.65}$$

Upon defining $\omega = (\nu, \lambda, \tau^-, \tau^+)$, the KKT conditions read

$$
\begin{aligned}
\nabla_\nu \mathcal{L}(\omega) &= \nabla\theta_0(\nu) + \sum_{i=1}^{m} \lambda_i \nabla\theta_i(\nu) - \tau^- + \tau^+ = 0\,, \\
\theta_i(\nu) &\leq 0 && i = 1, \dots, m\,, \\
\lambda_i \theta_i(\nu) &= 0 && i = 1, \dots, m\,, \\
\lambda_i &\geq 0 && i = 1, \dots, m\,, \\
\nu_{\min} &\leq \nu \leq \nu_{\max}\,, \\
\tau^-(\nu_{\min} - \nu) &= 0\,, \\
\tau^+(\nu - \nu_{\max}) &= 0\,, \\
\tau^+, \tau^- &\geq 0\,.
\end{aligned}
$$

Wherever the bound constraints are active, i.e. wherever $\nu = \nu_{\min}$, or $\nu = \nu_{\max}$, their corresponding Lagrange multipliers are $\tau^- = \nabla\theta_0(\nu) + \sum_{i=1}^{m} \lambda_i \nabla\theta_i(\nu)$ or $\tau^+ = -\left(\nabla\theta_0(\nu) + \sum_{i=1}^{m} \lambda_i \nabla\theta_i(\nu)\right)$ respectively. And since $\tau^- \geq 0$, we have $\nabla\theta_0(\nu) + \sum_{i=1}^{m} \lambda_i \nabla\theta_i(\nu) \geq 0$ and similarly since $\tau^+ \geq 0$ we have $\nabla\theta_0(\nu) + \sum_{i=1}^{m} \lambda_i \nabla\theta_i(\nu) \leq 0$. Using these inequalities in the complementary slackness, Equations (3.66).6 - (3.66).7, transforms them to

$$
\begin{aligned}
\lambda_i (\theta_i(\nu))^- &= 0\,,\ i = 1, \dots, m\,, \\
(\theta_i(\nu))^+ &= 0\,,\ i = 1, \dots, m\,, \\
\lambda_i &\geq 0\,,\ i = 1, \dots, m\,, \\
(\nu_{\min} - \nu) \left(\nabla\theta_0(\nu) + \sum_{i=1}^{m} \lambda_i \nabla\theta_i(\nu) \right)^+ &= 0\,, \\
(\nu_{\max} - \nu) \left(\nabla\theta_0(\nu) + \sum_{i=1}^{m} \lambda_i \nabla\theta_i(\nu) \right)^- &= 0\,, \\
\tau^+, \tau^- &\geq 0
\end{aligned}
\tag{3.66}
$$

where the Lagrange multipliers λ_i are obtained from the solution of the convex approximation problem (3.29). The norm of the KKT conditions is given by the norms of the left hand sides of Equations (3.66).1-(3.66).2 and (3.66).4-(3.66).5 where each norm is taken in its corresponding space, i.e., Equations (3.66).1-(3.66).2 in ℓ_2 and Equations (3.66).4-(3.66).5 in V. The primal-dual algorithm for the subproblem is outlined in Algorithm 2

Algorithm 2 Algorithm outline.

```
**Input** : Starting point for $l = 1$: $\nu^{(l)} = \frac{\alpha+\beta}{2}, y_i^{(l)} = s_i^{(l)} = \lambda_i^{(l)} = 1.0$, $\mu_i^{(l)} = \max\{1.0, 0.5c_i\}$ for $i = 1, ..., m$, $z^{(l)} = \zeta^{(l)} = 1.0$, $\varepsilon^{(l)} = \max\{1.0, \frac{1.0}{\nu^{(l)}-\alpha}\}, \eta^{(l)} = \max\{1.0, \frac{1.0}{\beta-\nu^{(l)}}\}$ and $\epsilon = 1$.
**while** $\epsilon > 10^{-5}$ **do**
    Calculate $\boldsymbol{\Delta \upsilon}$ in Equation (3.58).
    Calculate step length $\tau$ in Equation (3.60).
    Let $\boldsymbol{\upsilon}^{(l+1)} = \boldsymbol{\upsilon}^{(l)} + \tau \boldsymbol{\Delta \upsilon}^{(l)}$.
    **if** $\|F(\upsilon)\|_Q < 0.9\epsilon$ **then**
        $\epsilon = 0.1\epsilon$
    **end if**
**end while**
```

3.4 Numerical examples

To illustrate the effectiveness of incorporating the L_2 function space approach in the GCMMA, we solve three common topology optimization problems. For each case, we use quadrangular elements with first order Lagrange basis function. The modified GCMMA is implemented in parallel using data structures from PETSc [12] and the FEA uses the libMesh library [51]. The design field discretization uses the typical volume fraction based topology optimization approach with element-wise piecewise uniform basis functions. In our examples, we use $a_0 = 1$, $c_i = 10000$ and $a_i = d_i = 0$ for all $i \geq 1$.

The topology optimization problem is formulated as

$$\begin{aligned} \min_{\nu \in [\epsilon_\nu, 1]} \theta_0(\nu) &= \int_D \pi(\hat{\nu}, \mathbf{u}) \, dV \, , \\ \text{such that } \mathbf{u} \in V \text{satisfies } a(\nu; \mathbf{u}, \mathbf{v}) &= L(\mathbf{v}) \text{ for all } \mathbf{v} \in V \, , \\ \theta_i(\nu) &= \int_D g_i(\hat{\nu}, \mathbf{u}) dV \leq 0 \qquad i = 1, 2...m \, , \end{aligned} \tag{3.67}$$

where

$$a(\nu; \mathbf{u}, \mathbf{v}) = \int_D r(\hat{\nu}) \mathbb{C}[\nabla \mathbf{u}] \cdot \nabla \mathbf{v} \, dV \, , \tag{3.68}$$

and

$$L(\mathbf{v}) = \int_\Gamma \mathbf{t} \cdot \mathbf{v} \, da \, . \tag{3.69}$$

The filtered volume fraction $\hat{\nu}$ in the above is obtained from the PDE-based filter [55] to generate a well posed topology optimization problem

$$-\kappa \nabla^2 \hat{\nu} + \hat{\nu} = \nu \, , \tag{3.70}$$

where κ determines the minimum length scale of the design such that a small (large) κ allows for fine (coarse) scale design fluctuations. We solve Equation (3.70) with a finite volume scheme. A lower bound ϵ_ν on ν is imposed in this ersatz approach to ensure the stiffness matrix in the finite element analysis is not singular. We use the RAMP penalization [87] to encourage a 0-1 design on the filtered volume fractions $\hat{\nu}$ such that

$$r(\hat{\nu}) = \frac{\hat{\nu}}{1 + q(1 - \hat{\nu})} \,, \tag{3.71}$$

where $q = 20$.

Finally, $\mathbb{C}$ is the elasticity tensor of an isotropic material with Young modulus $E = 1$ and Poisson ratio $\mathsf{v} = 0.3$ and $\mathbf{t}$ is the applied traction on the surface Γ. We run all the optimization problems until the number of iterations reaches 800 or the stopping criteria of Equations (3.66) is below 10^{-3}. These two criteria were deemed sufficient to obtain optimized designs.

As per usual, a reduced space approach is taken wherein we account for dependence of $\mathbf{u}$ on ν, i.e. $\mathbf{u} \mapsto \mathbf{u}(\nu)$ and the adjoint method is used to calculate the derivative.

The first problem we study is the proverbial structural compliance minimization subject to a volume constraint $\hat{V}$, i.e.

$$\theta_0 = \int_\Gamma \mathbf{t} \cdot \mathbf{u} \, da \,, \tag{3.72}$$

$$\theta_1 = \int_D \hat{\nu} \, dV - \hat{V} \,. \tag{3.73}$$

The design domain D, cf. Figure 3.2, is subject to the traction $\mathbf{t}(\mathbf{x}) = \exp(-4.0(y - 25)/(5))\mathbf{e}_2$ and the length scale parameter is $\kappa = 0.75$. We perform four different optimizations corresponding to uniform and nonuniform meshes with optimizations in ℓ_2 and L_2. The uniform mesh contains 196,608 elements. Our non-uniform mesh is illustrated in Figure 3.3 with four levels of refinement, but we perform six additional levels of uniform refinement for a total of 380,928 elements.

The optimized designs in Table 3.2 show that the ℓ_2 algorithm is mesh dependent as opposed to the L_2 algorithm. Figures 3.4 - 3.6 show the evolution of the cost and constraint functions and the stopping criteria, cf. Equations (3.66). The original ℓ_2 algorithm not only depends on the mesh, but it also performs worse than the L_2 algorithm. A common misconception in the topology optimization community is that the restriction by filtration ensures mesh independent designs. However, as seen here, this does not hold true for meshes with varying element sizes unless the optimization algorithm is formulated in the L_2 space. Also note that the L_2 algorithm converges sooner than the ℓ_2 algorithm.

We next benchmark our algorithm with the compliant mechanism design problem, cf. Figure 3.7 where

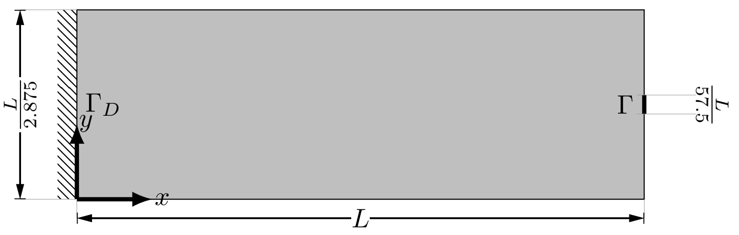


Figure 3.2: Compliance domain. $L = 150$ mm.

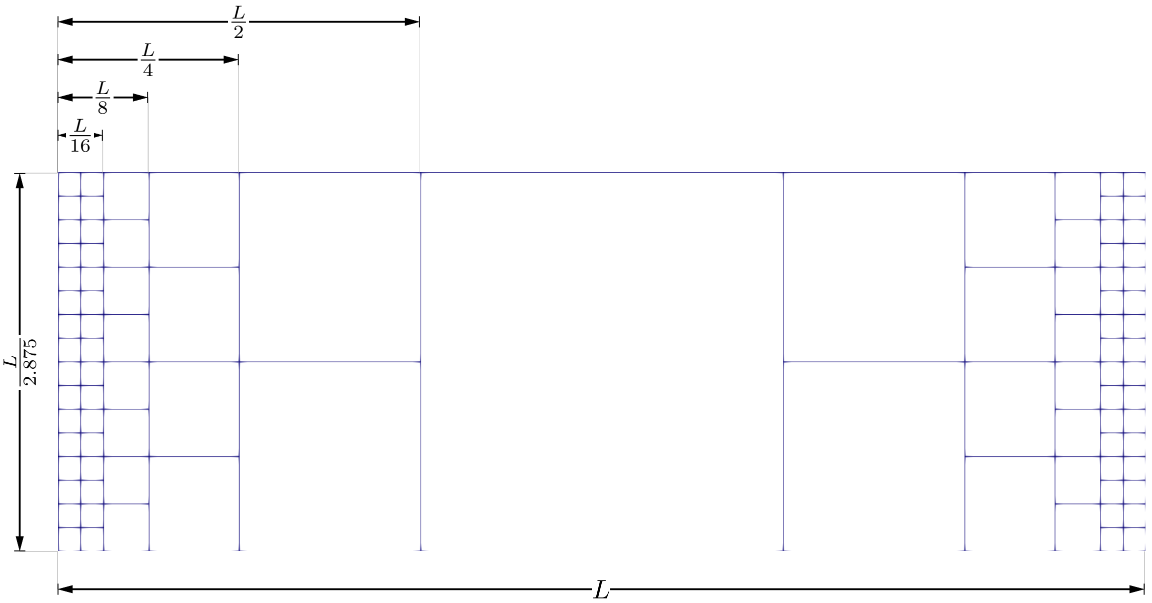


Figure 3.3: Non-uniform mesh showing four levels of refinement; six additional levels of uniform refinement are used for the computation.

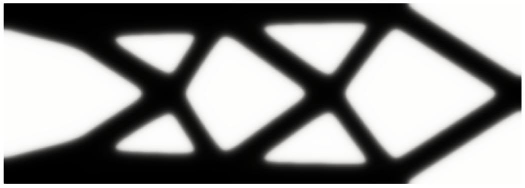
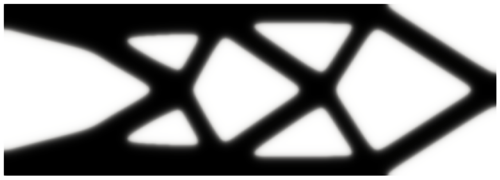
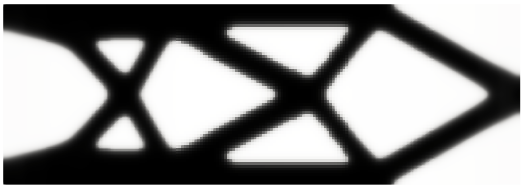
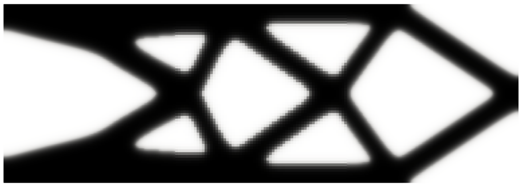
	Optimization in ℓ_2	Optimization in L_2
Uniform mesh		
Non-uniform mesh		

Table 3.2: Optimized designs for the compliance problem.

we minimize the horizontal displacement $u = \mathbf{u} \cdot \mathbf{e}_1$ in the output port Γ_2, subject to a maximum volume

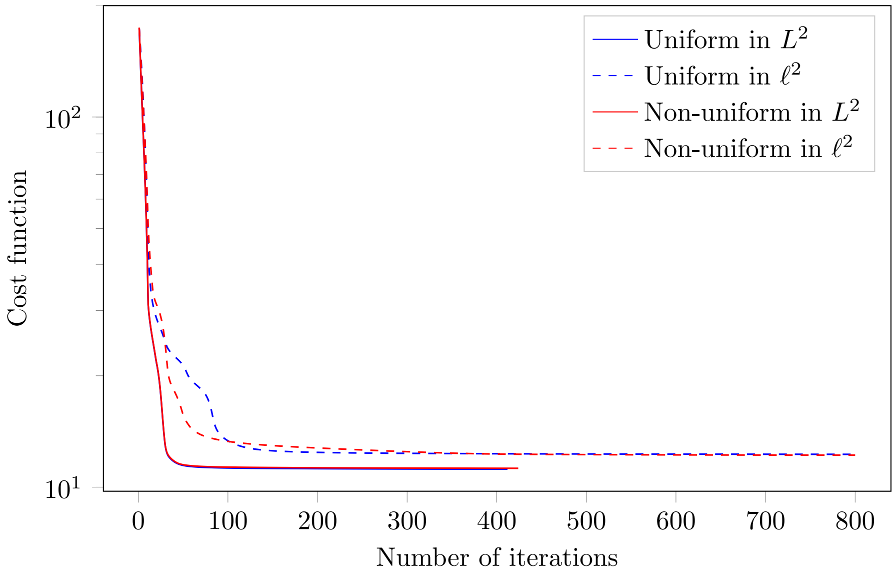


Figure 3.4: Cost function evolution for the compliance problem.

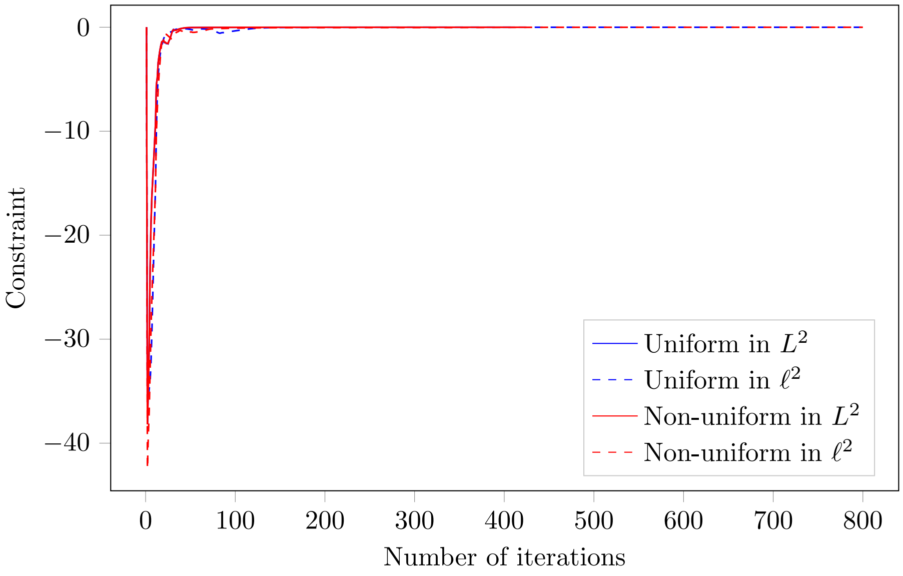


Figure 3.5: Constraint function evolution for the compliance problem.

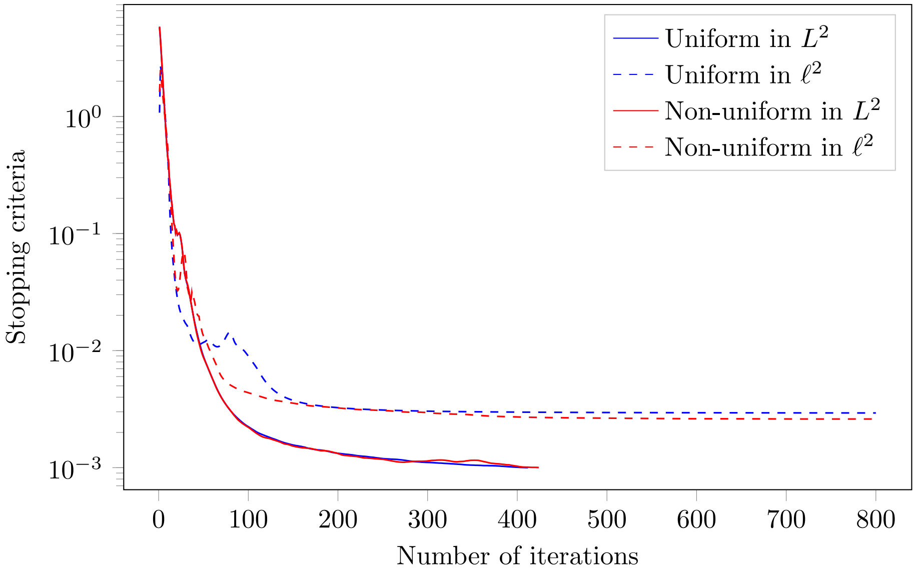


Figure 3.6: Stopping criteria evolution for the compliance problem.

constraint $\hat{V}$, viz.

$$\theta_0 = \int_{\Gamma_2} u \; da \,, \tag{3.74}$$

$$\theta_1 = \int_D \hat{\nu} \; dV - \hat{V} \,. \tag{3.75}$$

Consistent with [30], we introduce Robin boundary conditions into the (3.68) formulation.

$$\begin{aligned} \mathbf{n} \cdot \mathbb{C}[\nabla \mathbf{u}]\mathbf{n} &= -k_{in}(\mathbf{u} \cdot \mathbf{n}) + f_x \qquad \text{on } \Gamma_1 \,, \\ \mathbf{n} \cdot \mathbb{C}[\nabla \mathbf{u}]\mathbf{n} &= -k_{out}(\mathbf{u} \cdot \mathbf{n}) \qquad\qquad \text{on } \Gamma_2 \,, \end{aligned} \tag{3.76}$$

where $f_x = 10$, the spring coefficients are $k_{in} = 1.0$ and $k_{out} = 0.001$. Figures 3.7 and 3.8 illustrate the design domain and the different levels of refinement in the non-uniform mesh. No additional refinement is done in this example. The number of elements in the uniform and non-uniform meshes are 115,200 and 382,068.

The optimized designs for the length scale $\kappa = 1.1$ illustrated in Table 3.3 and Figures 3.9 - 3.11 again show that the original ℓ_2 algorithm is mesh dependent and performs worse than the L_2 version. The stopping criteria is small at the beginning of the optimization, however the designs are still evolving. As such, we allow the algorithm pass this region before terminating it.

Our last example is the stress constrained problem where the goal is to minimize the volume of an L-bracket subject to a pointwise constraint in the stress field, cf. Figure 3.12. We follow the formulation in

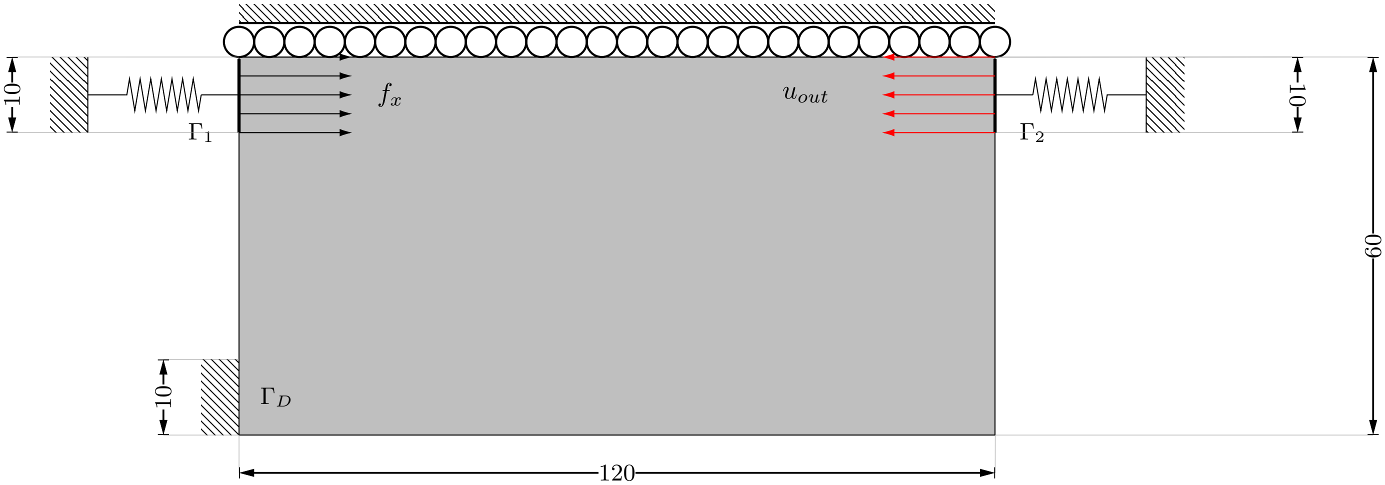


Figure 3.7: Design domain and boundary conditions for the compliant mechanism problem. Domain symmetry is used whereby only the lower half of the structure is analyzed.

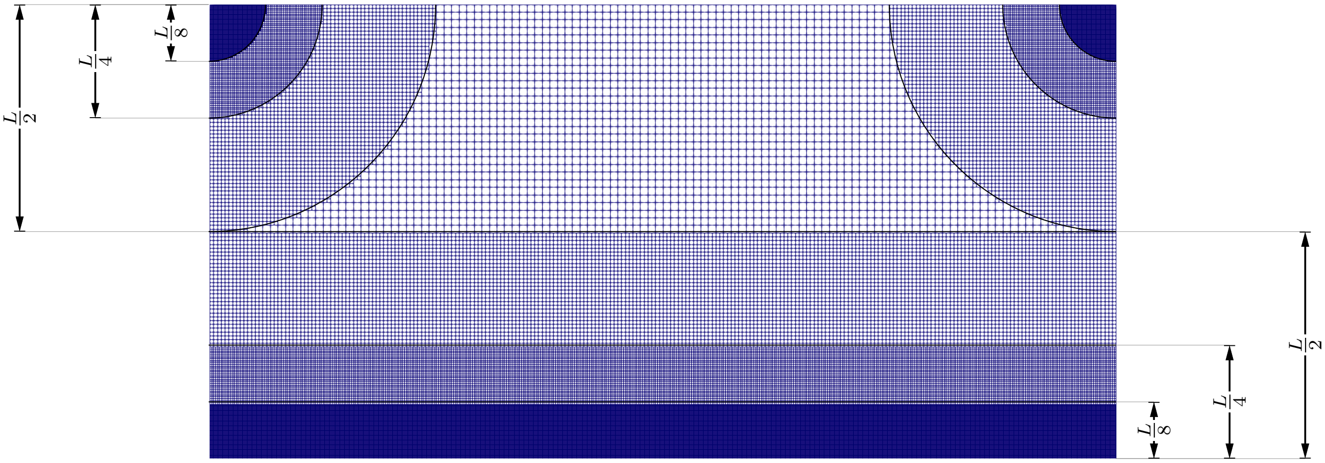


Figure 3.8: Non-uniform mesh of the compliant mechanism problem ($L = 60$).

[10], where the stress constraint is imposed via a penalty method, i.e. our cost function is

$$\theta_0 = \int_D \hat{\nu}\ da + \gamma \left\|\sigma_{VM} - \sigma_y\right\|_+ , \tag{3.77}$$

where the penalty parameter is $\gamma = 100$,

$$\left\|\sigma_{VM} - \sigma_y\right\|_+ = \int_D R_p\left(\frac{\sigma_{VM}}{\sigma_y}\right) dV , \tag{3.78}$$

$$R_p(x) = \left(1 + (x)^p\right)^{\frac{1}{p}} - 1 , \tag{3.79}$$

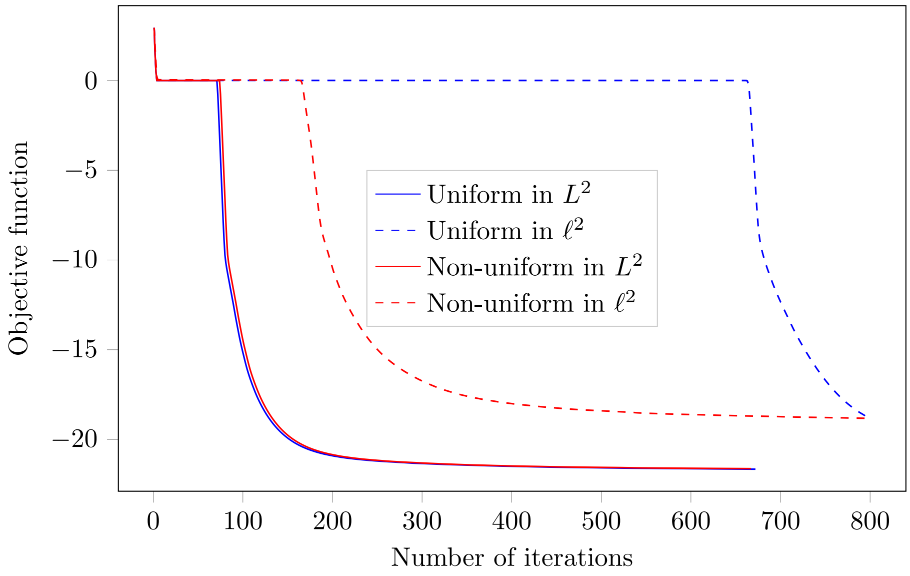


Figure 3.9: Cost function evolution for the compliant mechanism problem.

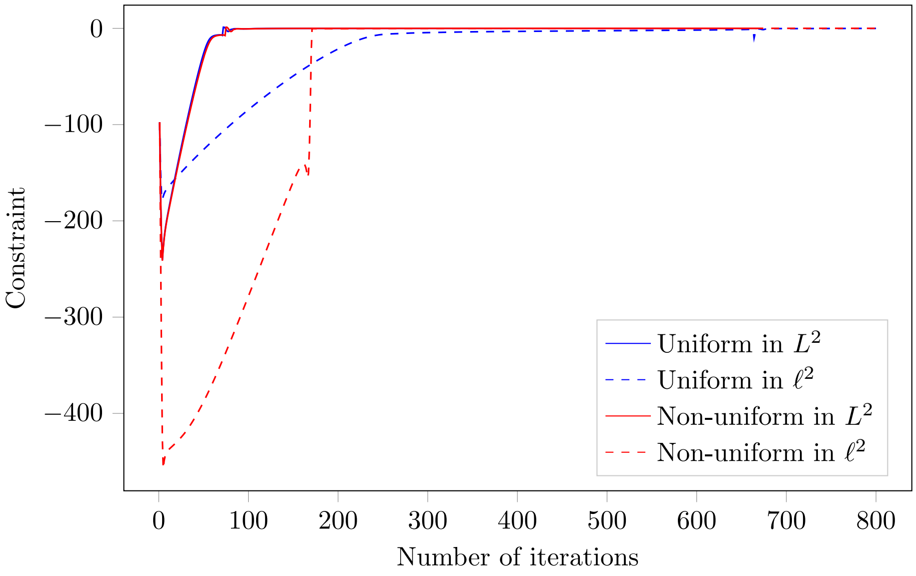


Figure 3.10: Constraint function evolution for the compliant mechanism problem.

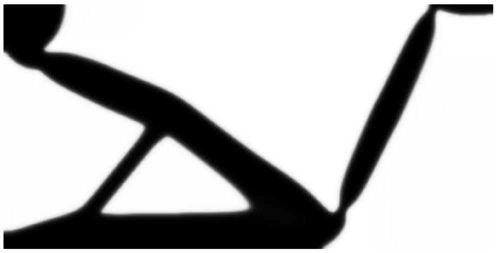
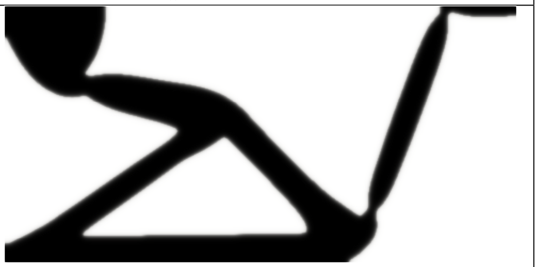
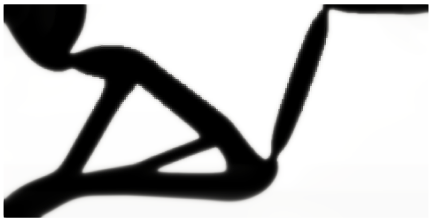
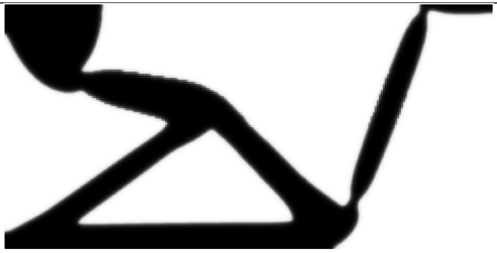
	Optimization in ℓ_2	Optimization in L_2
Uniform mesh		
Non-uniform mesh		

Table 3.3: Optimized designs for the compliant mechanism problem.

σ_{VM} is the Von Mises stress, $\sigma_y = 2.0$ MPa and $p = 20$. A relaxed stress formulation similar to [57] uses

$$\hat{\mathbb{C}} = \eta_c \mathbb{C}\,, \tag{3.80}$$

to calculate the Von Mises stress σ_{VM}, where η_c is the inverse RAMP function

$$\eta_c(\hat{\nu}) = \hat{\nu}\frac{1+q}{1+q\hat{\nu}}\,, \tag{3.81}$$

with $q = 20$. The filter parameter is $\kappa = 1.0$. To obtain a better design we extend the domain by adding a layer of finite elements near the reentrant corner, cf. the red dotted region in Figure 3.13. These new elements however, are not included in the design as they are assigned the uniform volume fraction $\varepsilon_\nu = 10^{-3}$. This added region lessens the boundary effect of the filter operation in the reentrant corner region which otherwise adversely affects our results. The non-uniform mesh shown in Figure 3.13 (which does not show two additional levels of uniform refinement) refines the regions surrounding the reentrant corner and applied load; it contains 46,501 elements, whereas the uniform mesh contains 74,752 elements.

The optimized designs in Table 3.4 again illustrate the mesh dependence of the original ℓ_2 algorithm. This time however, the difference in the cost function values between the ℓ_2 and L_2 designs is barely noticeable, cf. the log plot in Figure 3.14. Due to the high oscillation in the stopping criteria, we plot them separately in Figures 3.15 - 3.18. The stopping criteria is still high after 800 iterations despite having designs that appear to be optimal.

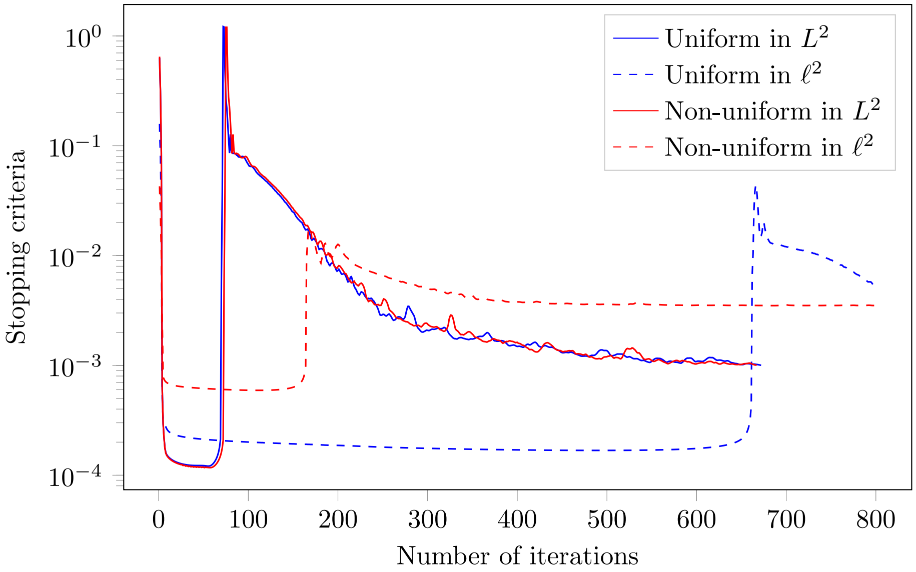


Figure 3.11: Stopping criteria evolution for the compliant mechanism problem.

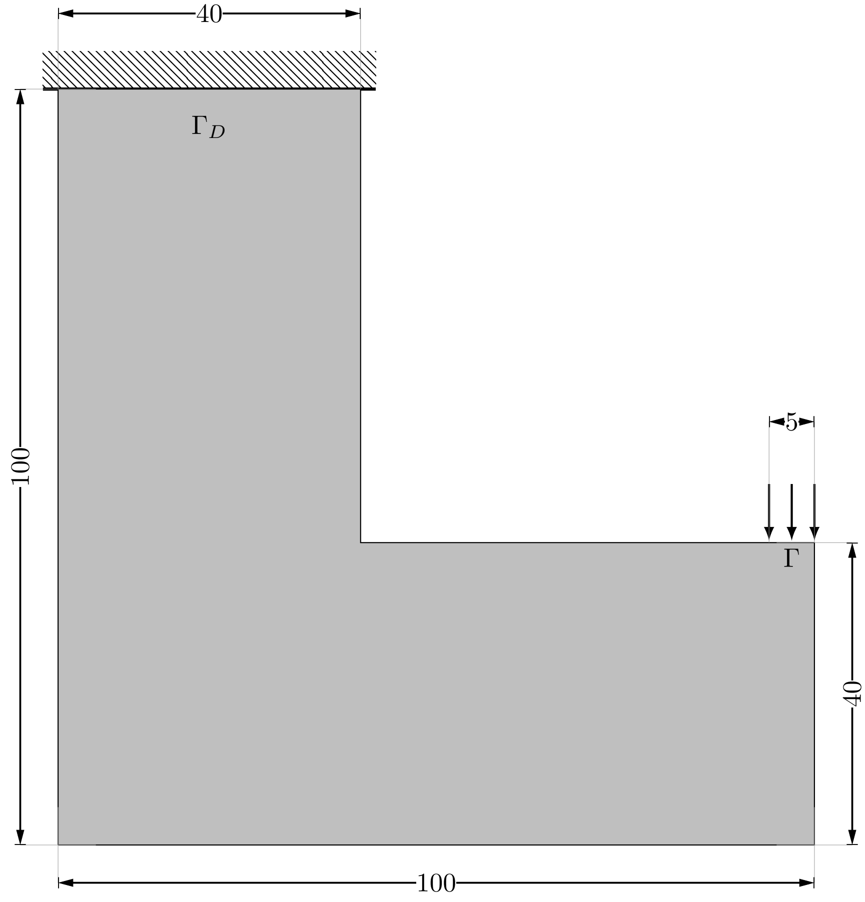


Figure 3.12: Intended design domain for the stress constrained problem. All dimensions are in mm.

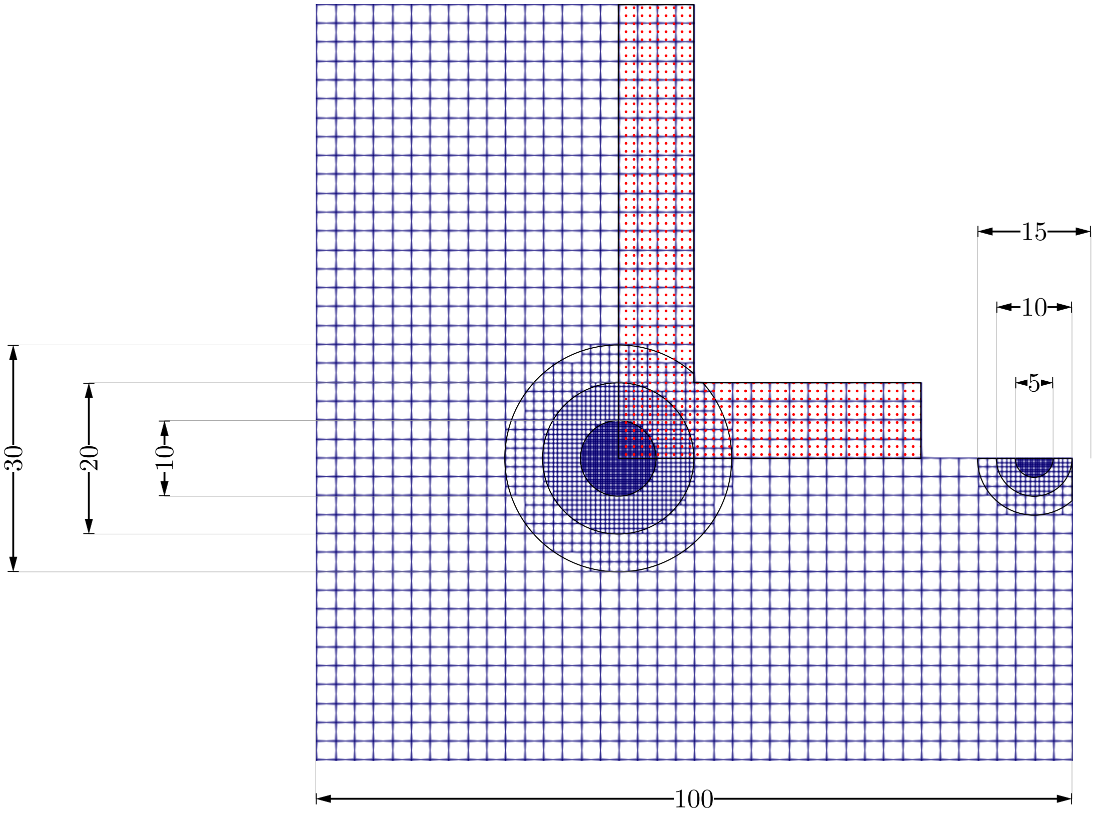


Figure 3.13: Non-uniform mesh for the stress constrained problem; two additional levels of uniform refinement are used for the computation. Red dots mark the extended region.

3.5 Conclusion

In this work, we presented the necessary mathematical concepts to obtain mesh independent topology optimization algorithms and applied them to the GCMMA algorithm. Our mesh-independent L_2 GCMMA implementation outperforms the original ℓ_2 GCMMA implementation on three common problems in topology optimization. Notably our results are independent of the mesh, while those with the algorithm in ℓ_2 are not. More importantly, mesh independence makes this L_2 algorithm suitable for use with AMR strategies within the optimization. However, the stopping criteria employed does not work well for the stress-constrained problem, which motivates future research.

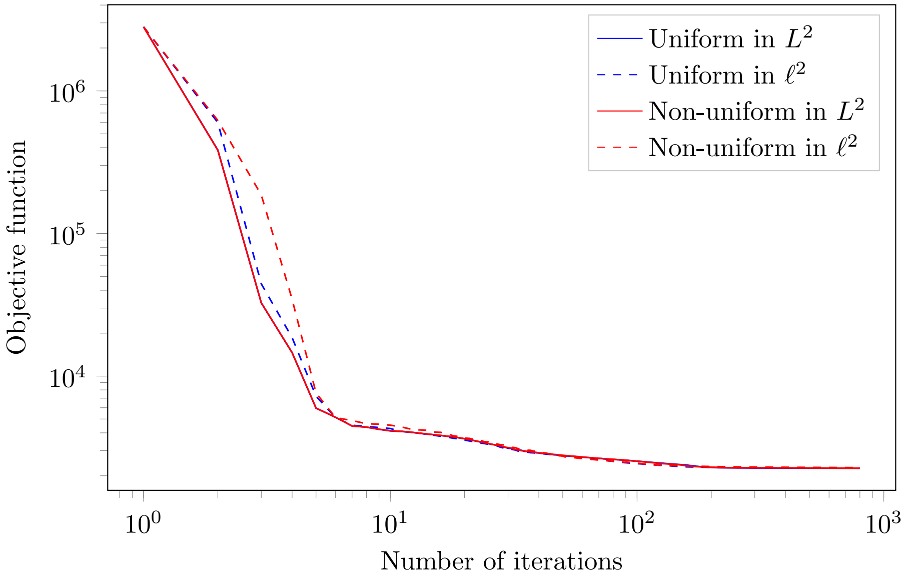


Figure 3.14: Cost function evolution for the stress constrained problem.

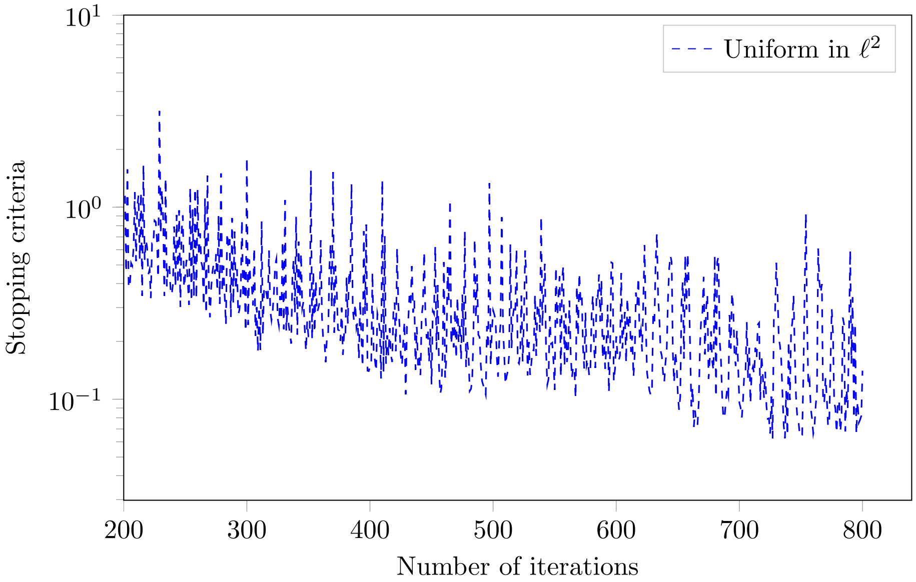


Figure 3.15: Stopping criteria evolution for the stress constrained problem with uniform mesh in ℓ_2.

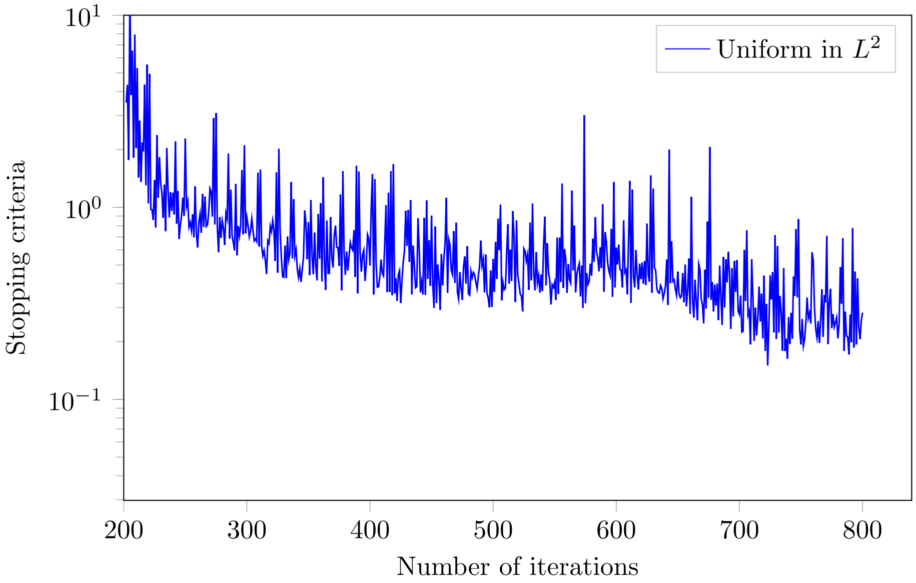


Figure 3.16: Stopping criteria evolution for the stress constrained problem with uniform mesh in L_2.

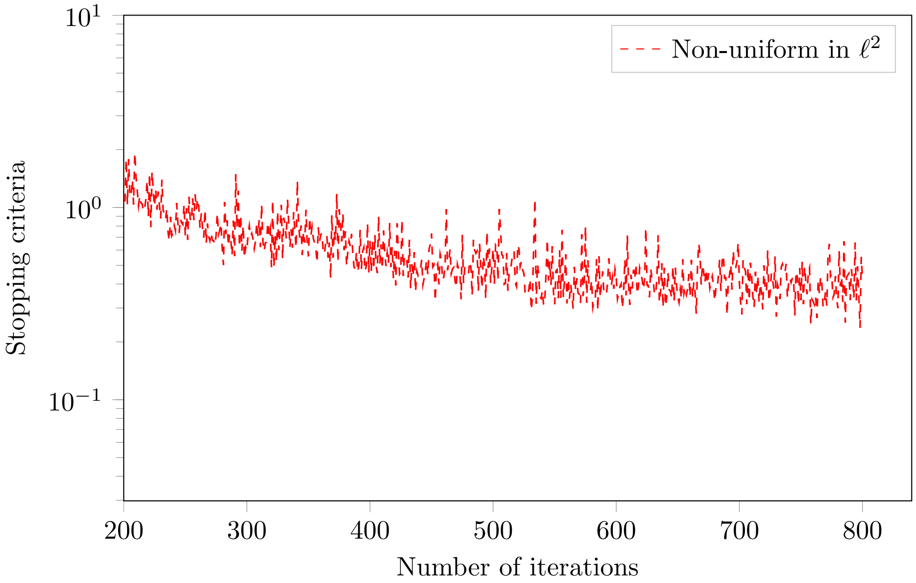


Figure 3.17: Stopping criteria evolution for the stress constrained problem with non uniform mesh in ℓ_2.

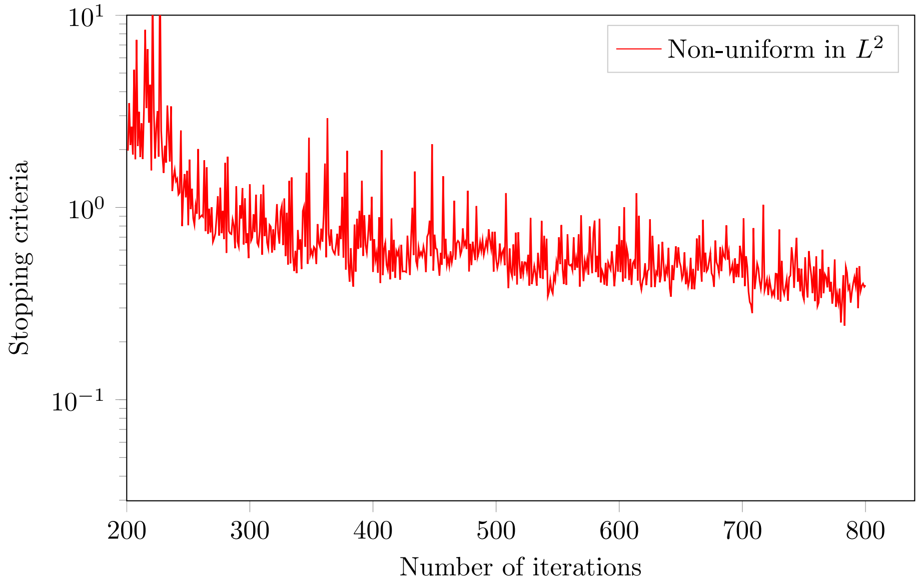


Figure 3.18: Stopping criteria evolution for the stress constrained problem with non uniform mesh in L_2.

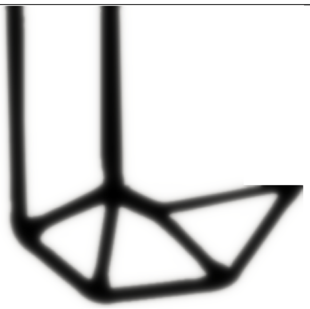
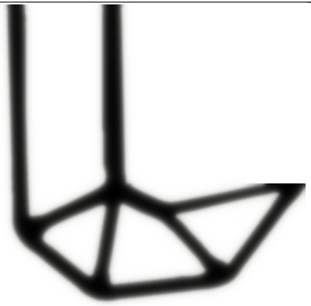
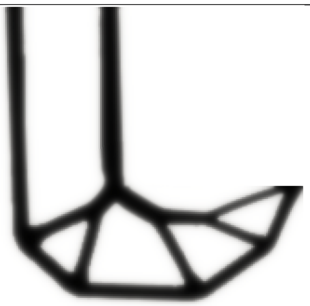
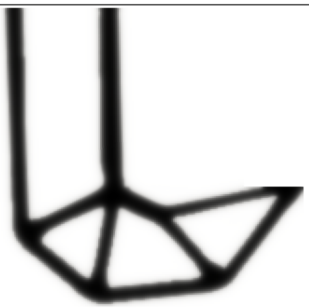
	Optimization in ℓ_2	Optimization in L_2
Uniform mesh		
Non-uniform mesh		

Table 3.4: Optimized designs for the stress constrained problem.

Chapter 4

Three dimensional adaptive mesh refinement in stress constrained topology optimization

4.1 Introduction

Topology optimization has found application in a wide range of industries: aerospace, automotive, medical, etc. However, to further extend its range of application, it is necessary to solve computationally expensive high resolution three dimensional design problems that involve multiple physical phenomena and cost and constraint functions. The challenges to reach this goal are several: First, direct linear solvers are no longer feasible when solving problems in excess of one million degrees of freedom. Only iterative linear solvers with physics dependent preconditioners can perform in reasonable time scales. Second, the mesh discretization cannot be excessively fine since the computational resources are limited. And thus, Adaptive Mesh Refinement (AMR) is required to efficiently resolve the design. Third, the same mesh used to discretize the displacement field is also used to discretize the design and the optimization Lagrange multiplier fields. Therefore, whenever AMR modifies the mesh, the design and Lagrange multiplier discretizations also change. This coupling between the optimization and the finite element analysis cannot be resolved with black box optimization algorithms. And fourth, the optimization algorithm must be able to accommodate arbitrary cost and constraint functions e.g. stress constraints, which proves difficult at large scales.

Large-scale design problems in topology optimization have been the focus of numerous previous works. To solve compliance problems with more than 1 M design variables, [101] takes advantage of the small design change between optimization iterations and proposes the recycling of the Krylov spaces and [35] highlights the convergence problems when implementing a domain decomposition linear solver, i.e. finite element tearing and interconnection/dual primal solver. Building upon the PETSc library [12] and MMA optimization algorithm [89], [4] implements a topology optimization framework to solve the compliance problem with more than 27 M variables with good scalability. [2] uses the PETSc framework with an optimality criteria nonlinear programming algorithm to design an airplane wing with more than 1 B design variables and [7] designs a heat sink cooled with natural convection with up to 27 M design variables.

Until now, most large scale topology optimization designs have been limited to compliance problems.

Large scale stress constrained designs are scarce, despite the fact that most designs need to satisfy them. This can be explained by the inherent difficulty to solve the stress constrained problem even over two dimensional domains. These difficulties arise from three specific sources: stress singularity, locality of the stress constraint and stress accuracy. We refer to [78] for more details on these issues. Using a full space formulation, [65] proposes a sequential approximate optimization algorithm to obtain stress-constrained designs with more than 200 K variables with neither aggregation nor stress relaxation techniques. Applying AMR to obtain high resolution designs, [58] solves a large scale stress and frequency constrained problem with more than 5 M design variables. However, they only apply AMR once during the optimization and their error indicator is not specified.

In this work, we solve three dimensional stress constrained topology optimization problems. We implement our topology optimization framework on top of the parallel finite element library libMesh [51], which allows us to address large scale problems as it employs the PETSc [12] linear algebra library. We solve the optimization problem using a mesh independent large scale version of MMA [89] implemented in PETSc. cf. Chapter 3. Besides the challenges mentioned above, we face an additional challenge in our implementation when solving the PDE-filter [55] to impose a minimum length scale on the design. To preserve volume fraction values between 0 and 1 over coarse mesh regions, we solve the PDE-filter using a finite volume instead of a Lagrange finite element discretization. We explain the mathematical phenomena which motivates our choice and derive the necessary sensitivities, as they differ from the original implementation in [55].

The layout of this chapter is as follows: In Section 4.2, we present the problem formulation of a stress constrained problem. Section 4.3 explains the motivation to use a finite volume solver for the PDE-filter. Additional considerations for the optimization algorithm and the preconditioner for the linear iterative solver are discussed in Section 4.4. Finally, Section 4.5 presents optimized stress constrained L-bracket and three-dimensional compliant mechanism designs.

4.2 Adaptive mesh refinement in stress constrained topology optimization

At the risk of being redundant, we restate the problem formulation for the stress constrained topology optimization problem with AMR

$$\begin{aligned} \min_{\nu\in(\epsilon_\nu,1]} \theta_0(\nu) &= \int_D \pi(\nu,\mathbf{u})\ dV + \gamma \left\|\sigma_{VM} - \sigma_y\right\|_+ \ , \\ \text{s.t. } \mathbf{u} \in V \text{satisfies } a(\nu;\mathbf{u},\mathbf{v}) &= L(\mathbf{v}) \text{ for all } \mathbf{v}\in V\ , \\ \theta_i(\nu) = \int_D g_i(\nu,\mathbf{u}) dV &\leq 0 \ i = 1,2...\mathrm{n}_i\ , \end{aligned} \tag{4.1}$$

where

$$a(\nu;\mathbf{u},\mathbf{v}) = \int_D r(\tilde{\nu})\mathbb{C}[\nabla\mathbf{u}]\cdot\nabla\mathbf{v}\ dV\ ,$$

and

$$L(\mathbf{v}) = \int_{\Gamma_N} \mathbf{t}\cdot\mathbf{v}\ dS\ .$$

In the above, ν is the volume fraction that defines the design geometry and $\tilde{\nu}$ is the soon to be defined thresholded filtered volume fraction. From the displacement $\mathbf{u}$, we calculate the Von Mises stress σ_{VM} and constrain it to be less than the yield stress σ_y by means of a ramp-like functional $\|\cdot\|_+$ [10] weighted by a penalty parameter γ. $\mathbb{C}$ is the elasticity tensor of an isotropic material with Young modulus $E = 1$ and Poisson ratio $\mathrm{v} = 0.3$ and $\mathbf{t}$ is the applied traction on the surface Γ_N. In this ersatz approach, the lower bound $\epsilon_\nu = 10^{-3}$ ensures the stiffness matrix in the finite element analysis is not singular. The cost function π and constraint functions g_i are defined in the examples.

To obtain a 0–1 layout, we use the RAMP function r [86] to penalize the stiffness tensor $\mathbb{C} \to r(\tilde{\nu})\mathbb{C}$ and an inverse RAMP function η_c to penalize the Von Mises stress $\sigma_{VM} \to \eta_c(\tilde{\nu})\sigma_{VM}$ at intermediate volume fractions defined such that

$$r(\nu) = \frac{\nu}{1+q(1-\nu)} \tag{4.2}$$

and

$$\eta_c(\nu) = \nu\frac{1+q}{1+q\nu}\ , \tag{4.3}$$

with $q = 20$.

Topology optimization problems are notoriously ill-posed. The optimal design consist of a non converging sequence of structures with highly oscillatory material-void regions. One way to obtain a well–posed problem is via restriction wherein a length scale constraint is imposed on the design space. This is often achieved by filtering [17] the volume fraction field such that [21].

$$\hat{\nu}(\mathbf{x}, \nu) = \int_{\mathcal{B}_\epsilon(\mathbf{x})} K(\mathbf{x} - \mathbf{y})\nu(\mathbf{y})dV \,. \tag{4.4}$$

where the kernel K is typically a cone function defined over a ball $\mathcal{B}_\epsilon$ of radius ϵ centered at $\mathbf{x}$ and $\hat{\nu}$ is the filtered volume fraction. We are interested in obtaining very sharp domain boundaries so it is necessary to employ a large number of elements in the discretization. This causes two issues when constructing the cone filter. The linear operator that defines the cone filter in Equation (4.4) has finite support over $\mathcal{B}_\epsilon(\mathbf{x})$. To find the elements covering $\mathcal{B}_\epsilon(\mathbf{x})$, a depth-first search algorithm is used which is time consuming when the size of the elements are small compared to the filter radius ϵ. Additionally, parallel implementations of depth-first searches requires much communication.

The drawbacks of the cone filter motivate [55] to develop an alternative “filter” based on a diffusion with decay Partial Differential Equation (PDE)

$$-\kappa^2 \nabla^2 \hat{\nu}(\mathbf{x}) + \hat{\nu}(\mathbf{x}) = \nu(\mathbf{x}) \text{ for } \mathbf{x} \in D \,, \tag{4.5}$$

where κ determines the minimum length scale of the design such that a small (large) κ allows for fine (coarse) scale design fluctuations. Solving the PDE is analogous to applying the filter Equation (4.4) because the Green’s function $G(\mathbf{x} - \mathbf{y})$, i.e. the solution of Equation (4.5) corresponding to a concentrated load $\delta(\mathbf{x} - \mathbf{y})$ has a similar smoothing effect as the cone filter. Indeed

$$\hat{\nu}(\mathbf{x}, \nu) = \int_D G(\mathbf{x} - \mathbf{y})\nu(\mathbf{y})dV \approx \int_{\mathcal{B}_\epsilon(\mathbf{x})} K(\mathbf{x} - \mathbf{y})\nu(\mathbf{y})dV \,, \tag{4.6}$$

since $G(\mathbf{x} - \mathbf{y})$ tends to zero for $\mathbf{y} \notin \mathcal{B}_\epsilon(\mathbf{x})$. Although we need to solve a PDE to obtain the filtered volume fraction field and to calculate the sensitivities, this computation is readily parallelized and hence efficient.

Unfortunately, the use of a filter results in blurred design boundaries. To lessen this effect, we replace the filtered volume fraction $\hat{\nu}$ with its thresholded value [98].

$$\tilde{\nu}(\hat{\nu}) = \frac{\tanh(0.5\beta) + \tanh(\beta(\hat{\nu} - 0.5))}{\tanh(0.5\beta) + \tanh(\beta(1.0 - 0.5))} \,, \tag{4.7}$$

where β is a parameter defined such that

$$\lim_{\beta\to\infty} \tilde{\nu}(\hat{\nu}) = H(x - 0.5)\,, \tag{4.8}$$

and H is the unit step function. β is increased throughout the optimization with a continuation strategy.

Our AMR algorithm is the same as that in [78], nonetheless we repeat it here for completeness. In each element, we calculate the residual-based a posteriori error indicator for the global energy norm [94]. We then refine those elements whose error indicator is above the 70-th percentile and coarsen the elements below the 5-th percentile as well as their siblings in the mesh hierarchy. This refinement/coarsening operation is performed whenever the relative change in the cost function between iterations is smaller than tol_{AMR}.

4.3 PDE filter solver

The so called minimum principle which applies to the diffusion with decay Equation (4.5), states that if $\nu(\mathbf{x}) \geq 0$ for all $\mathbf{x} \in D$, then $\hat{\nu}$ attains its minimum $\hat{\nu}^*$ on ∂D and $\hat{\nu} \geq 0$. If the numerical method we use to solve Equation (4.5) does not preserve this principle, $\hat{\nu}$ can be negative in some regions of the domain and destroy the coercivity property of the elasticity operator in Equation (4.2), and thus the congujate gradient linear solver and its preconditioner will fail. Discretizing Equation (4.5) via the Galerkin method with Lagrange finite elements does not ensure satisfaction of the minimum principle. Indeed, for a sufficiently small h/κ where h quantifies element size, the reaction term in Equation (4.5) dominates the diffusion term. As such, $\hat{\nu}$ effectively becomes the L^2-orthogonal projection of ν onto H^1, which necessitates oscillations to minimize the L^2-distance. As stated in [18], sufficient resolution prevents these oscillations, but this contradicts our AMR strategy, which uses a coarse mesh in some regions. To combat such oscillations we solve Equation (4.5) with a second-order finite volume method over a mesh of rectangular/cuboid elements (with varying sizes) as this inherently fulfills the minimum principle. We note here that an arbitrary unstructured mesh with higher-order geometry will require a more elaborate finite volume method to fulfill the minimum principle [36].

An alternative method to satisfy the minimum principle uses stabilized methods that add mesh-dependent artificial diffusion. However, this approach increases the effective filter radius. We could have also employed nonlinear schemes [23] to satisfy the minimum principle, but this increases the computational cost. Postprocessing techniques such as clipping $\hat{\nu}$ are also undesirable as they adversely affect the gradient computation and hence the optimization convergence.

We discuss the sensitivity calculation for the PDE filter since it significantly differs from the original

PDE filter [55]. We solve Equation (4.5) using a second-order finite volume scheme by a weak formulation in which we require $\hat{\nu}$ to satisfy

$$\int_D \left(-\kappa^2 \nabla^2 \hat{\nu} + \hat{\nu}\right) \tau \; dV = \int_D \nu \tau \; dV \, , \text{ for all } \tau \in W \tag{4.9}$$

where W is the space of elementwise uniform functions over the mesh $\mathcal{T}$ which consists of elements K with faces $\mathcal{E}$. Next, we apply divergence theorem to transform Equation (4.9) to

$$\sum_{K \in \mathcal{T}} \left[\sum_{\mathcal{E} \in K} \int_{\mathcal{E}} -\kappa^2 [\![\widehat{\nabla_{\mathbf{n}} \hat{\nu}} \tau]\!] \; dS + \int_K \hat{\nu} \tau \; dV \right] = \sum_{K \in \mathcal{T}} \int_K \nu \tau \; dV \text{ for all } \tau \in W \, . \tag{4.10}$$

Note the appearance of the jump term $[\![\widehat{\nabla_{\mathbf{n}} \hat{\nu}} \tau]\!] = ((\widehat{\nabla_{\mathbf{n}} \hat{\nu}} \tau)^+ - (\widehat{\nabla_{\mathbf{n}} \hat{\nu}} \tau)^-)$ [1] across the intra-element faces $\mathcal{E}$ shared by two elements K^+ and K^- . Next, we define the uniform element face flux $\widehat{\nabla_{\mathbf{n}} \hat{\nu}}$ over both K^+ and K^- by the central difference, i.e.

$$\widehat{\nabla_{\mathbf{n}} \hat{\nu}}^{\,+} = \widehat{\nabla_{\mathbf{n}} \hat{\nu}}^{\,-} = \frac{[\![\hat{\nu}]\!]}{\|\mathbf{c}^+ - \mathbf{c}^-\|} \, , \tag{4.11}$$

where $\mathbf{c}^+$ ($\mathbf{c}^-$) are the centroidal coordinates of the elements K^+ (K^-). In this way

$$[\![\widehat{\nabla_{\mathbf{n}} \hat{\nu}} \tau]\!] = \frac{[\![\hat{\nu}]\!]}{\|\mathbf{c}^+ - \mathbf{c}^-\|} [\![\tau]\!] \, , \tag{4.12}$$

where we use the jump identity $[\![ab]\!] = [\![a]\!]\{b\} + \{a\}[\![b]\!]$ and $\{a\} = \frac{1}{2}(a^+ + a^-)$ denotes the average operator.

For the sensitivity analysis, we express Equation (4.10) in a more compact way as

$$b(\hat{\nu}, \tau) = l(\tau) \text{ for all } \tau \in W \, , \tag{4.13}$$

where we use Equation (4.12) and define the bilinear

$$b(\hat{\nu}, \tau) = \sum_{K \in \mathcal{T}} \left[\sum_{\mathcal{E} \in K} \int_{\mathcal{E}} -\kappa^2 \frac{[\![\hat{\nu}]\!]}{\|\mathbf{c}^+ - \mathbf{c}^-\|} [\![\tau]\!] \; dS + \int_K \hat{\nu} \tau \; dV \right] \, , \tag{4.14}$$

and linear

$$l(\tau) = \sum_{K \in \mathcal{T}} \int_K \nu \tau \; dV \, , \tag{4.15}$$

forms.

[1] The normal gradient $\nabla_{\mathbf{n}}$ is defined such that $\nabla_{\mathbf{n}} a = \nabla a \cdot \mathbf{n}$

Given $\delta_{\hat{\nu}}\theta = \delta\theta(\hat{\nu}; \delta\nu)$, i.e. the variation of θ with respect to $\hat{\nu}$, we proceed to calculate

$$\delta_{\nu}\theta = \delta\theta(\hat{\nu}(\nu); \delta\hat{\nu}(\nu; \delta\nu)) \text{ for all } \delta\nu \in W\,, \tag{4.16}$$

i.e. the variation of θ at ν acting on $\delta\nu$. In the above, $\hat{\nu}$ satisfies (4.13) and $\delta\hat{\nu}(\nu; \delta\nu)$ is the variation of $\hat{\nu}$ at ν acting on $\delta\nu$. Taking the variation in Equation (4.13) with respect to ν yields the equality that $\delta\nu$ must satisfy

$$\delta b(\hat{\nu}(\nu), \tau; \delta\hat{\nu}(\nu; \delta\nu)) - \delta l(\nu, \tau; \delta\nu) = 0 \text{ for all } \tau \in W\,, \tag{4.17}$$

where $\delta b(\hat{\nu}(\nu), \tau; \delta\hat{\nu}(\nu; \delta\nu)) = b(\delta\hat{\nu}(\nu; \delta\nu), \tau)$ and $\delta l(\nu, \tau; \delta\nu) = \int_D \tau\, \delta\nu\, dV$ are the variations of b and l at ν acting on $\delta\nu$. We add Equation (4.17) to (4.16) to obtain

$$\delta_{\nu}\theta = \delta\theta(\hat{\nu}(\nu); \delta\hat{\nu}(\nu; \delta\nu)) + b(\delta\hat{\nu}(\nu; \delta\nu), \tau) - \delta l(\nu, \tau; \delta\nu)\,, \tag{4.18}$$

which holds for all $\tau \in W$. As per usual, in this adjoint method we annihilate $\delta\hat{\nu}(\nu; \delta\nu)$, by requiring τ to solve the adjoint problem

$$b(\delta\hat{\nu}(\nu), \tau) = -\delta\theta(\hat{\nu}(\nu); \delta\hat{\nu}(\nu; \delta\nu)) \text{ for all } \delta\hat{\nu}(\nu; \delta\nu) \in W\,. \tag{4.19}$$

We then use this τ in Equation (4.18) which reduces to

$$\begin{aligned} \delta_{\nu}\theta &= -\delta l(\nu, \tau; \delta\nu) \\ &= -\int_D \tau\delta\nu \; dV \text{ for all } \delta\nu \in W\,. \end{aligned} \tag{4.20}$$

In the forward PDE analysis, we obtain the filtered volume fractions $\hat{\nu}$ by solving Equation (4.13) for the given ν which defines the linear form l of Equation (4.15). In the adjoint analysis, we obtain the adjoint field τ by solving Equation (4.19) for a given $\delta\theta(\hat{\nu}; \delta\hat{\nu}(\nu; \delta\nu))$ that acts as a linear operator with respect to the implicitly defined variation $\delta\hat{\nu}(\nu; \delta\nu)$. We finally use τ to evaluate the linear operator in Equation (4.20) which gives the desired sensitivity $\delta_{\nu}\theta$.

4.4 Optimization algorithm and iterative solver

We solve the optimization problem (4.1) with a large scale implementation of the MMA algorithm [89] in PETSc [12] similar to the implementation by [5], although we solve the convex approximation using an interior point method, as the original algorithm by [89], instead of a dual method as in [5].

Three dimensional stress constrained problems are difficult to solve with the MMA. The reason being the lack of a convergence mechanism that prevents oscillations in the cost function value and convergence to a local minimum. To reduce the oscillations without including a convergence strategy, the MMA tightens the bound constraints for the convex approximation at each iteration k such that in each element l

$$\begin{aligned} \nu_{\max}^{(k,l)} &= \min(\nu_{\max}, \nu^{(k,l)} + \sigma(\nu_{\max} - \nu_{\min}))\,, \\ \nu_{\min}^{(k,l)} &= \max(\nu_{\min}, \nu^{(k,l)} - \sigma(\nu_{\max} - \nu_{\min}))\,, \end{aligned} \tag{4.21}$$

where σ is the move limit parameter, a value $\sigma = 0.01$ hinders the convergence rate, but avoids oscillations. The GCMMA contains an additional inner loop in each optimization iteration. However, when changing the Heaviside threshold parameter β in Equation (4.7) with our continuation approach, the number of inner loop iterations skyrocketed and the convex approximation becomes too conservative whereby the design change between iterations was almost zero. For this reason, we opted for the MMA.

Whenever possible, we implement the MMA using the same data structures we use to discretize the design field ν. Notably, we use the ν data structure to discretize the Lagrange multipliers for the bound constraints, the convex approximation bound constraints and the moving asymptotes, to ensure they share the same mesh partitioning. Finally, we formulate the MMA algorithm in the L_2 space making it mesh independent, a crucial aspect in AMR, cf. Chapter 3.

To solve the linear system arising in the primal and adjoint problems, we use a conjugate gradient method with the geometric algebraic multigrid preconditioner (GAMG) in PETSc [12]. We pass the rigid body modes to the multigrid method so it can better approximate the near nullspace for the elasticity solver. The specific PETSc options are

```
-ksp_type cg
-ksp_atol 1e-7
-ksp_rtol 1e-9
-ksp_norm_type unpreconditioned
-pc_type gamg
-pc_gamg_verbose 7
-pc_gamg_type agg
-pc_gamg_agg_nsmooths 1
-pc_gamg_threshold 0.001
-mg_levels_esteig_ksp_type cg
-mg_levels_ksp_chebyshev_esteig_steps 50
-mg_levels_ksp_type chebyshev
-mg_levels_pc_type sor
```

For the PDE-filter solve of Equation (4.5), we employ the algebraic multigrid in hypre [60] as the preconditioner of the congujate gradient PETSc implementation. For this, we use the hypre options

```
-ksp_type cg
-pc_type hypre
-pc_hypre_type boomeramg
-pc_hypre_boomeramg_max_iter 200
-pc_hypre_boomeramg_coarsen_type HMIS
-pc_hypre_boomeramg_agg_nl 1
-pc_hypre_boomeramg_strong_threshold 0.25
-pc_hypre_boomeramg_interp_type ext+i
-pc_hypre_boomeramg_P_max 4
-pc_hypre_boomeramg_relax_type_all sequential-Gauss-Seidel
-pc_hypre_boomeramg_grid_sweeps_all 1
-pc_hypre_boomeramg_max_levels 25
```

4.5 Results

We first benchmark the algorithm with a three dimensional version of the common L-bracket design problem, cf. Figure 4.1. The cost function θ_0 in Equation 4.1 is the volume plus the penalty term for the stress constraint, i.e. $\theta_0 = \int_D \tilde{\nu} \; dV + \gamma \left\| \sigma_{VM} - \sigma \right\|_+$. The initial design domain D in Figure 4.1 is augmented by an extended red region D_{red} to mitigate the boundary effects due to the filter operation. To recover the original problem we impose the constraint

$$\int_{D_{\text{red}}} \tilde{\nu} \; dV \leq 0 \tag{4.22}$$

We apply a $-1\mathbf{e}_3$ N/mm^2 vertical load over the 10×10 mm^2 red square region in Figure 4.1. The maximum allowable Von Mises stress is $\sigma_y = 20$ MPa and the length scale parameter is $\kappa = 0.7$. We use an initial coarse mesh with 390 elements and apply two levels of uniform refinement for a total of 24,960 elements to begin the optimization. During the optimization, we allow for six additional levels of refinement from the initial 24,960 element coarse mesh. As in [78], we modify several parameters during the optimization to obtain sharp boundaries and avoid falling into local minima. First, the Heaviside threshold parameter β in Equation (4.7) starts with a value of 1 until iteration number 100. Thereafter it is increased by 5 every 20 iterations up to a maximum value of 20. Second, the penalty parameter γ starts with an initial value of 1 and increases by 2 every 20 iterations after iteration 10. Third, the maximum refinement level is fixed at 4 until iteration 200, and 6 thereafter. Finally, we use $tol_{AMR} = 0.1$ [78]. These strategies give us the best results. For example, they prevent areas with intermediate volume fraction values, which are removed by the threshold function later in the optimization, from being overly refined.

The optimized design in Figure 4.2 contains 2,815,813 elements in the thresholded design and 3,477,577 elements in the extended domain $D \cup D_{\text{red}}$. An equivalent uniform mesh with the same level of refinement would contain 102,236,160 elements. Figure 4.3 shows the design evolution at select iterations. Figure 4.4 contains a clipped portion of the optimized design's mesh to illustrate the different levels of refinement. Our design shows a fairly uniform Von Mises stress field, cf. Figure 4.5, which is a desirable aspect of optimal structures. In Figure 4.6 we plot the evolution in the cost function, and the number of degrees of freedom. All visualizations were done with Paraview [11].

The simulations were run with 50 2.60 GHz Intel Xeon E5-2670 processors. The optimization ran for 20 hours and 368 design iterations. The linear iterative solver for the primal analysis took an average of 30 iterations to converge for $\beta = 1$ and increased to an average of 50 for $\beta = 20$. The entire primal solver over the 3,477,577 element mesh with $\beta = 20$ took on average 163.77 seconds, out of which the preconditioner setup took 81.78 seconds on average. Over the same mesh, the PDE filter linear solve took an average of 2 iterations to converge for a total time of 161.62 seconds with 0.62 seconds devoted to the preconditioner setup. Our iterative solver is notably slower than the implementation in [5]. Indeed, their geometric multigrid preconditioner implementation on a uniform structured mesh significantly hastens the iterative solver, especially for the PDE filter. Unfortunately, implementing this preconditioner for adaptively refined nonuniform meshes is non-trivial; libMesh does not have this capability, although there is an action plan to implement it.

Next, we design a compliant mechanism whose purpose is to transfer an input displacement to an output displacement through elastic deformation, thereby reducing wear, friction and backlash, commonly seen

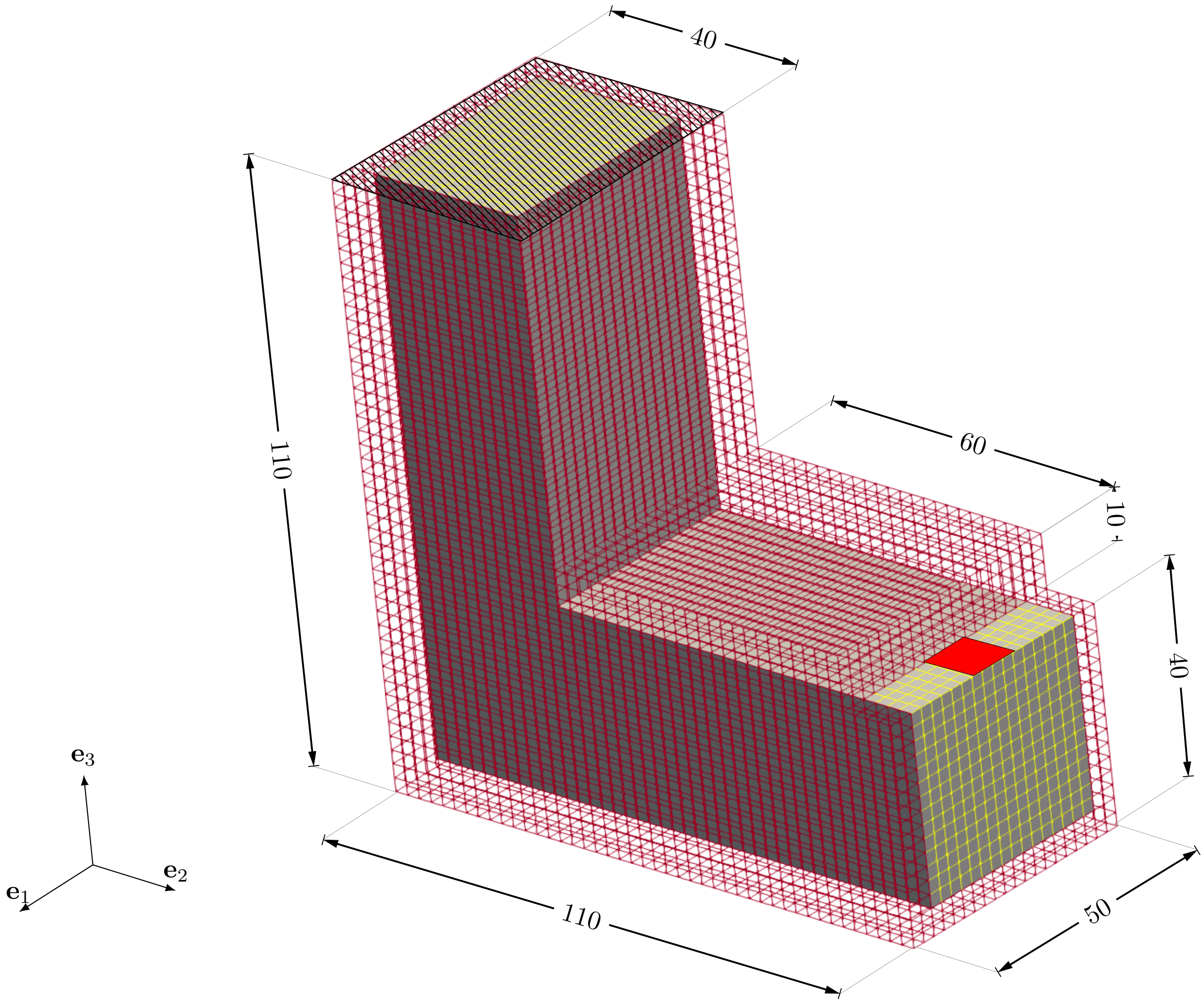
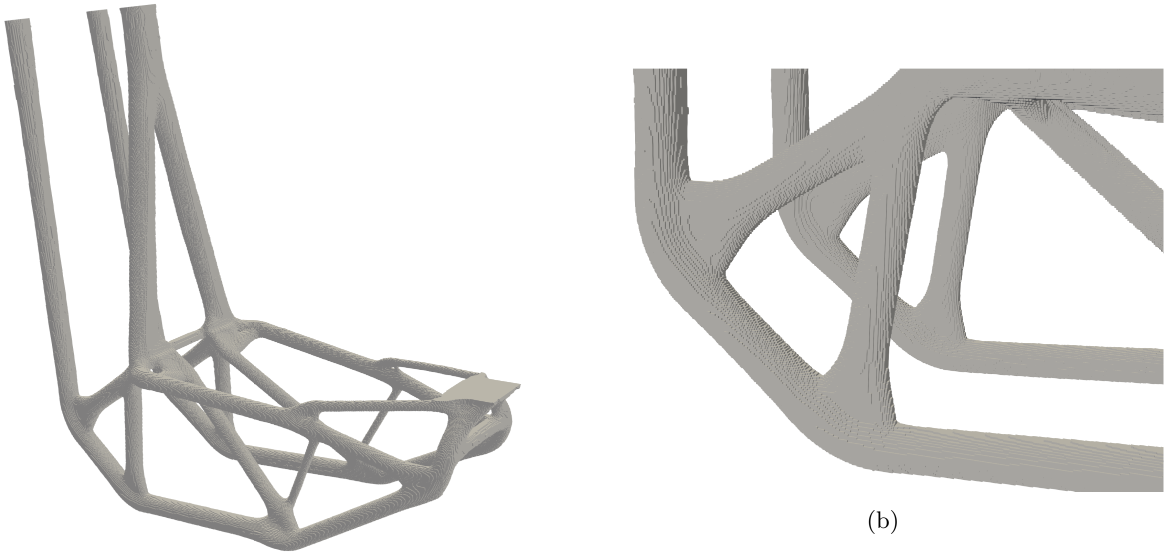


Figure 4.1: Mesh for initial L-bracket domain D with the extended simulation domain D_{red} in red. Dimensions in mm.



(b)

(a)

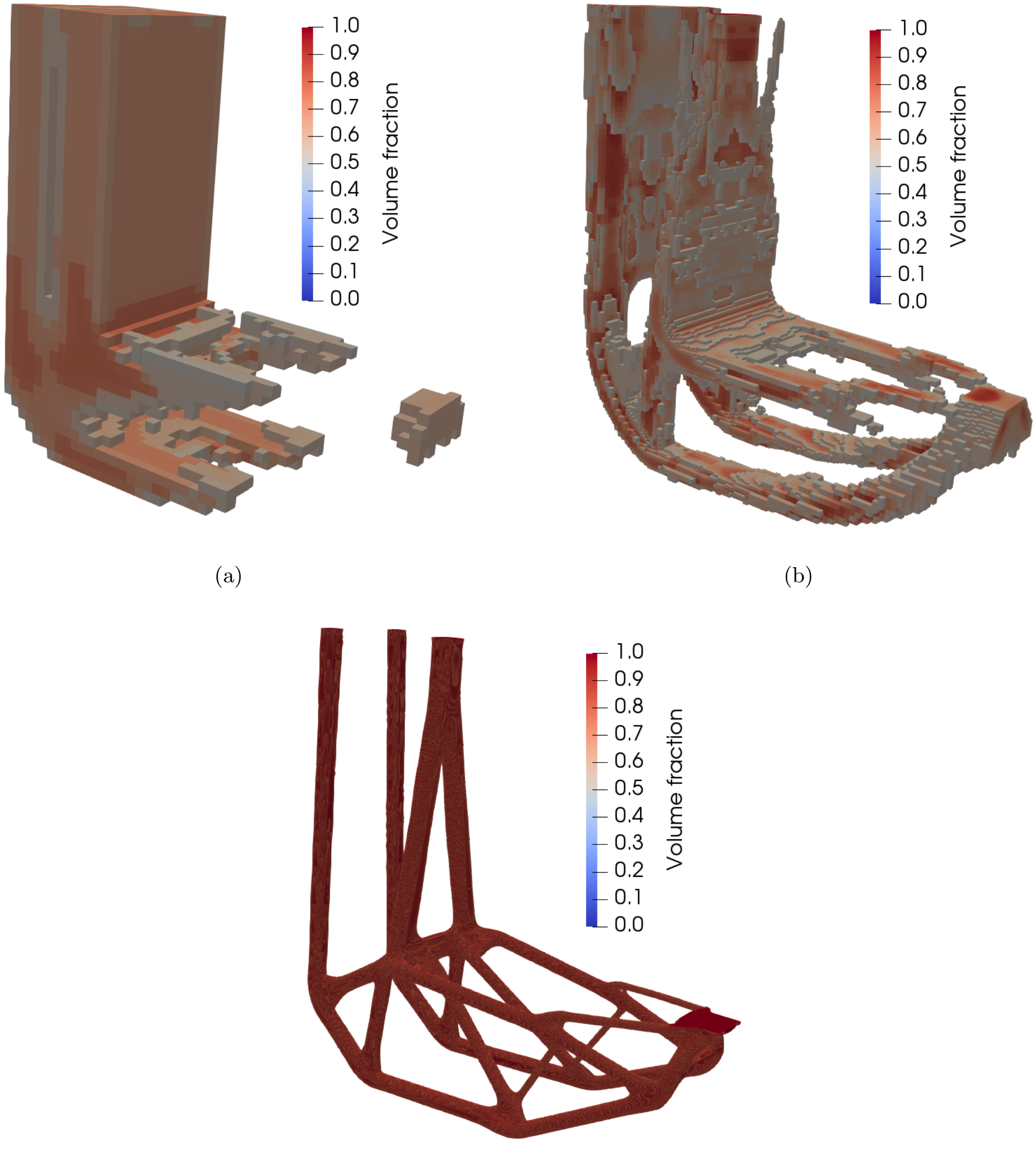
Figure 4.2: Optimized L-bracket design shows only thresholded volume fractions $\tilde{\nu}$ greater than 0.8 (a) and detailed region (b).

in rigid-body mechanisms. Most compliant mechanism designs contain "hinge" regions to maximize the output displacement. Such hinge regions experience high stress and are susceptible to fatigue failure. This motivated several previous authors [30, 64, 79] to include a maximum allowable stress constraint. We use our framework to extend this work to three dimensional structures. Besides the usual Dirichlet boundary conditions over Γ_D cf. Figure 4.7, we follow [30] and include Robin boundary conditions to surfaces Γ_1 and Γ_2, i.e.

$$\begin{aligned} \mathbf{n} \cdot \mathbb{C}[\nabla \mathbf{u}]\mathbf{n} &= -k_{in} u + t_x \qquad \text{on } \Gamma_1 \, , \\ \mathbf{n} \cdot \mathbb{C}[\nabla \mathbf{u}]\mathbf{n} &= -k_{out} u \qquad\qquad \text{on } \Gamma_2 \, , \end{aligned} \tag{4.23}$$

where $u = \mathbf{u} \cdot \mathbf{e}_1$, $k_{in} = 1 \ N/\text{mm}^3$, $k_{out} = 0.001 \ N/\text{mm}^3$ and $\mathbf{n}$ is the surface normal. We apply a horizontal load t_x of 10 N/mm^2 over the Γ_1 surface and equate the length scale parameter to $\kappa = 0.4$. Refering to Equation (4.1), we maximize the normal displacement over the surface Γ_2 subject to the penalized stress constraint, i.e. $\theta_0 = -\int_{\Gamma_2} \mathbf{u} \cdot \mathbf{n} \, dS + \gamma \, \|\sigma_{VM} - \sigma_y\|_+$ and subject to the volume constraint $\theta_1 = \int_D \tilde{\nu} \, dV \leq \hat{V}$, where $\hat{V}$ is 30% of the total domain volume. As in Figure 4.1, we extend the design domain D by adding a fine boundary layer D_{red} and enforce a zero volume constraint on the added region. To avoid stress concentrations caused by the boundary conditions, we prescribe unit volume fractions in the cuboid regions encompassing the surfaces Γ_1 and Γ_D and we do not include these regions in the stress constraint evaluation. We employ the same continuation and AMR strategies as in the L-bracket example.

Using domain symmetry, we simulate one quarter of the total domain and use the usual symmetry



(a)

(b)

(c)

Figure 4.3: L-bracket designs at iterations 15 (a), 45 (b) and 368 (c).

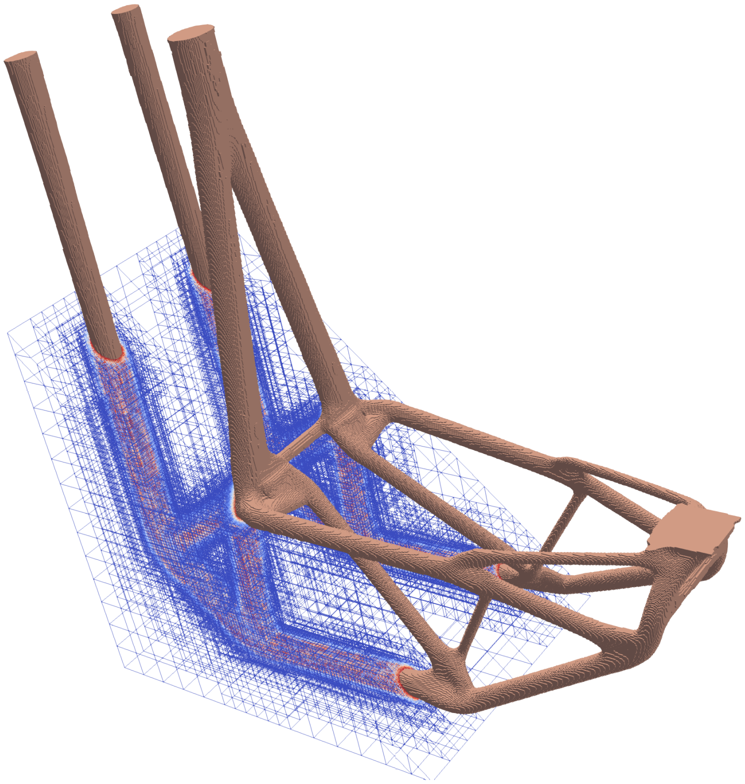


Figure 4.4: Clipped mesh of optimized L-bracket.

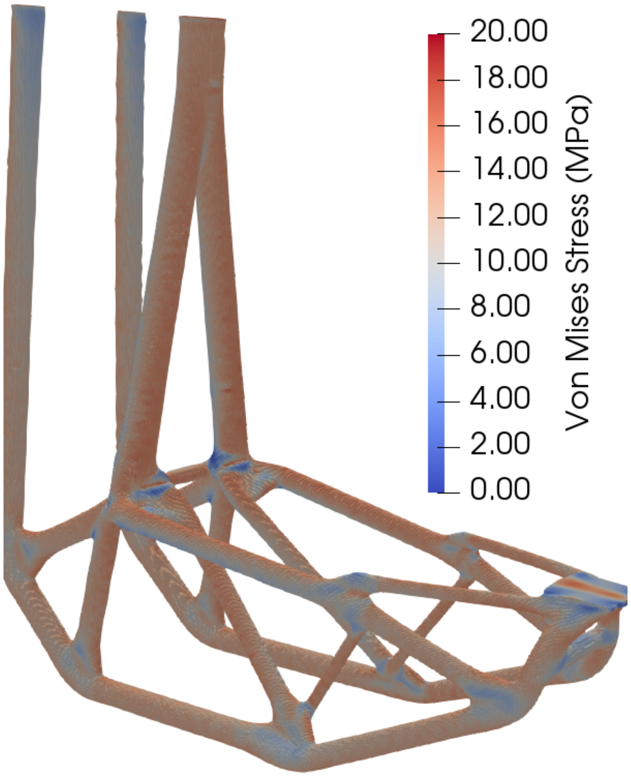


Figure 4.5: Von Mises stress field for the optimized L-bracket.

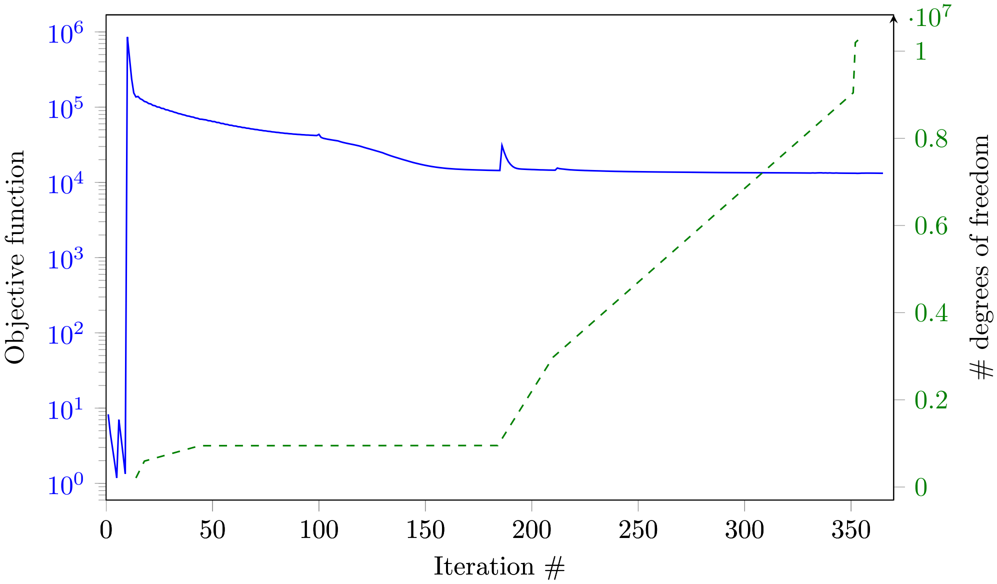


Figure 4.6: Cost function (solid) and number of degrees of freedom (dashed) evolutions for the L-bracket design problem.

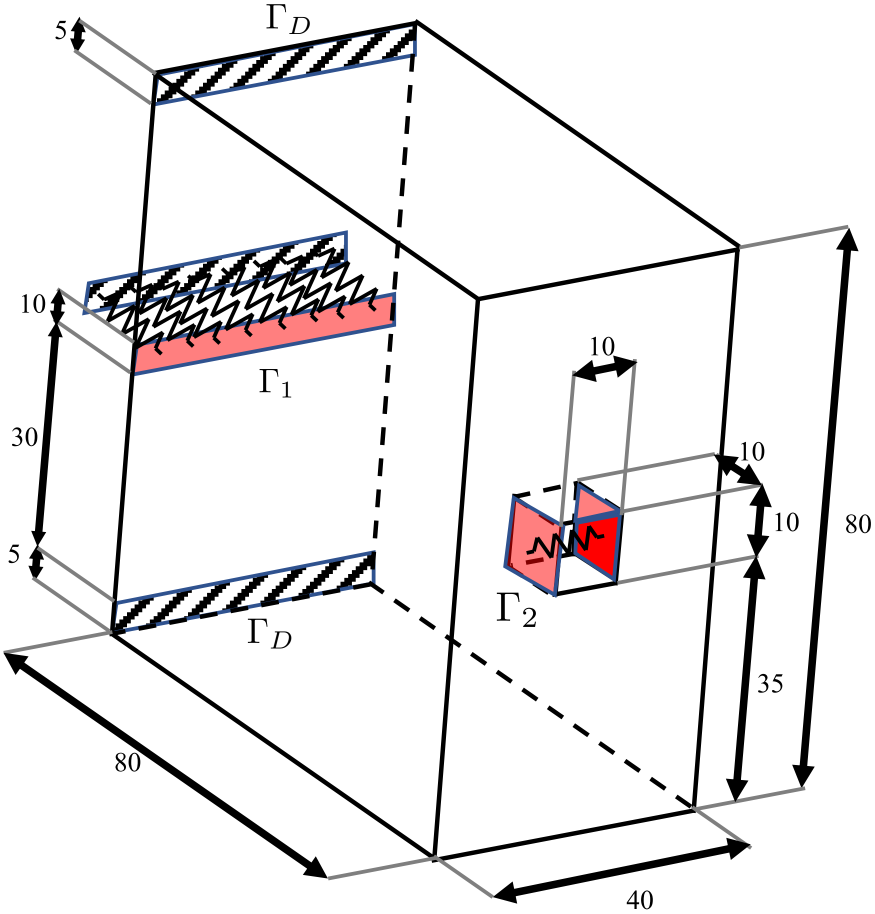


Figure 4.7: Compliant mechanism design domain. Dimensions in mm.

boundary conditions over the cut surfaces. We use an initial coarse mesh with 5,037 elements and apply one level of uniform refinement for a total of 40,296 elements for the initial design. During the optimization, we allow a maximum of four levels of refinement from the initial 40,296 element coarse mesh.

We first optimize a design with no stress constraint, i.e. $\gamma = 0$ in Equation (4.1). The optimized design, cf. Figure 4.8, for the quarter model contains 2,030,128 thresholded elements, and 2,498,605 total elements. The stress field illustrated in Figure 4.9 shows large stress concentrations in the eight line hinge regions. This motivates us to constrain the maximum allowable Von Mises stress to be less than $\sigma_y = 0.15$. The resulting optimized design replaces the line hinges with slender members cf. Figures 4.10-4.11. Not surprisingly, the cost function value has substantially increased compared to the unconstrained case, cf. Figure 4.12.

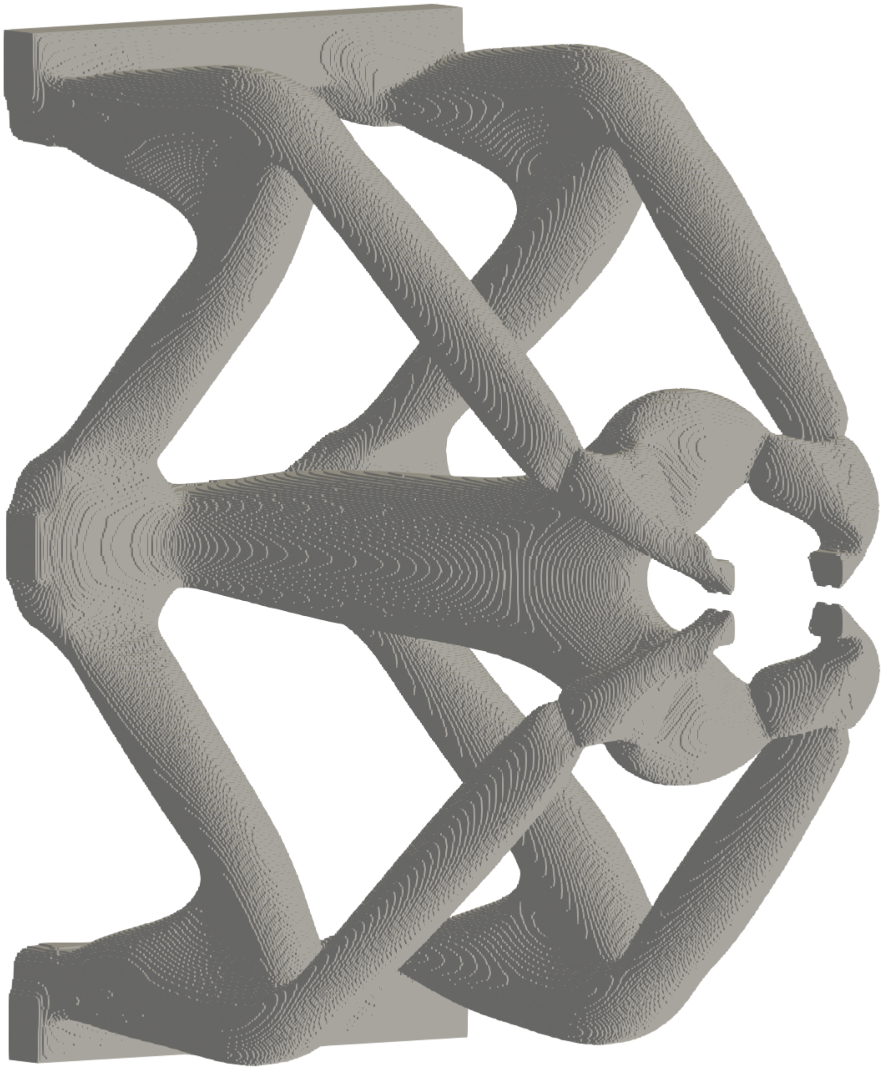


Figure 4.8: Optimized compliant mechanism design with no stress constraint shows only thresholded volume fractions $\tilde{\nu}$ greater than 0.8.

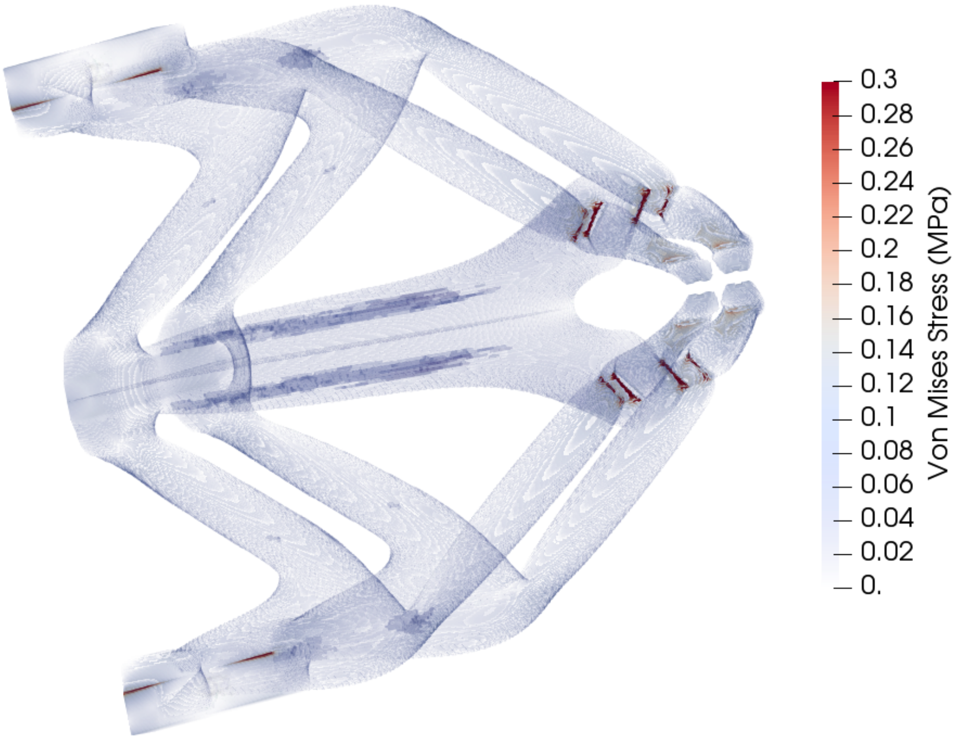


Figure 4.9: Von Mises stress field for the compliant mechanism design with no stress constraint.

4.6 Conclusions

We present a high resolution topology optimization framework that accommodates more than three million design variables in stress constrained problems by combining AMR with a parallel mesh independent optimization algorithm and parallel linear algebra library. To preserve the 0–1 bounds on the filtered volume fractions, we use a finite volume solver instead of a Galerkin method to resolve the PDE filter. When solving three dimensional stress constrained problems with the MMA we noticed an oscillatory convergence behavior which we attribute to the lack of a convergence mechanism. To facilitate convergence, we tighten the bounds of the convex approximation cf. Equation (4.21). Three dimensional L-bracket and compliant mechanism designs are optimized to benchmark the framework. Using our high resolution scheme allows us to obtain designs with smooth surfaces that respect the stress constraint while saving computational resources. Such resolution is not possible using a uniform mesh. Not surprisingly, the stress constraint enforcement severely affects the optimized compliant mechanism design by replacing line hinges with slender members.

Among the most immediate future work improvements is adopting the libMesh geometric multigrid preconditioner once it is implemented. Despite the computational savings by using AMR, there is still a high computational cost. We believe that uncoupling the geometry from the simulation mesh by combining

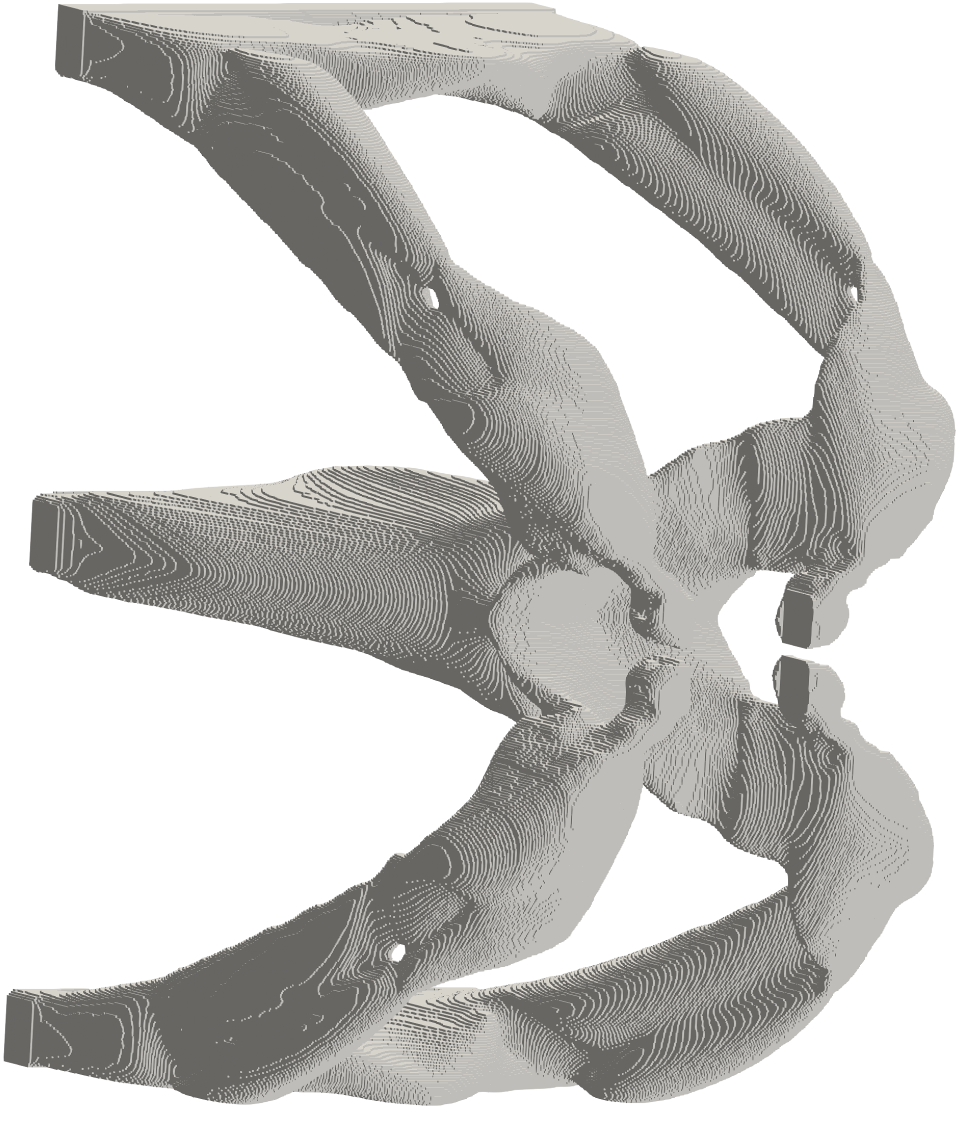


Figure 4.10: Optimized compliant mechanism design with stress constraint of $\sigma_y = 0.15$ shows only thresholded volume fractions $\tilde{\nu}$ greater than 0.8.

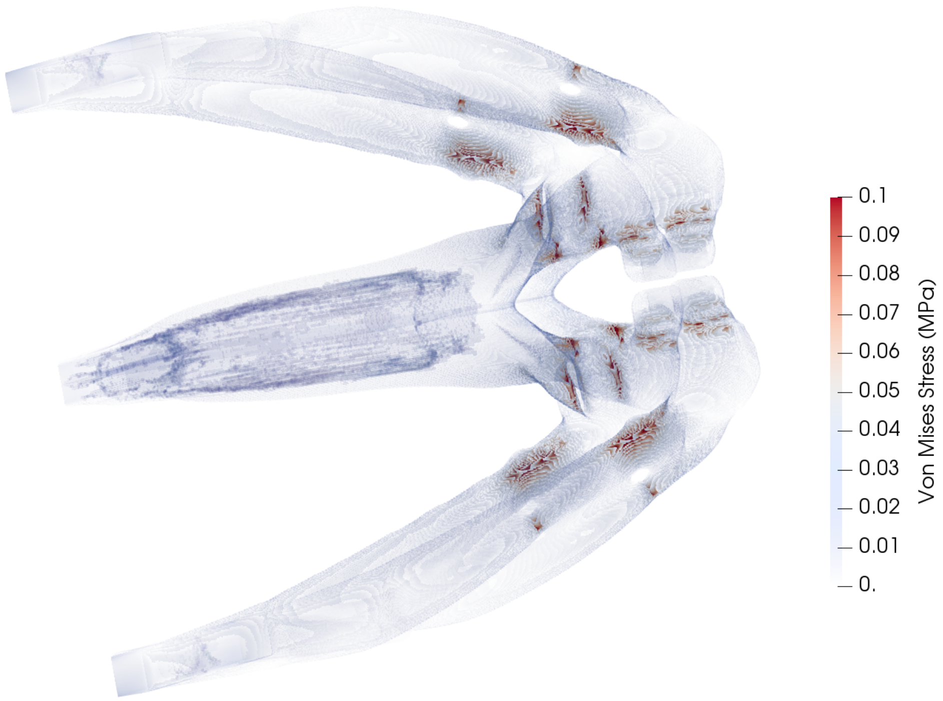
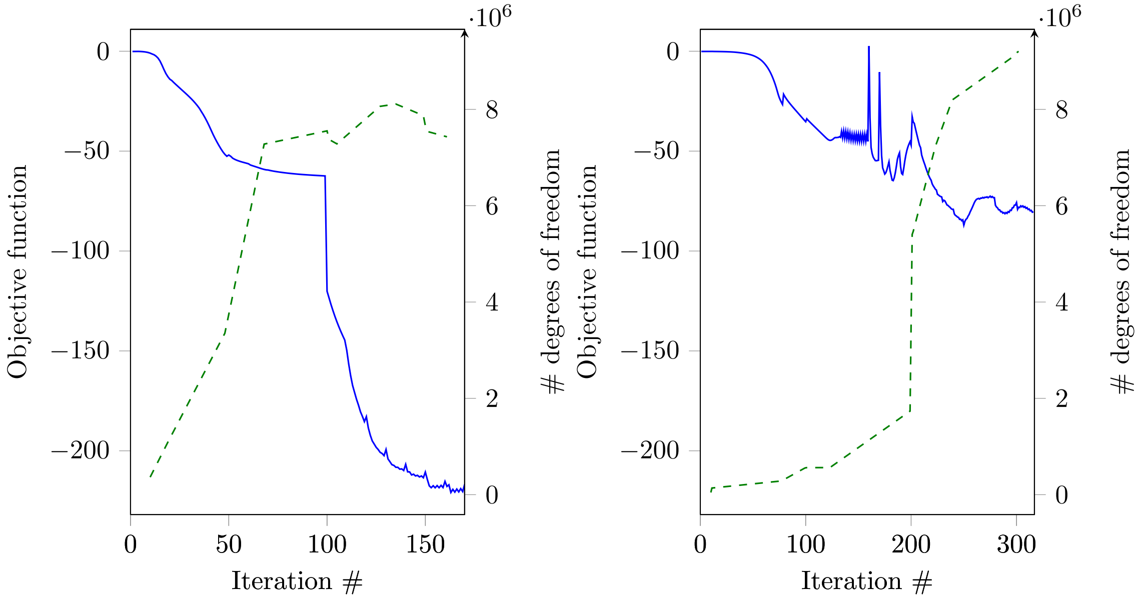


Figure 4.11: Von Mises stress field for the compliant mechanism design with stress constraint.



(a)

(b)

Figure 4.12: Cost function (solid) and number of degrees of freedom (dashed) evolutions for compliant mechanism designs without (a) and with (b) stress constraint.

geometry projection schemes [66] and using Finite Cell [80] or the CuTFEM [22] methods will hasten the computations without losing resolution.

Chapter 5

Conclusions and future directions

In this dissertation, we have applied AMR to stress constrained topology optimization problems to obtain highly accurate stress predictions and design resolutions at minimal computational expense.

In Chapter 2, a new approach to obtain accurate stress fields in topology optimization is presented for the usual element-wise uniform volume fraction discretization. By combining a volume fraction thresholding with AMR, the boundary is resolved with a high degree of accuracy which allows for accurate stress field computations. To drive the adaptive refinement, error estimates based on the accuracies of global energy norm and on the cost function approximations are used. The AMR stress computations are validated and used to optimize two dimensional designs. The primal and sensitivity analyses and optimization are implemented in a parallel environment.

Chapter 3 modifies the GCMMA optimization algorithm to operate in the L_2 function space to better accomodate meshes with elements of varying sizes. The goal of this chapter is to introduce several concepts in functional analysis to the topology optimization community and to motivate their usage. We compared the new L_2 algorithm to the originally posed Euclidean space ℓ_2 algorithm via three benchmark topology optimization problems. Our optimized designs and number of iterations using the L_2 algorithm are independent of the mesh configuration and the number of elements, while those obtained with the ℓ_2 algorithm are not.

We used the developments from Chapters 2 and 3 to optimize stress constrained three-dimensional structures with high resolution in Chapter 4. We layed out the obstacles to resolve and explained our solutions. Namely the need for an efficient filtering scheme across different levels of mesh refinement and for large scale iterative solvers for both primal and adjoint the analyses and the PDE filter.

Future work will address more rigurous mesh refinement strategies that consider the inexactness of the derivatives in the finite element approximation and still ensure convergence to a local minimum [54], [53]. These strategies have been developed only for full space optimization problems, where a posteriori error estimates for the cost function and its gradient are calculated without additional cost, and not the reduced space approach taken in this thesis, where the same error estimates require additional linear solves.

However, full space formulations are more difficult to converge and they have yet to be implemented on more challenging optimization problems like topology optimization. New problem formulations that avoid using continuation procedures to obtain sharp boundaries should be also investigated.

As for the optimization algorithm, despite obtaining mesh independent results, it proved difficult to obtain a reliable convergence criteria for stress constrained problems. It is necessary to dive deep into the stress constrained problem to develop robust convergence criteria. One possible approach would use an Augmented Lagrangian Method [45], to eliminate the need for global aggregate constraints and their high degree of nonlinearity.

The main drawback of our approach is visible in our optimized designs, where a large number of elements, and hence design parameters, are required to obtain properly resolved structures. Such goal could potentially be achieved by combining geometry projection schemes [66] with immersed finite element methods that perform an adaptive quadrature integration, i.e. the Finite Cell Method [80] or enrich the finite element space by doubling the elements cut by the geometry, i.e. the CutFEM [22], build new shape functions, i.e. the XFEM [15] or multiscale methods that enrich only the fine scale [61] to accomodate the discontinuity. The last three approaches require a crisp definition of the geometry and therefore only work with the level set method.

High order parametrizations such as the Radial Basis Functions [99], B-splines [75] or Fourier series [1] make the problem inherently well-posed and do not require a filter operation. These parametrizations define an inner product weighted by a non-diagonal mass matrix which should be included in the optimization to obtain optimal convergence, cf. Chapter 3.

Finally, the topology optimization framework developed in this thesis could be quickly applied to compliance mechanisms designs that account for fatigue. Compliance mechanisms must undergo hundreds of thousands of loading cycles in their lifetime and therefore fatigue constraints must be considered in their design. Fatigue contraints have already been accounted for in compliance designs [107], [50], [48], however, there is no literature in compliant mechanisms, let alone the large scale designs included in this thesis.

References

[1] Daniel A. White, Mark Stowell, and Daniel A. Tortorelli. Toplogical optimization of structures using fourier representations. *Structural and Multidisciplinary Optimization*, 58:1–16, 04 2018.

[2] Niels Aage, Erik Andreassen, Boyan S Lazarov, and Ole Sigmund. Giga-voxel computational morphogenesis for structural design. *Nature*, 550(7674):84, 2017.

[3] Niels Aage, Erik Andreassen, and Boyan Stefanov Lazarov. Topology optimization using petsc: An easy-to-use, fully parallel, open source topology optimization framework. *Structural and Multidisciplinary Optimization*, 51(3):565–572, 2015.

[4] Niels Aage, Erik Andreassen, and Boyan Stefanov Lazarov. Topology optimization using petsc: An easy-to-use, fully parallel, open source topology optimization framework. *Structural and Multidisciplinary Optimization*, 51(3):565–572, Mar 2015.

[5] Niels Aage and Boyan S. Lazarov. Parallel framework for topology optimization using the method of moving asymptotes. *Structural and Multidisciplinary Optimization*, 47(4):493–505, Apr 2013.

[6] Mark Ainsworth and J Tinsley Oden. *A posteriori error estimation in finite element analysis*, volume 37. John Wiley & Sons, 2011.

[7] Joe Alexandersen, Ole Sigmund, and Niels Aage. Large scale three-dimensional topology optimisation of heat sinks cooled by natural convection. *International Journal of Heat and Mass Transfer*, 100:876–891, 2016.

[8] Grégoire Allaire, Charles Dapogny, and Pascal Frey. Shape optimization with a level set based mesh evolution method. *Computer Methods in Applied Mechanics and Engineering*, page doi:10.1016/j.cma.2014.08.028, December 2014.

[9] Martin S. Alnæs, Jan Blechta, Johan Hake, August Johansson, Benjamin Kehlet, Anders Logg, Chris Richardson, Johannes Ring, Marie E. Rognes, and Garth N. Wells. The fenics project version 1.5. *Archive of Numerical Software*, 3(100), 2015.

[10] Samuel Amstutz and Antonio A. Novotny. Topological optimization of structures subject to von mises stress constraints. *Structural and Multidisciplinary Optimization*, 41(3):407–420, Apr 2010.

[11] Utkarsh Ayachit. The paraview guide: a parallel visualization application. 2015.

[12] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, Hong Zhang, and Hong Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 3.7, Argonne National Laboratory, 2016.

[13] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997.

[14] Roland Becker and Rolf Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta numerica*, 10:1–102, 2001.

[15] Ted Belytschko and Tom Black. Elastic crack growth in finite elements with minimal remeshing. *International journal for numerical methods in engineering*, 45(5):601–620, 1999.

[16] M. P. Bendsøe and O. Sigmund. Material interpolation schemes in topology optimization. *Archive of Applied Mechanics*, 69(9):635–654, Nov 1999.

[17] Blaise Bourdin. Filters in topology optimization. *International Journal for Numerical Methods in Engineering*, 50(9):2143–2158, 2001.

[18] Jan H Brandts, Sergey Korotov, and Michal Křížek. The discrete maximum principle for linear simplicial finite element approximations of a reaction–diffusion problem. *Linear Algebra and its Applications*, 429(10):2344–2357, 2008.

[19] Matteo Bruggi. On an alternative approach to stress constraints relaxation in topology optimization. *Structural and multidisciplinary optimization*, 36(2):125–141, 2008.

[20] Matteo Bruggi and Marco Verani. A fully adaptive topology optimization algorithm with goal-oriented error control. *Computers & Structures*, 89(15):1481 – 1493, 2011.

[21] Tyler E. Bruns and Daniel A. Tortorelli. Topology optimization of non-linear elastic structures and compliant mechanisms. *Computer Methods in Applied Mechanics and Engineering*, 190(26):3443 – 3459, 2001.

[22] Erik Burman, Susanne Claus, Peter Hansbo, Mats G. Larson, and André Massing. Cutfem: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501.

[23] Erik Burman and Alexandre Ern. Discrete maximum principle for galerkin approximations of the laplace operator on arbitrary meshes.

[24] Zhiqiang Cai, Cuiyu He, and Shun Zhang. Improved {ZZ} a posteriori error estimators for diffusion problems: Conforming linear elements. *Computer Methods in Applied Mechanics and Engineering*, 313:433 – 449, 2017.

[25] GD Cheng and X Guo. A note on star shape feasible domain in structural topology optimization. In *Proceedings of the Second China-Japan Symposium on Optimization of Structural and Mechanical Systems, Beijiag*, 1995.

[26] GD Cheng and Xiao Guo. ε-relaxed approach in structural topology optimization. *Structural and Multidisciplinary Optimization*, 13(4):258–266, 1997.

[27] Gengdong Cheng and Zheng Jiang. Study on topology optimization with stress constraints. *Engineering Optimization*, 20(2):129–148, 1992.

[28] Geuzaine Christophe and Remacle Jean-François. Gmsh: A 3-d finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[29] João Carlos Arantes Costa Jr, Marcelo Krajnc Alves, and Hazim Ali Al-Qureshi. A stress based h-adaptive topology optimization method. *Proceedings of COBEM 2005 : 18th international congress of mechanical engineering*, 2005.

[30] Daniel M De Leon, Joe Alexandersen, Jun SO Fonseca, and Ole Sigmund. Stress-constrained topology optimization for compliant mechanism design. *Structural and Multidisciplinary Optimization*, 52(5):929–943, 2015.

[31] Daniel M. De Leon, Joe Alexandersen, Jun S. O. Fonseca, and Ole Sigmund. Stress-constrained topology optimization for compliant mechanism design. *Structural and Multidisciplinary Optimization*, 52(5):929–943, Nov 2015.

[32] Alp Dener, Todd Munson, Jason Sarich, Stefan Wild, Steven Benson, and Lois Curfman McInnes. Tao 3.9 users manual. Technical Report ANL/MCS-TM-322, Mathematics and Computer Science Division, Argonne National Laboratory, 2018. http://www.mcs.anl.gov/tao.

[33] Willy Dörfler. A convergent adaptive algorithm for poisson's equation. *SIAM Journal on Numerical Analysis*, 33(3):1106–1124, 1996.

[34] Pierre Duysinx and Martin P Bendsøe. Topology optimization of continuum structures with local stress constraints. *International journal for numerical methods in engineering*, 43(8):1453–1478, 1998.

[35] Anton Evgrafov, Cory J Rupp, Kurt Maute, and Martin L Dunn. Large-scale parallel topology optimization using a dual-primal substructuring solver. *Structural and Multidisciplinary Optimization*, 36(4):329–345, 2008.

[36] Robert Eymard, Thierry Gallouët, and Raphaèle Herbin. Finite volume methods. *Handbook of numerical analysis*, 7:713–1018, 2000.

[37] Arun L. Gain and Glaucio H. Paulino. A critical comparative assessment of differential equation-driven methods for structural topology optimization. *Structural and Multidisciplinary Optimization*, 48(4):685–710, Oct 2013.

[38] Vikram Vinod Garg. *Coupled flow systems, adjoint techniques and uncertainty quantification.* PhD thesis, The University of Texas at Austin, 2012.

[39] Philip E. Gill, Walter Murray, and Michael A. Saunders. User's guide for snopt version 7: Software for large-scale nonlinear programming. `http://www.cam.ucsd.edu/~peg/papers/sndoc7.pdf`, 2015.

[40] Carlos Eduardo Marcos Guilherme and Jun Sérgio Ono Fonseca. Topology optimization of continuum structures with epsilon-relaxed stress constraints. *International Symposium on Solid Mechanics (2007 mar. 05-07 : São Paulo, SP).*, 2007.

[41] Xu Guo, Wei Sheng Zhang, Michael Yu Wang, and Peng Wei. Stress-related topology optimization via level set approach. *Computer Methods in Applied Mechanics and Engineering*, 200(47):3439 – 3452, 2011.

[42] Xu Guo, Weisheng Zhang, and Wenliang Zhong. Stress-related topology optimization of continuum structures involving multi-phase materials. *Computer Methods in Applied Mechanics and Engineering*, 268:632 – 655, 2014.

[43] Robert B Haber, Chandrashekhar S Jog, and Martin P Bendsøe. A new approach to variable-topology shape design using a constraint on perimeter. *Structural and Multidisciplinary Optimization*, 11(1):1–12, 1996.

[44] B. Hassani and E. Hinton. A review of homogenization and topology optimization iii—topology optimization using optimality criteria. *Computers & Structures*, 69(6):739 – 756, 1998.

[45] Magnus R Hestenes. Multiplier and gradient methods. *Journal of optimization theory and applications*, 4(5):303–320, 1969.

[46] Michael Hintermüller and Michael Hinze. Moreau-Yosida regularization in state constrained elliptic control problems: Error estimates and parameter adjustment. *SIAM Journal on Numerical Analysis*, 47(3):1666–1683, 2009.

[47] Erik Holmberg, Bo Torstenfelt, and Anders Klarbring. Stress constrained topology optimization. *Structural and Multidisciplinary Optimization*, 48(1):33–47, Jul 2013.

[48] Erik Holmberg, Bo Torstenfelt, and Anders Klarbring. Fatigue constrained topology optimization. *Structural and Multidisciplinary Optimization*, 50(2):207–219, 2014.

[49] Kristian Ejlebjerg Jensen. Solving stress and compliance constrained volume minimization using anisotropic mesh adaptation, the method of moving asymptotes and a global p-norm. *Structural and Multidisciplinary Optimization*, 54(4):831–841, 2016.

[50] Seung Hyun Jeong, Dong-Hoon Choi, and Gil Ho Yoon. Fatigue and static failure considerations using a topology optimization method. *Applied Mathematical Modelling*, 39(3-4):1137–1162, 2015.

[51] B. S. Kirk, J. W. Peterson, R. H. Stogner, and G. F. Carey. `libMesh`: A C++ Library for Parallel Adaptive Mesh Refinement/Coarsening Simulations. *Engineering with Computers*, 22(3–4):237–254, 2006.

[52] Uri Kirsch. On singular topologies in optimum structural design. *Structural optimization*, 2(3):133–142, 1990.

[53] Drew P Kouri, Matthias Heinkenschloss, Denis Ridzal, and Bart G van Bloemen Waanders. A trust-region algorithm with adaptive stochastic collocation for pde optimization under uncertainty. *SIAM Journal on Scientific Computing*, 35(4):A1847–A1879, 2013.

[54] Drew P Kouri, Matthias Heinkenschloss, Denis Ridzal, and Bart G van Bloemen Waanders. Inexact objective function evaluations in a trust-region algorithm for pde-constrained optimization under uncertainty. *SIAM Journal on Scientific Computing*, 36(6):A3011–A3029, 2014.

[55] B. S. Lazarov and O. Sigmund. Filters in topology optimization based on helmholtz-type differential equations. *International Journal for Numerical Methods in Engineering*, 86(6):765–781, 2011.

[56] Chau Le, Julian Norato, Tyler Bruns, Christopher Ha, and Daniel Tortorelli. Stress-based topology optimization for continua. *Structural and Multidisciplinary Optimization*, 41(4):605–620, Apr 2010.

[57] Chau Le, Julian Norato, Tyler Bruns, Christopher Ha, and Daniel Tortorelli. Stress-based topology optimization for continua. *Structural and Multidisciplinary Optimization*, 41(4):605–620, Apr 2010.

[58] Mark K Leader, Ting Wei Chin, and Graeme Kennedy. High resolution topology optimization of aerospace structures with stress and frequency constraints. In *2018 Multidisciplinary Analysis and Optimization Conference*, page 4056, 2018.

[59] Haojie Lian, Asger N. Christiansen, Daniel A. Tortorelli, Ole Sigmund, and Niels Aage. Combined shape and topology optimization for minimization of maximal von mises stress. *Structural and Multidisciplinary Optimization*, 55(5):1541–1557, May 2017.

[60] LLNL. *hypre: High Performance Preconditioners*. Lawrence Livermore National Laboratory, 2018.

[61] Arif Masud and Ahmad A. Al-Naseem. Variationally derived discontinuity capturing methods: Fine scale models with embedded weak and strong discontinuities. *Computer Methods in Applied Mechanics and Engineering*, 340:1102 – 1134, 2018.

[62] Matlab optimization toolbox, 2016a. The MathWorks, Natick, MA, USA.

[63] K. Maute and E. Ramm. Adaptive topology optimization. *Structural optimization*, 10(2):100–112, Oct 1995.

[64] Luís Renato Meneghelli and Eduardo Lenz Cardoso. *Design of Compliant Mechanisms with Stress Constraints Using Topology Optimization*, pages 35–48. Springer International Publishing, Cham, 2013.

[65] Dirk Munro and Albert Groenwold. Local stress-constrained and slope-constrained sand topology optimisation. *International Journal for Numerical Methods in Engineering*, 110(5):420–439.

[66] Ahmad R Najafi, Masoud Safdari, Daniel A Tortorelli, and Philippe H Geubelle. Shape optimization using a nurbs-based interface-enriched generalized fem. *International Journal for Numerical Methods in Engineering*, 111(10):927–954, 2017.

[67] Alexandre Nana, Jean-Christophe Cuillière, and Vincent Francois. Towards adaptive topology optimization. *Advances in Engineering Software*, 100:290–307, 2016.

[68] H. Nguyen-Xuan. A polytree-based adaptive polygonal finite element method for topology optimization. *International Journal for Numerical Methods in Engineering*, 110(10):972–1000, 2017. nme.5448.

[69] John Tinsley Oden and Serge Prudhomme. Goal-oriented error estimation and adaptivity for the finite element method. *Computers & mathematics with applications*, 41(5-6):735–756, 2001.

[70] JS Ovall. The dangers to avoid when using gradient recovery methods for finite element error estimation and adaptivity. Technical report, Technical Report 6, Max Planck Institute for Mathematics in the Sciences, 2006.

[71] Ajit Panesar, David Brackett, Ian Ashcroft, Ricky Wildman, and Richard Hague. Hierarchical remeshing strategies with mesh mapping for topology optimisation. *International Journal for Numerical Methods in Engineering*, 111(7):676–700, 2017. nme.5488.

[72] J París, F Navarrina, I Colominas, and M Casteleiro. Improvements in the treatment of stress constraints in structural topology optimization problems. *Journal of computational and applied mathematics*, 234(7):2231–2238, 2010.

[73] Youn Kyu Park. *Extensions of optimal layout design using the homogenization method.* PhD thesis, University of Michigan, 1995.

[74] Joakim Petersson and Ole Sigmund. Slope constrained topology optimization. *International Journal for Numerical Methods in Engineering*, 41(8):1417–1434, 1998.

[75] Xiaoping Qian. Topology optimization in b-spline space. *Computer Methods in Applied Mechanics and Engineering*, 265:15–35, 2013.

[76] Denis Ridzal and Drew Philip Kouri. Rapid optimization library. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2017.

[77] GIN Rozvany and T Birker. On singular topologies in exact layout optimization. *Structural and Multidisciplinary Optimization*, 8(4):228–235, 1994.

[78] Miguel A. Salazar de Troya and Daniel A. Tortorelli. Adaptive mesh refinement in stress-constrained topology optimization. *Structural and Multidisciplinary Optimization*, Oct 2018.

[79] A. Saxena and G. K. Ananthasuresh. Topology optimization of compliant mechanisms with strength considerations*. *Mechanics of Structures and Machines*, 29(2):199–221, 2001.

[80] Dominik Schillinger and Martin Ruess. The finite cell method: A review in the context of higher-order structural analysis of cad and image-based geometric models. *Archives of Computational Methods in Engineering*, 22(3):391–455, Jul 2015.

[81] Tobias Schwedes, Simon Funke, and D Ham. An iteration count estimate for a mesh-dependent steepest descent method based on finite elements and riesz inner product representation. 06 2016.

[82] Tobias Schwedes, D Ham, Simon W. Funke, and Matthew Piggott. *Mesh dependence in PDE-constrained optimisation.* 07 2017.

[83] Ashesh Sharma and Kurt Maute. Stress-based topology optimization using spatial gradient stabilized XFEM. *Structural and Multidisciplinary Optimization*, 57(1):17–38, January 2018.

[84] Magne A Nordaas Simon W Funke. Optizelle: An open source software library designed to solve general purpose nonlinear optimization problems. https://github.com/funsim/moola, 2016.

[85] Roman Stainko. An adaptive multilevel approach to the minimal compliance problem in topology optimization. *Communications in Numerical Methods in Engineering*, 22(2):109–118, 2006.

[86] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*, 22(2):116–124, Sep 2001.

[87] M. Stolpe and K. Svanberg. An alternative interpolation scheme for minimum compliance topology optimization. *Structural and Multidisciplinary Optimization*, 22(2):116–124, Sep 2001.

[88] Krister Svanberg. Mma and gcmma – two methods for nonlinear optimization. https://people.kth.se/~krille/mmagcmma.pdf.

[89] Krister Svanberg. The method of moving asymptotes—a new method for structural optimization. *International journal for numerical methods in engineering*, 24(2):359–373, 1987.

[90] Krister Svanberg. A class of globally convergent optimization methods based on conservative convex separable approximations. *SIAM Journal on Optimization*, 12(2):555–573, 2002.

[91] Henrik Svärd. Interior value extrapolation: a new method for stress evaluation during topology optimization. *Structural and Multidisciplinary Optimization*, 51(3):613–629, Mar 2015.

[92] G Sved and Z Ginos. Structural optimization under multiple loading. *International Journal of Mechanical Sciences*, 10(10):803–805, 1968.

[93] Michael Ulbrich and Stefan Ulbrich. Primal-dual interior-point methods for pde-constrained optimization. *Mathematical Programming*, 117(1-2):435–485, 2009.

[94] Rüdiger Verfürth. A review of a posteriori error estimation techniques for elasticity problems. *Computer Methods in Applied Mechanics and Engineering*, 176(1-4):419–440, 1999.

[95] Rüdiger Verfürth. *A posteriori error estimation techniques for finite element methods.* Oxford University Press, Oxford, UK, 2013.

[96] Andreas Wächter and Lorenz T. Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 106(1):25–57, Mar 2006.

[97] Mathias Wallin, Matti Ristinmaa, and Henrik Askfelt. Optimal topologies derived from a phase-field method. *Structural and Multidisciplinary Optimization*, 45(2):171–183, Feb 2012.

[98] Fengwen Wang, Boyan Stefanov Lazarov, and Ole Sigmund. On projection methods, convergence and robust formulations in topology optimization. *Structural and Multidisciplinary Optimization*, 43(6):767–784, Jun 2011.

[99] Shengyin Wang and Michael Yu Wang. Radial basis functions and level set method for structural topology optimization. *International journal for numerical methods in engineering*, 65(12):2060–2090, 2006.

[100] Shun Wang, Eric de Sturler, and Glaucio H Paulino. Dynamic adaptive mesh refinement for topology optimization. *arXiv preprint arXiv:1009.4975*, 2010.

[101] Shun Wang, Eric de Sturler, and Glaucio H. Paulino. Large-scale topology optimization using preconditioned krylov subspace methods with recycling. *International Journal for Numerical Methods in Engineering*, 69(12):2441–2468.

[102] Yiqiang Wang, Zhan Kang, and Qizhi He. Adaptive topology optimization with independent error control for separated displacement and density fields. *Computers & Structures*, 135:50 – 61, 2014.

[103] Qi Xia, Tielin Shi, Shiyuan Liu, and Michael Yu Wang. A level set solution to the stress-based structural shape and topology optimization. *Computers & Structures*, 90:55 – 64, 2012.

[104] RJ Yang and CJ Chen. Stress-based topology optimization. *Structural optimization*, 12(2-3):98–105, 1996.

[105] Joseph Young. Optizelle: An open source software library designed to solve general purpose nonlinear optimization problems. `http://www.optimojoe.com`, 2018.

[106] Warren Clarence Young and Richard Gordon Budynas. *Roark's formulas for stress and strain*, volume 7. McGraw-Hill New York, 2002.

[107] Shanglong Zhang, Chau Le, Arun L. Gain, and Julián A. Norato. Fatigue-based topology optimization with non-proportional loads. *Computer Methods in Applied Mechanics and Engineering*, 345:805 – 825, 2019.

[108] Wei Sheng Zhang, Xu Guo, Michael Yu Wang, and Peng Wei. Optimal topology design of continuum structures with stress concentration alleviation via level set method. *International Journal for Numerical Methods in Engineering*, 93(9):942–959, 2013.

[109] J. Carsten Ziems and Stefan Ulbrich. Adaptive multilevel inexact sqp methods for pde-constrained optimization. *SIAM Journal on Optimization*, 21(1):1–40, 2011.

[110] J Carsten Ziems and Stefan Ulbrich. Adaptive multilevel inexact sqp methods for pde-constrained optimization. *SIAM Journal on Optimization*, 21(1):1–40, 2011.

[111] Olgierd Cecil Zienkiewicz and Jian Zhong Zhu. The superconvergent patch recovery and a posteriori error estimates. part 1: The recovery technique. *International Journal for Numerical Methods in Engineering*, 33(7):1331–1364, 1992.

[112] Olgierd Cecil Zienkiewicz and Jian Zhong Zhu. The superconvergent patch recovery and a posteriori error estimates. part 2: Error estimates and adaptivity. *International Journal for Numerical Methods in Engineering*, 33(7):1365–1382, 1992.