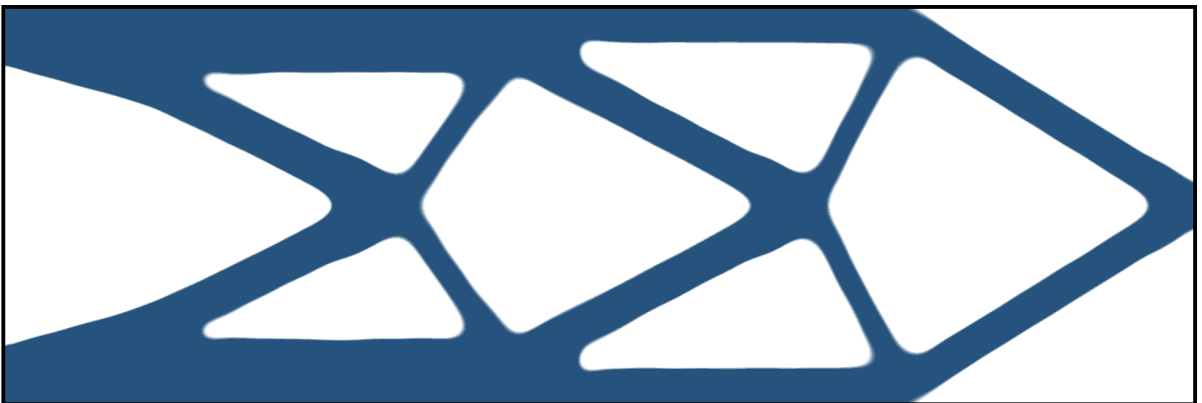
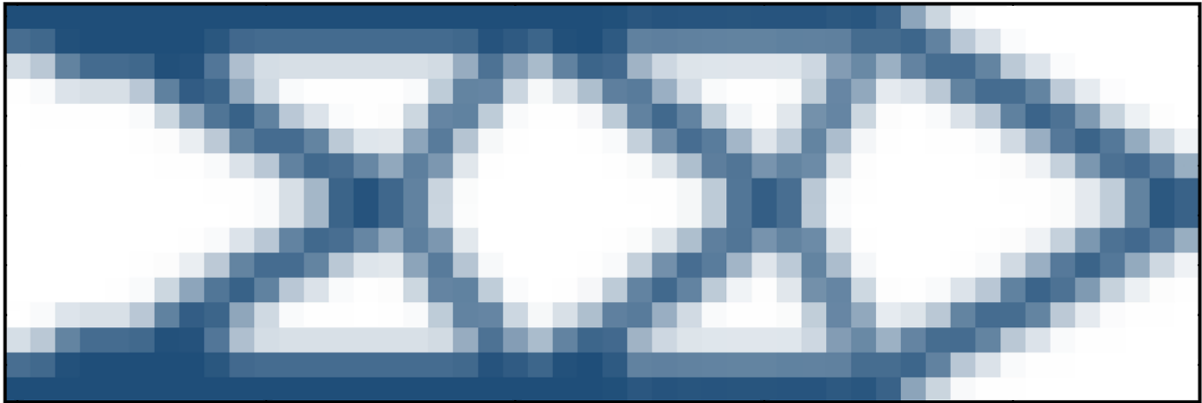


Post-Processing of Topology Optimized Results

A method for retrieving smooth and crisp geometries

M.K. Swierstra



Post-Processing of Topology Optimized Results

A method for retrieving smooth and crisp geometries

by

M.K. Swierstra

in partial fulfilment of the requirements for the degree of

Master of Science
in Mechanical Engineering

at the Delft University of Technology,
to be defended publicly on Friday September 1, 2017 at 3:00 PM.

Student number:	4160363	
Project duration:	November 1, 2016 – September 1, 2017	
Thesis committee:	Dr. ir. M. Langelaar,	TU Delft, supervisor
	MSc. D. Gupta,	TU Delft, supervisor
	Dr. ir. M. A. N. Hendriks,	TU Delft



Abstract

Structural design optimization is the process of obtaining an optimized structure while satisfying a set of criteria. The process can be divided into three stages: topology optimization (stage one), geometry extraction (stage two) and shape optimization (stage three). The last two stages are regarded to be the post-processor of topology optimized results and this MSc. Thesis proposes a new method for this part. At stage one, topology optimized results show unwanted features, namely jagged boundaries (i.e. poor smoothness) and intermediate densities (i.e. poor crispness). The post-processor should overcome these unwanted properties.

During post-processing the geometry is described implicitly by a Level Set Function (LSF). The zero-level contour of the LSF describes the actual geometry. The LSF is constructed by summing a set of Radial Basis Functions (RBFs) each multiplied with a weight, resulting in a smooth summation. At stage two, a geometry is extracted by setting up and solving a set of equations linking the RBFs to the densities obtained at stage one.

At stage three, a shape optimization is done to compensate the loss of structural performance, which resulted from translating the densities to an LSF at stage two. The geometry described by the LSF does not match the mesh created for stage one. A fictitious domain method called the Finite Cell Method (FCM) is used to perform a structural analysis on the non-matching mesh. A sensitivity analysis is done to provide gradient information to the gradient-based optimizer, the Method of Moving Asymptotes (MMA). Controlling the slope of the LSF is needed to make sure the sensitivities do not become zero throughout the domain. The maximum possible slope of the LSF can be fixed by setting a bound on the weights of the RBFs. Furthermore, intermediate densities are penalized such that these provide relatively low stiffness compared to its material use.

Several case studies are done using the proposed method. The post-processor: (1) improves the smoothness due to the use of the smooth LSF, (2) decreases the amount of intermediate densities by an average factor of 5.5 and (3) achieves an average 10% improve in performance between stage two and three. The computation time is strongly problem dependent, test cases are either: slower, equally fast or faster than the topology optimization of stage one.

Acknowledgements

Precisely six years ago I was preparing myself for the OWee, the introduction week in Delft, at precisely the same place as where I am writing the final part of my MSc. Thesis report on 'Post-Processing of Topology Optimized Results'. In these six years I did some relaxing and sporting at DSZ WAVE, visited New Zealand and above all I did a lot of studying.

The acknowledgements page allows me to thank people for their help and support during this MSc. Thesis, which I will, only after describing the road to my MSc. Thesis. I mean, this is the only part of the report where I can do whatever I want, so I will. As I said already, I came to Delft six years ago starting my BSc. Mechanical Engineering. I also joined DSZ WAVE to play water polo. To give something back to the club for all the fun it gave me, I was a board member for one year. During the same year I finished my BSc. degree. This was a busy period but I managed, mainly because of the help of Ronald, Timo, Patrick and Sander. After that I did start my Masters, first some courses, then an internship at Femto Engineering and finally my MSc. Thesis project. During this project I got the help of Dr. ir. M. Langelaar and MSc. D. Gupta. I would like to thank both of them for allowing me to just randomly walk into their offices and taking their precious time. I sincerely hope Deepak is going to finish his PhD despite me taking his time. The discussions we had really improved my work. A note for other people reading this, I also sometimes scheduled a real meeting instead of crashing my supervisors their offices.

I would like to thank my family, and especially Heit and Mem, for providing me a place where I could go in the weekend without having any obligations like cooking and cleaning. Those weekends I could charge myself to get ready for a new week of studying. Furthermore, they were my main sponsor, together with 'Ome DUO', funding this entire project. I think that is also a point worth mentioning. Besides visiting family in Fryslân, there were also a bunch of friends sticking around in that region. Their presence always resulted in good fun and a nice distraction from life in Delft.

Where family and friends were there when I travelled north, there was also support in Delft. Fieke, thank you for thinking along with the problems I faced during this project and for just being there. Also everyone at DSZ WAVE I celebrated another promotion to a higher water polo league with, thank you.

*M.K. Swierstra
Boalsert, August 2017*

Nomenclature

List of acronyms

CAD	Computer Aided Design
DOF	Degrees Of Freedom
FCM	Finite Cell Method
FEM	Finite Element Method
FG	Fixed Grid
GFEM	Generalized Finite Element Method
hFEM	h-version of the Finite Element Method
LSF	Level Set Function
LSM	Level Set Method
MMA	Method of Moving Asymptotes
NURBS	Non-Uniform Rational Basis Splines
pFEM	p-version of the Finite Element Method
RBF	Radial Basis Function
SIMP	Solid Isotropic Material with Penalization
XFEM	eXtended Finite Element Method

List of symbols

In this work, a lower- and upper-case letter represents a scalar, a lower-case bold letter represents a vector and an upper-case bold letter represents a matrix.

B	bandwidth
$\mathbf{b}$	body force
C	compliance
c_{ri}	number of reinitialization cycles during shape optimization
E	Young's modulus
F	point load
F_q	distributed load
$\mathbf{f}$	force vector
g	volume fraction constraint
$\tilde{H}$	approximate Heaviside function
$\mathbf{K}$	stiffness matrix
L	length
N	basis function
n_{el}	number of elements
$n_{\max}$	number of iterations
q	number of quadtree refinements
P	polynomial order
p	penalization factor of intermediate densities
r	radius of Radial Basis Function
s_i	design variables Level Set Method
t	thickness
$\mathbf{u}$	displacement vector
u^h	approximated displacement field
U	strain energy
V	volume
x	horizontal coordinate
y	vertical coordinate

α	move limit parameter
ε	convergence tolerance
ε_r	relative error in the energy norm
η	local vertical coordinate
κ	approximate Heaviside parameter controlling steepness
λ	Lagrange multiplier
ν	Poisson's ratio
ξ	local horizontal coordinate
ρ	density
σ	stress tensor
ϕ	Level Set Function

Contents

Abstract	iii
Acknowledgements	v
Nomenclature	vii
Contents	x
1 Introduction	1
1.1 Structural design optimization	1
1.1.1 Stage 1 - Topology Optimization	1
1.1.2 Stage 2 - Geometry extraction	3
1.1.3 Stage 3 - Shape optimization.	5
1.2 Problem formulation	5
1.2.1 Goals.	6
1.2.2 Methodology.	6
1.3 Tools and implementation	6
1.4 Outline report.	7
2 Geometry extraction	9
2.1 Geometry description.	9
2.2 Level set basis functions	9
2.3 Nodal densities	10
2.4 Level Set Description	11
2.5 Projecting LSF to a material distribution	11
3 Structural analysis	13
3.1 Finite Element Method	13
3.2 Analyse geometry on a non-matching mesh	13
3.2.1 Adaptive meshing	13
3.2.2 Fictitious domain methods	14
3.3 p-version of the Finite Element Method (pFEM)	14
3.3.1 Hierarchical shape functions.	14
3.3.2 Convergence pFEM	15
3.4 Adaptive numerical integration	15
3.5 Example analyses	17
3.5.1 Axially loaded bar with thinner mid-section	17
3.5.2 Rectangular plate with circular hole	18
3.5.3 Geometry obtained at stage two for the MBB beam case.	19
4 Sensitivity analysis	21
4.1 Review of sensitivity analysis methods	21
4.2 Sensitivities for shape optimization.	22
4.2.1 Sensitivity analysis objective - compliance.	22
4.2.2 Sensitivity analysis constraint - volume fraction	23
4.2.3 Finite Difference check	23
4.3 Link with SIMP sensitivities.	23
5 LSF slope control	25
5.1 Possible slope control methods	26
5.2 Maximum slope RBF summation	26
5.2.1 Dependency maximum slope on bandwidth.	27
5.3 Calculate Heaviside variable κ	28

6	Shape optimization	29
6.1	Optimizers	29
6.1.1	Method of Moving Asymptotes.	29
6.2	Procedure stage three	30
6.2.1	Initial geometry	31
6.2.2	Quadtree integration band.	31
6.2.3	Reinitialization.	33
6.2.4	Result	33
6.3	Final design to CAD	33
7	Case studies	35
7.1	General optimization settings case studies	36
7.2	MBB beam	37
7.2.1	Results MBB beam	37
7.2.2	Performance stage three for MMB beam	37
7.2.3	Check sensitivities	38
7.3	Cantilever beam	39
7.3.1	Results cantilever beam	39
7.3.2	Performance stage three for cantilever beam.	39
7.4	Bridge with distributed load	41
7.4.1	Results bridge with distributed load	41
7.4.2	Performance stage three for bridge with distributed load	42
7.4.3	Topological change	43
7.5	Conclusions case studies	43
8	Conclusions and recommendations	45
8.1	Goals post-processing method	45
8.1.1	Usable final geometry	45
8.1.2	Simple and all-round tool	45
8.1.3	Fast	46
8.2	Other findings.	46
8.2.1	Structural analysis	46
8.2.2	LSF slope control	46
8.2.3	Penalization vs. design freedom	47
8.3	Recommendations	47
8.3.1	Extension to 3D	47
8.3.2	Quadtree integration band.	47
8.3.3	Convergence behaviour shape optimization	47
8.3.4	RAM usage.	47
8.3.5	Minimum feature size	48
8.3.6	Stress constraints	48
A	Algorithms	49
B	Performance stage three	53
	Bibliography	57

Introduction

This chapter is an introduction to the MSc. Thesis topic 'Post-Processing of Topology Optimized Results'. First the background of the topic is discussed using the literature available. After that, the scope of the project is defined and at last an outline of the report is given.

1.1. Structural design optimization

Structural design optimization is the process of obtaining a most efficient or optimal structure while satisfying a certain set of criteria. In this process mathematical optimization techniques are applied to structural problems to optimize the shapes or sizes of structural members of a design. Or even more general, these techniques can also be used to find the optimal placement of material, also called the topology, of a structure. In that case, no initial structure needs to be defined.

The structural design optimization process can be divided into three stages according to Papalambros and Chirehdast [25], see figure 1.1:

1. topology optimization;
2. extracting a smooth and crisp geometry from the topology optimized result;
3. shape optimization in which the contours/shape of the geometry are optimized.

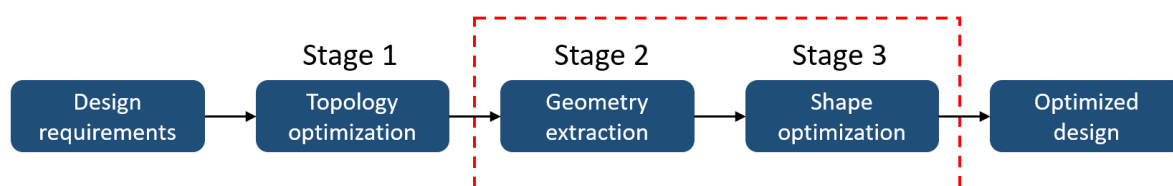


Figure 1.1: The three staged process schematically shown for structural design optimization. The dotted red box covers the post-processing of topology optimized results, which is the combination of stage two and three.

The main reason for the existence of a stage two is the use of a different geometric description between stage one and three, e.g. splines in stage three and finite elements in stage one (Seo et al. [28]). This MSc. Thesis focusses on the post-processing of results obtained at stage one, hence the dotted red box in figure 1.1 including the other stages.

1.1.1. Stage 1 - Topology Optimization

The first stage of the design optimization process is the topology optimization. These optimization procedures are typically very free in initial design requirements.

Density Based Methods

Density based methods can be divided in the homogenization method, introduced by Bendsøe and Kikuchi [4], and the Solid Isotropic Material with Penalization (SIMP) method, independently introduced by Bendsøe [3] and Zhou and Rozvany [45]. Only the SIMP method is discussed further in this section because it is more commonly used.

In the SIMP method one defines a density in each element of the domain as its design variable that can vary between 0 (void) and 1 (solid). The stiffness properties of an element are linked to this density using the following power law:

$$E = \rho^p \cdot E_0 \quad (1.1)$$

where p is the penalization parameter and E_0 is the Young's modulus for the isotropic material. Equation 1.1 penalizes intermediate densities as these provide relatively low stiffness, as can be seen in figure 1.2. Different penalization laws exist besides equation 1.1, such as the 'Rational Approximation of Material Properties' (RAMP) function proposed by Stolpe and Svanberg [33].

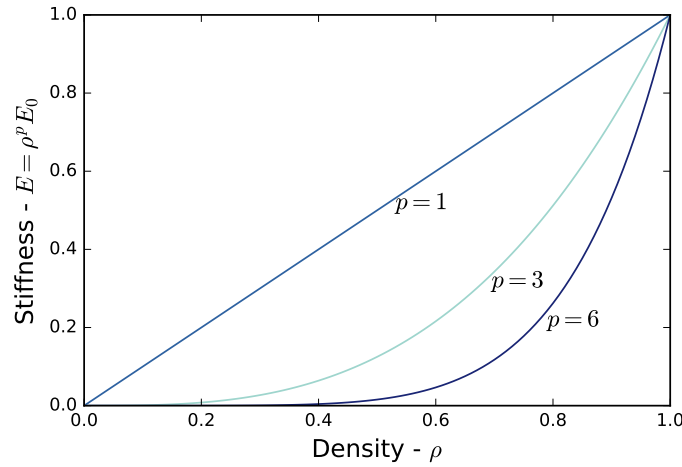


Figure 1.2: Penalization of intermediate density values as done in equation 1.1, shown for different values of p .

Filtering is needed to prevent the so-called 'checkerboarding'. A downside of filtering is that the topology optimized results show more intermediate densities (Bendsøe and Sigmund [5]). Guest et al. [12] did suggest a method using a Heaviside function to get rid off these blurred boundaries. Unfortunately, this method needs a continuation scheme which adds many optimization iterations. A typical topology optimized result using the SIMP method is shown in figure 1.3. Clearly, the topology optimized result shows these intermediate densities as a result of the filtering but also another unwanted feature, namely jagged boundaries. The SIMP method in combination with some filtering technique is the most commonly used topology optimization method for stage one according to the review paper of Deaton and Grandhi [9] for example.

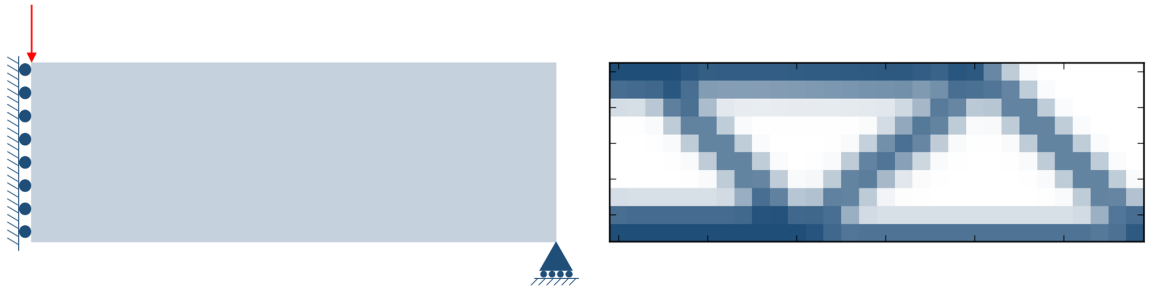


Figure 1.3: Typical topology optimized result using the SIMP method. The example is called the MMB beam. The left image shows the boundary conditions: a load on the upper left corner, a vertical support on the bottom right corner and a horizontal support on the entire left edge. The right image is the topology optimized result for minimum compliance (i.e. maximum stiffness), for which only 40% of the design domain was allowed as material, mesh of 30x10 elements, penalization $p = 3$ and a filter radius of 1.5 elements.

Level Set Methods

The first Level Set Methods (LSMs) were introduced by Sethian and Wiegmann [29] and Wang et al. [42]. The previously mentioned SIMP method uses element based design variables. LSMs on the other hand define the interfaces implicitly by iso-contours of a Level Set Function (LSF), see figure 1.4. This implicit function allows a crisp and smooth description of the boundaries. Changing the LSF alters the shape and possibly the topology of the design domain. LSMs are closely related to shape optimization (stage three) but rather than representing the geometry explicitly using e.g. splines, LSMs use an implicit description allowing for easy treatment of geometry changes. An overview of different LSMs is given by van Dijk et al. [38].

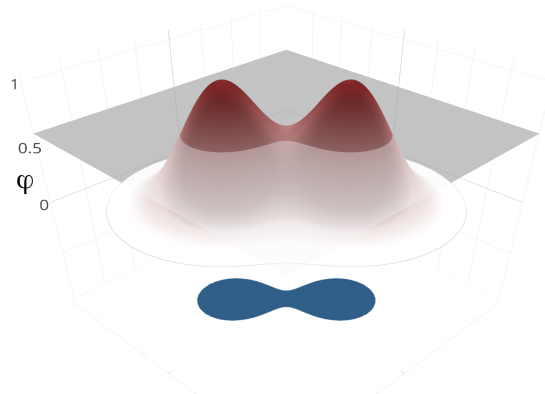


Figure 1.4: Example of the LSM, showing the LSF (red relief) being cut off on the 0.6 iso-contour. The resulting geometry is projected below in blue.

Note that LSMs do not fit the structural design optimization process introduced by Papalambros and Chirehdast [25] because LSMs do not need post-processing of the topology optimized results. Typical design from the LSM are satisfactory already such that stage two and three can be omitted.

A disadvantage of LSMs is the difficulty of introducing new holes in the design. To do so, topological derivatives are needed which can be hard to derive. Also the structural analysis of the design described by an LSF is not straightforward because the LSF and the analysis its mesh are decoupled, which is another disadvantage. The last main disadvantage of LSMs is controlling the LSF such that its gradient stays within certain bounds. This is needed to do a good sensitivity analysis for the optimization. Altogether, the SIMP method is more widely used mainly because of its simplicity compared to the LSMs.

1.1.2. Stage 2 - Geometry extraction

Since the SIMP method is the most commonly used method for stage one, the approaches for stage two are related to extracting geometries from topology optimized results obtained using the SIMP method. The geometry extraction stage can be divided in several approaches. These approaches should overcome the previously mentioned unwanted features of the SIMP method: jagged boundaries and intermediate densities. The following list is composed from methods found in literature:

- Image processing techniques
- Model reconstruction methods
- Iso-density contour methods

Figure 1.5 provides a graphical overview of the methods used for stage two.

Image processing techniques

Papalambros and Chirehdast [25] proposed an image processor consisting of three steps:

1. Thresholding converts the topology optimized result, with element densities ranging from 0 to 1, to a binary image.
2. Regions and boundaries are extracted.
3. Curve smoothing on selected control points is performed using B-splines and linear segments.

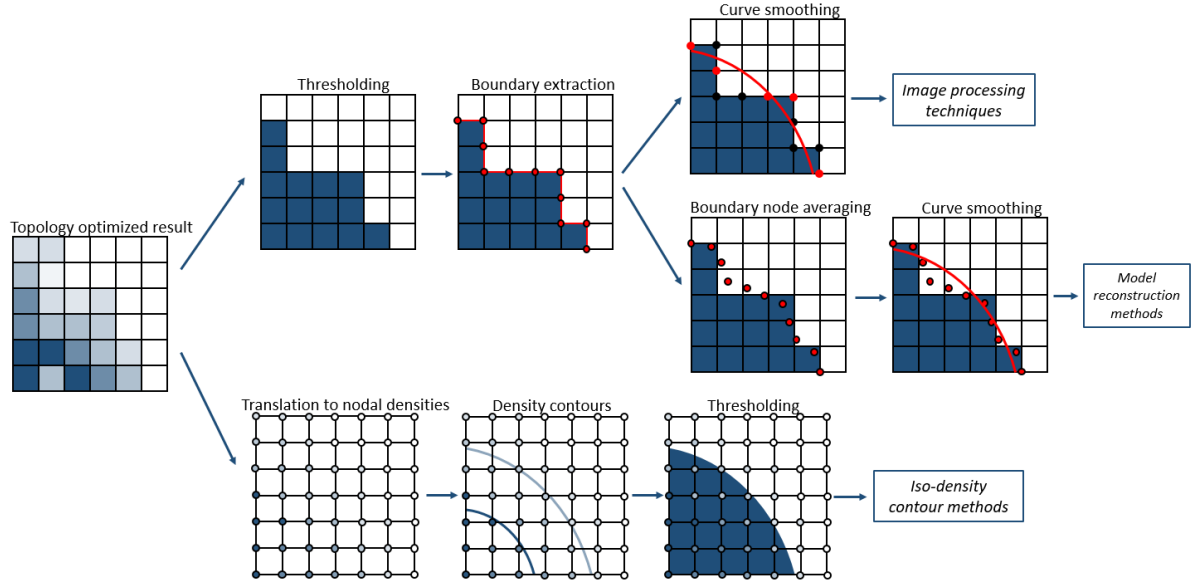


Figure 1.5: Overview of the different approaches available for stage two: geometry extraction. The upper branch are the 'image processing techniques' proposed by Papalambros and Chirehdast [25]. The middle branch are the 'model reconstruction methods' proposed by Tang and Chang [37]. The bottom branch are the 'iso-density contour methods' proposed by Maute and Ramm [23].

The threshold density value is chosen iteratively by assuming that the amount of material in the topology optimized result should be equal to the amount of material in the final binary image.

Marsan and Dutta [22] follow a similar image processing technique to obtain 3D structures. Firstly, a series of planar contours is extracted from the volume data using the three steps given above. Secondly, these contours are interpolated with parametric surfaces to obtain a 3D structure.

Lin and Chao [20] follow a similar approach but use B-splines to represent the external boundary. The inner boundaries are represented by standard geometric shapes. They claim this method can represent certain topology optimized results better and this results in a better manufacturability of the design.

Yi and Kim [44] use the active contour method to extract boundary segments from a binary image. The segments are simultaneously pulled towards features such as lines and edges. Similar to Lin and Chao [20], they used standard geometric shapes to construct their design.

Model reconstruction methods

Tang and Chang [37] introduced a geometric reconstruction technique with the aim to support topology optimization and Computed Aided Design (CAD) based shape optimization. Their approach uses partly the same steps as image processing techniques to extract a geometry from the topology optimized result. This results in the following four steps:

1. Thresholding converts the topology optimized result, with element densities ranging from 0 to 1, to a binary image.
2. Regions and boundaries are extracted.
3. Jagged boundary nodes are averaged.
4. Least-square fitting of the averaged nodes using B-spline curves.

The averaging of the boundary nodes is done by averaging the coordinates of the previous two, the current and the next two nodes. The boundary is constructed by least-square fitting the averaged boundary nodes, which makes it different from the image processing techniques. They also apply their method to a 3D problem by constructing the 3D geometry using a set of 2D sections.

Kumar and Saxena [17] use a similar process as averaging the boundary nodes. After identifying the boundary nodes, new points are projected on the middle of the boundary line segments. A new line is drawn through these mid-points after which the original boundary nodes are projected on this new line. This way the boundary nodes are slowly moved and smoothed. Although they use an unconventional method for stage

one, Material Mask Overlay Scheme (MMOS), the smoothing technique can be applied to any topology optimized result using a discretized design domain.

Iso-density contour methods

Maute and Ramm [23] use the contours of a constructed density field to define the shape of their topology optimized result. The three steps corresponding to this method are:

1. Elemental densities are translated to nodal densities.
2. Density contours are defined by spline functions based on the nodal densities.
3. Thresholding of the density contours defines the shape.

Hsu et al. [14] also use density contours to define the shape of the design. The density values of the elements are extrapolated in a non-linear fashion to nodal density values. Hsu and Hsu [13] extended the use of density contours to 3D problems by sweeping the sectional 2D contours to form 3D contours. More recently, a method based on density contours is proposed by [21] to obtain a truss geometry from a topology optimized result.

Extension to 3D applications

For all of the three methods shown above, it appears the methods are hard to extend to 3D problems. All of the proposed methods use the proposed 2D principle to create smooth section cuts of the 3D topology optimized result. Then these different layers were interpolated or 'swept' into a 3D object.

1.1.3. Stage 3 - Shape optimization

The design extracted at stage two usually loses a lot of its performance compared to the topology optimized result from stage one. A final shape optimization is done to obtain an optimized design again. Shape optimization is similar to stage one approaches because it also handles the material layout but typically cannot introduce new holes to the design. Moreover, the LSM introduced in section 1.1.1 can also be used for shape optimization.

Independent node movement

One of the first methods used in shape optimization is using the nodes of the structural analysis its mesh as design variables. As a follow up, Le et al. [18] proposed a method in which a length control is added to the node movement approach. In this way, a well-posed shape optimization can be reached and numerical stability can be ensured.

Isogeometric shape optimization

Nodal based shape optimization has some difficulties in the parametrization of the design domain (Cho and Ha [7]). New shape optimization methods have been introduced to overcome this problem. Wall et al. [41] presented an isogeometric shape optimization using non-uniform rational basis splines (NURBS) as basis functions. In this isogeometric approach, the design is embedded in the NURBS shape functions. So, one advantage is that the geometric flexibility of the NURBS basis allows for the exact representation of the geometry. Another big advantage is that the mesh refinement approaches are simplified which could improve the resolution of the results (Cho and Ha [7]).

Shape optimization using fictitious domains

The LSF was introduced in section 1.1.1 and in the previous paragraph the use of NURBS. These could be used as a smooth description of the boundaries of a structure. Researchers proposed methods using such a boundary description in combination with a fictitious domain method. This allows elements of the classic Finite Element Method (FEM) to be intersected by the boundary. Common methods are Fixed Grid (FG) by Garcia et al. [11], Finite Cell Method (FCM) by Parvizian et al. [26] and Generalized Finite Element Method (GFEM) by Fries and Belytschko [10].

1.2. Problem formulation

This MSc. Thesis focuses on the post-processing of topology optimized results obtained using the SIMP method, as denoted by the dotted red box in figure 1.1 surrounding stage two and three. Unwanted features as intermediate densities and jagged boundaries are exhibited by topology optimized results from stage one as shown in the example of figure 1.3. Although sections 1.1.2 and 1.1.3 already showed a few procedures for these stages to get rid of these features, the proposed methods are not satisfactory because of the following

reasons.

The extension to 3D applications of the proposed methods is not straightforward because mostly splines are used to describe the geometry. These spline representations can be obtained quite easily in 2D, creating spline surfaces out of 3D topology optimized results is a lot harder.

The design obtained at stage two usually loses a lot of performance compared to the topology optimized result obtained at stage one. This has mostly to do with the thresholding of the intermediate densities somewhere in the process. A good shape optimization in stage three could make a big difference. Although good shape optimization algorithms exist, these are mostly not related to the way the problem is described in stage one. A completely different mesh or analysis formulation is needed in those cases.

In short, there is not yet a structural design optimization process which conveniently couples the three stages of the process and finally results in an usable optimized design.

1.2.1. Goals

As already mentioned before, the goal of this MSc. Thesis is to come up with a method for post-processing topology optimized results. What problems the method should be able to overcome is defined above. The following list translates these problems into wishes for the post-processing method of topology optimized results.

Usable final geometry

- Final geometry should be crisp (i.e. no intermediate densities) and smooth (i.e. no jagged patterns).
- Final geometry should be convertible into a file format suitable for CAD, e.g. .stl-file.

Simple and all-round tool

- Only the model and result from the topology optimization should be used, no extra intervention should be needed to successfully use the post-processor.
- This tool should be of use for anyone using the 99-line MATLAB code for topology optimization written by Sigmund [31] (or an equivalent version in another programming language).
- It should be possible to easily post-process 3D results as well.

Fast

- Post-processing should be done in a reasonable amount of time. It should not take longer than the topology optimization in stage one.

1.2.2. Methodology

The proposed method in this thesis to post-process topology optimized results is summarized in this section. The motivation for certain choices in this methodology are given later on in this report. The proposed method aims to meet the goals set in section 1.2.1.

The boundary is described implicitly using an LSF. This means in stage two the element densities from stage one should be translated to an LSF. The LSF is a summation of Gaussian Radial Basis Functions (RBFs) which are located at every node of the mesh. A fictitious domain method is used to do the analysis of the geometry during the shape optimization. Typically the topology optimization uses a basic grid structure of elements which can be directly used as the fictitious domain method. The results from the structural analysis are then used by the sensitivity analysis to determine how to improve the shape of the geometry. An overview of the post-processing is given in figure 1.6.

1.3. Tools and implementation

The methodology developed for this MSc. Thesis is written in the open source programming language Python. The default Anaconda library is installed containing most of the commonly used Python packages. Additionally, the following packages are installed besides the Anaconda library: pyvtk, vtk and plotly. These packages are used for visualizing results. The first two packages are used to write .vtk-files. These files can be opened in Paraview, a convenient program for showing any kind of results. The third one is a package used for plotting/making images.

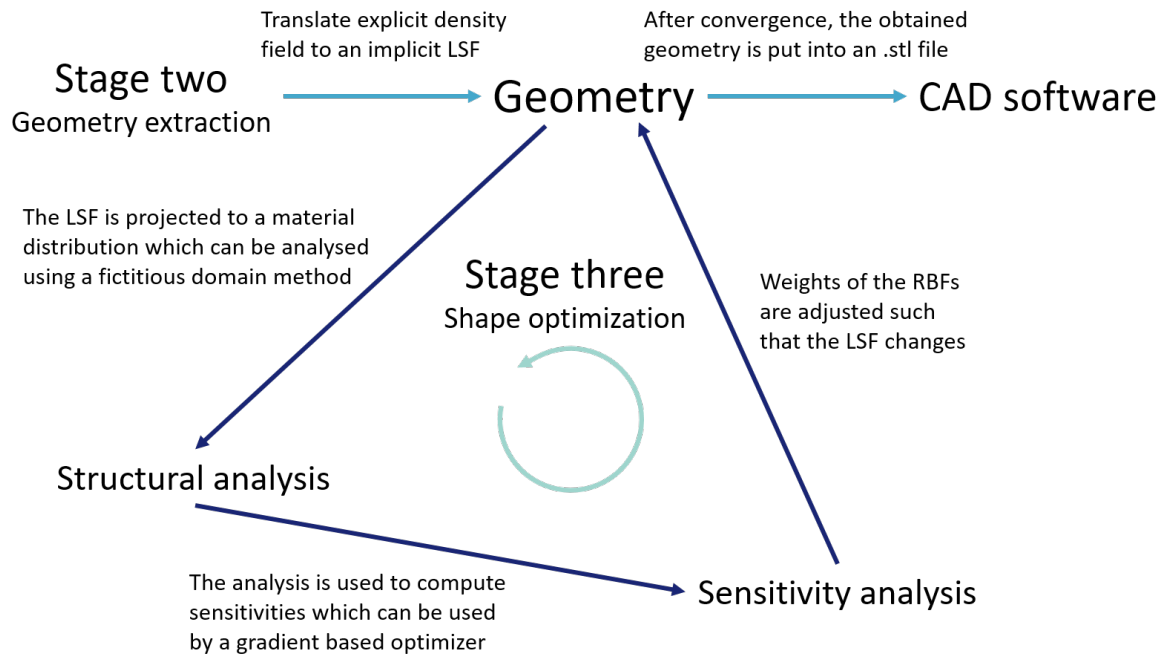


Figure 1.6: Overview of the structural design optimization using the proposed method for post-processing topology optimized results.

1.4. Outline report

This first chapter provided some background information and the scope of this project. The next chapter shows stage two of the structural design optimization using the proposed method. The shape optimization of stage three can be divided into the structural analysis in chapter 3, the sensitivity analysis in chapter 4, the LSF slope control in chapter 5 and the shape optimization in chapter 6. Several case studies are conducted in chapter 7 to show the results using the proposed post-processing method. The last chapter contains conclusions and recommendations. The outline of the report is shown schematically in figure 1.7. Also its relation with the structural design optimization process, already shown in figure 1.1, becomes apparent.

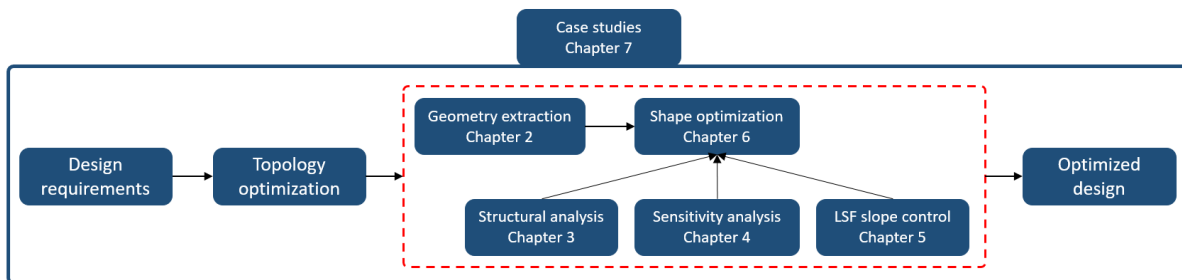


Figure 1.7: The outline of the report shown schematically in relation with the structural design optimization process shown in figure 1.1. The dotted red box again corresponds to the post-processing part of the entire process. The case studies in chapter 7 cover the entire structural design optimization.

2

Geometry extraction

The previous chapter provided an introduction to structural design optimization. This chapter describes stage two of that process. A geometry is extracted from the density field of the SIMP result. Algorithm 1 in appendix A shows the procedure for stage two, as explained in this chapter, in pseudo-code.

2.1. Geometry description

One of the unwanted features typically shown by SIMP results are jagged boundaries, whereas the optimized design should be smooth. The problem of the jagged boundaries lies in the difference between the discretization of the analysis using finite elements and the final smooth geometric description. Smooth boundaries can be represented using different methods. Interpolation or a least-square minimization are commonly used to find the best ‘fit’ of these methods to describe the initial geometry using the new format.

Spline functions

Spline functions are commonly used in CAD software. Extracting a 3D geometry from a topology optimized result is hard because several patches have to be constructed and swept to a 3D geometry, as already mentioned in section 1.1.2. Splines also show problems when large shape changes are needed in stage three, as can be seen in the work of Cervera and Trevelyan [6]. However, this is probably not a problem for the post-processing since the initial geometry is already relatively close to the final design.

Level Set Functions

LSFs are commonly used in topology optimization (see section 1.1.1) and are relatively easy to extend to 3D applications. The elemental densities obtained from the topology optimization provide initial values for the LSF. Creating an LSF from a density field (SIMP) would involve solving a set of equations such that the weights of the basis functions fit the density field. Solving a set of equations is a relatively easy procedure compared to fitting spline curves (for 2D applications) or surfaces (for 3D applications) to the density field. The weights of the basis functions can easily act as design variables in a subsequent shape optimization. For these reasons an LSF is chosen as the geometry description over the use of spline functions.

2.2. Level set basis functions

There are numerous options as basis functions for the LSF. The most trivial criterion is that the summation of the basis functions should result in a smooth function to make sure the geometry is as well. A summation of linear Lagrange polynomials shows discontinuous properties for example and is therefore not suited as an LSF basis. The most common functions used to construct an LSF are Radial Basis Functions (RBFs) according to van Dijk et al. [38]. The Gaussian RBF is shown in figure 2.1 and described by equation 2.1.

$$N_i = e^{-\left(\frac{r_i}{B}\right)^2}, \quad (2.1)$$

where

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}, \quad (2.2)$$

with x_i and y_i denoting the RBF its nodal location, x and y denoting the location where the RBF is evaluated and

$$B = \frac{L_x}{n_{el,x}} \quad (2.3)$$

is the bandwidth, a number used to scale the RBF, with L_x the size of the domain and $n_{el,x}$ the number of elements both in the horizontal direction.

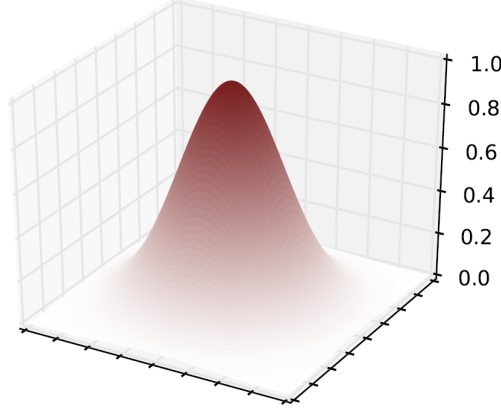


Figure 2.1: Gaussian Radial Basis Function

The RBFs are defined on every node of the fixed mesh which is also used for the SIMP formulation. The summation of these RBFs multiplied with different weights then gives the LSF ϕ .

$$\phi(x, y) = \sum_{i=1}^n N_i(x, y) s_i \quad (2.4)$$

where n is the number of nodes and s_i are the weights of the RBFs and hence the design variables. The LSF can be easily cast into a matrix format, just like regular finite element basis functions.

2.3. Nodal densities

The SIMP method results in a design having a certain density within each element. A linear system of equations could be set up such that at the center of each element, the summation of the RBFs is equal to the density of that element. This results in a system having a number of equations equal to the amount of elements. The number of variables is then equal to the amount of nodes because one RBF is defined on every node. This system of equations cannot be solved easily or exactly because the number of unknowns (i.e. the number of nodes) is larger than the number of equations (i.e. the number of elements).

The element densities can be translated to nodal densities, which means the number of equations becomes equal to the number of unknowns. The nodal densities are obtained by simply averaging the densities of elements which are connected to a node. An example of the nodal densities for the MBB beam case (see figure 1.3) is shown in figure 2.2.

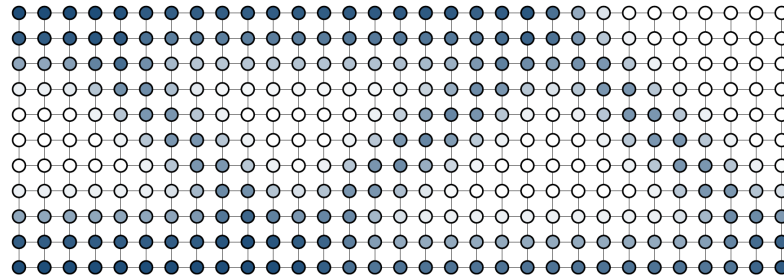


Figure 2.2: The nodal densities obtained after translating the element densities of the MMB beam case. From these densities a linear system of equations can be set up equal to the amount of nodes, which is $31 \times 11 = 341$ equations, to obtain an LSF.

2.4. Level Set Description

A linear system of equations can be set up to obtain the weights s_j of the level set basis functions such that the LSF corresponds to the nodal densities of the topology optimized result. A solution for the following matrix equation should be obtained:

$$\phi(x_j, y_j) = A_{ij} s_j = \rho_i \quad (2.5)$$

where $A_{ij} = N_i(x_j, y_j)$ and ρ_i is a vector containing the nodal densities. So, the system will have i equations, equal to the number of nodal densities and j unknowns, equal to the number of RBFs. As an example, the LSF for the MBB beam is shown in figure 2.3.

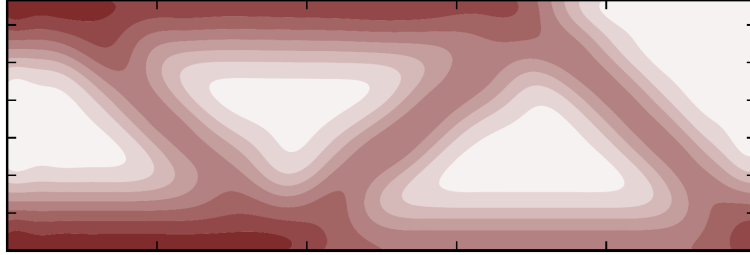


Figure 2.3: LSF of the MBB beam obtained after solving equation 2.5.

2.5. Projecting LSF to a material distribution

As already mentioned before, the LSF describes the geometry implicitly. So, the LSF does not directly represent material. The actual geometry (i.e. the density throughout the design domain) can be easily related to the LSF by using an approximate Heaviside function, see figure 2.4. The Heaviside function ($\tilde{H}$) can be approximated using equation 2.6.

$$\tilde{H}(\phi) = \frac{1}{1 + e^{-\kappa\phi}}, \quad (2.6)$$

where ϕ is the LSF and κ a parameter to control the steepness of the approximate Heaviside function. The density (i.e. the actual geometry) can be obtained using equation 2.6 as

$$\rho(\phi) = \rho_{\min} + (1 - \rho_{\min}) \cdot \tilde{H}(\phi), \quad (2.7)$$

where solid material is equivalent to $\rho = 1$ and void is $\rho = \rho_{\min} \approx 0$. Void always has a value and cannot be completely zero because this results in ill-conditioning of the stiffness matrix.

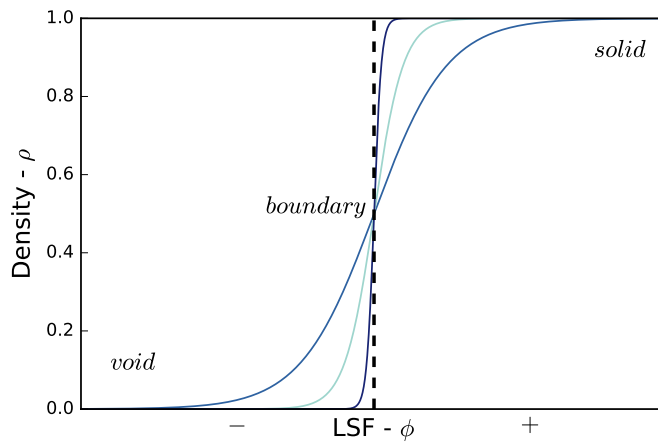


Figure 2.4: Heaviside projection on the LSF to obtain densities of the actual geometry. Negative LSF values correspond to a density of zero (void) and positive LSF values to a density of one (solid).

For more gradual slopes of $\tilde{H}$, intermediate densities occur around the boundary, resulting in a less crisp geometry. The approximate Heaviside function is plotted for three different values of κ in figure 2.4, resulting in different slopes. This step function is of use because it can simply relate all points in the domain associated to a negative LSF to a density of zero (i.e. void) and all points having a positive LSF to a density of one (i.e. solid).

However, the LSF obtained by solving equation 2.5 is fully positive because the nodal densities ρ_i are as well. So, the LSF should be shifted 'downward' as much such that the area within the zero-level contour satisfies the desired volume fraction. The area of an LSF contour can be computed for a trial threshold value between 0 and 1. If the area is larger than the volume fraction, the threshold is set higher. If the area is smaller than the volume fraction, the threshold is set lower. By computing the area in several cycles and adjusting the threshold accordingly, the correct threshold value can be obtained iteratively. The variables are then adjusted by not solving equation 2.5 for ρ_i but for the threshold value of the LSF ϕ_{th} obtained from the iterative procedure.

$$A_{ij} s_j = \phi_{th} \quad (2.8)$$

Finally, the variables become

$$s_j = s_{j,\rho_i} - s_{j,\phi_{th}}, \quad (2.9)$$

where s_{j,ρ_i} are the variables obtained after solving equation 2.5 for ρ_i and $s_{j,\phi_{th}}$ are the variables obtained after solving equation 2.5 for the threshold value.

The projection of the shifted LSF to a material distribution for the MBB beam is illustrated in figure 2.5.

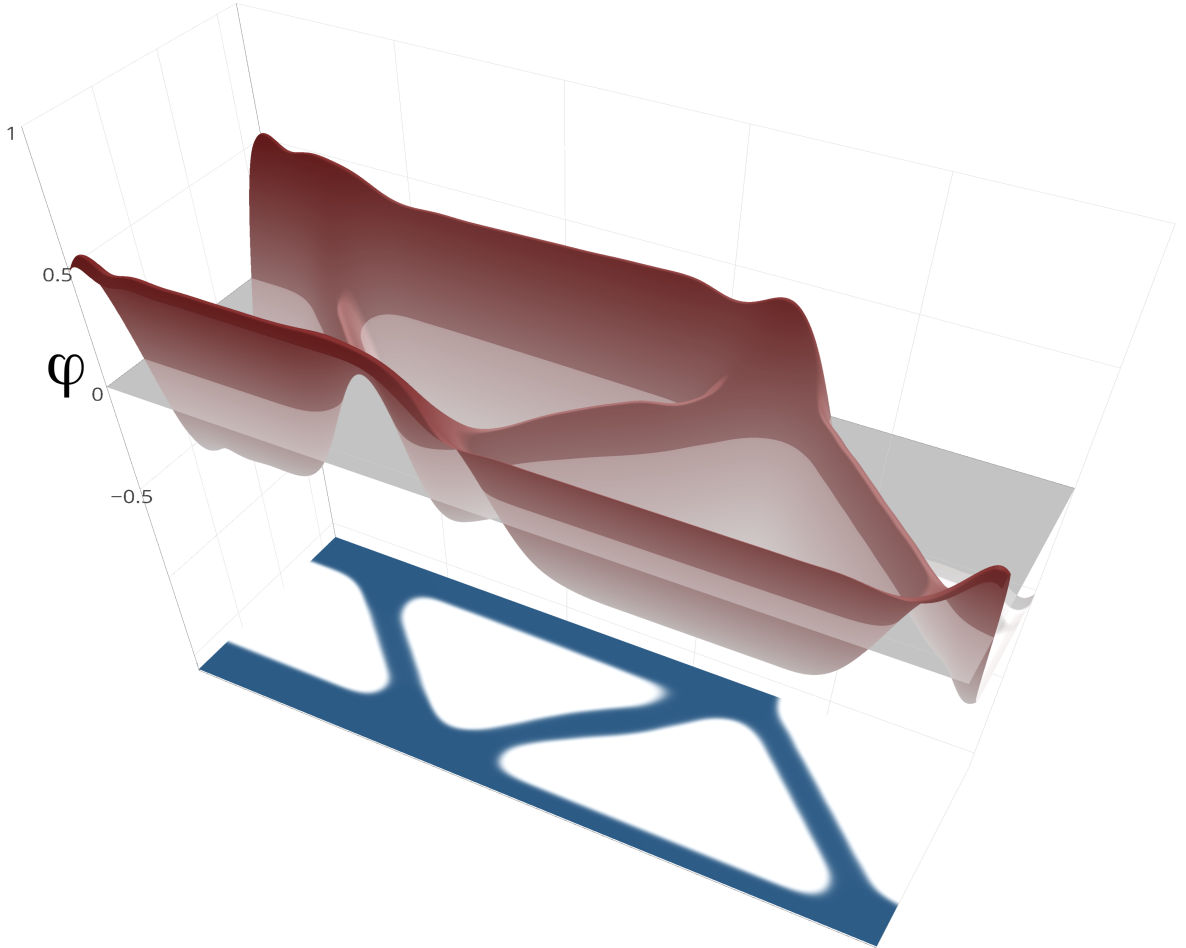


Figure 2.5: Illustration of projecting the implicit geometry description, the LSF (in red), to an explicit density description (in blue) using the approximate Heaviside function of equation 2.6.

Structural analysis

In the previous chapter a procedure for stage two was shown to obtain a geometry from the topology optimized result of stage one. Next is stage three of the structural design optimization, the shape optimization. The shape optimization is divided into four parts: the structural analysis, the sensitivity analysis, the LSF slope control and the optimization itself. This chapter covers the structural analysis, which is required to evaluate the performance of a design.

3.1. Finite Element Method

This section only gives the very principles of the Finite Element Method (FEM). Much literature can be found on FEM such as the extensive book of Szabo and Babuska [35] and the reader is referred to this book for a more detailed explanation if needed. The starting point of FEM is the strong form of the governing equation for continuum elasticity

$$\nabla \cdot \boldsymbol{\sigma} + \mathbf{b} = \mathbf{0}, \quad (3.1)$$

where $\boldsymbol{\sigma}$ is the stress tensor and $\mathbf{b}$ is a body force. The strong form can be multiplied with a weight function and can then be integrated by parts. This gives the weak form of the same elasticity problem. Finally, the Galerkin method introduces an approximate solution. The approximation consists of a summation of basis functions (e.g. Lagrange polynomials) multiplied with some unknown weights

$$u^h(x, y) = \sum_{i=1}^n N_i(x, y) \cdot u_i, \quad (3.2)$$

where u^h is the approximated solution, N_i are the basis functions and u_i are the unknown weights. The discretized equations that follow from the Galerkin method can then be numerically integrated over the domain using a Gauss quadrature scheme. Solving for the unknown weights u_i , gives the solution to the problem.

$$\mathbf{K}\mathbf{u} = \mathbf{f} \quad \rightarrow \quad \mathbf{u} = \mathbf{K}^{-1}\mathbf{f}, \quad (3.3)$$

where $\mathbf{K}$ is the stiffness matrix, $\mathbf{u}$ the unknown weights (or the displacement vector) and $\mathbf{f}$ the force vector.

3.2. Analyse geometry on a non-matching mesh

The structural analysis using FEM becomes less straightforward for a geometry which does not fit the elements of a finite element mesh. Two changes to the original FEM could make it possible to analyse such a problem.

3.2.1. Adaptive meshing

One option would be to make the elements fit the geometry perfectly. Some options are given which could make it possible to merge the analysis and the geometry into one model. The shape of the elements could be changed by moving the nodes such that they match the boundary of the geometry. This method is also called r-adaptivity (i.e. moving the nodes of the mesh).

Another option is to do an isogeometric analysis in which the basis of the elements is described using NURBS

or another appropriate spline function (Hughes et al. [15]). These NURBS can also describe the boundary between solid and void and therefore combines the analysis and geometry part into one model. Adaptive meshing is not considered to be very sound and robust. Changing elements can result too easily in distorted elements and moreover, has high computational costs. Furthermore, adaptive meshing only allows for relatively small shape changes in stage three to prevent mesh distortion.

3.2.2. Fictitious domain methods

On the other hand, one could try to make it possible for a boundary of a geometry to run through the finite elements of the analysis. The analysis and the geometry are then decoupled, resulting in a non-conforming mesh (i.e. a fictitious domain). Fictitious domain methods use a combination of enriching the elements (e.g. by increasing the polynomial order or adding functions to the existing basis) and a modified numerical integration.

Generalized Finite Element Method

A fictitious domain method commonly used in modelling cracks, is GFEM or sometimes also called the eXtended Finite Element Method (XFEM). GFEM can be seen as regular FEM (equation 3.2) which is enriched by for example a Heaviside function. This results in the approximation of the displacement field as

$$u^h(x) = \underbrace{\sum_i N_i(x) u_i}_{\text{regular FEM}} + \underbrace{\sum_j N_j(x) \cdot \psi(x) a_j}_{\text{enrichment}}, \quad (3.4)$$

where ψ can be the Heaviside function shown in figure 2.4 and a_j are the weights associated to the enrichment. Boundary elements, whose nodes are enriched by this Heaviside function, become ill-conditioned if the area on either side of the solid-void interface becomes very small (Fries and Belytschko [10]). During shape optimization the occurrence of these small area fractions within an element is inevitable. Several 'tricks' exist to circumvent this ill-conditioning. For example a node can be shifted away from a boundary, if this boundary gets close to the node. However, this artificial shift causes a jump in the sensitivities as described by Sharma et al. [30]. These problems make it hard to implement GFEM in a robust way for shape optimization using LSFs.

Finite Cell Method

Increasing the polynomial order of an element can be seen as a nonlinear enrichment to the linear FEM. This is known as the p-version of the Finite Element Method (pFEM) and has no problems with ill-conditioning like GFEM. The approximation for pFEM becomes

$$u^h(x) = \underbrace{\sum_i N_i(x) u_i}_{\text{regular FEM}} + \underbrace{\sum_j N_j(x) u_j}_{\text{nonlinear enrichment}}, \quad (3.5)$$

where i again denotes the linear FEM and j in this case is associated to the nonlinear enrichment. A disadvantage of pFEM is that it does not naturally incorporate the discontinuous solid-void interface. The difference in solid and void can be modelled using different numerical integration, see section 3.4. This method is also known as FCM and was introduced by Parvizian et al. [26]. FCM is considered to be a more appropriate fictitious domain method because it does not show the ill-conditioning problems GFEM does.

3.3. p-version of the Finite Element Method (pFEM)

An important part of FCM is the p-version of the Finite Element Method (pFEM). Hierarchical basis functions are used in pFEM, instead of standard Lagrangian basis functions, to be able to continuously increase the polynomial order of the elements without changing the mesh.

3.3.1. Hierarchical shape functions

The hierarchical shape functions can be divided into three groups for 2D and four groups for 3D. Only the 2D case is part of this explanation of pFEM. As for FEM, much literature can be found on pFEM as well, for example Szabó et al. [36].

1. **Nodal or vertex modes:** these modes are equivalent to the linear Lagrangian shape functions and are one on a node and zero on all other nodes. These are the functions N_i in equation 3.5.

$$N_{1,1} = \frac{1}{4}(1 + \xi_i \xi)(1 + \eta_i \eta) \quad (3.6)$$

2. **Edge modes:** these modes are defined on an edge of an element and are non-zero on an edge and zero on all other nodes and edges.

$$N_{i,1} = \frac{1}{2}(1 - \eta)\psi_i(\xi) \quad (3.7)$$

3. **Internal modes:** these modes are defined on the interior of an element and are non-zero in the interior and zero on all nodes and edges. Together with the edges modes, these are the functions N_j in equation 3.5.

$$N_{i,j} = \psi_i(\xi)\psi_j(\eta) \quad (3.8)$$

The function ψ can be obtained using the following formula

$$\psi_i(\xi) = \sqrt{\frac{2i-1}{2}} \int_{-1}^{\xi} L_{i-1} dx = \frac{1}{\sqrt{4i-2}}(L_i(\xi) - L_{i-2}(\xi)), \quad (3.9)$$

where L_i are Legendre polynomials. An example of every type of hierarchical shape function is shown in figure 3.1.

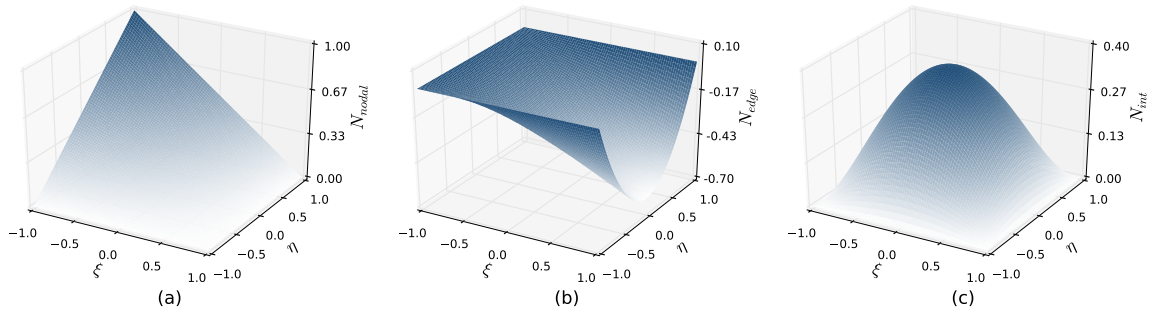


Figure 3.1: A nodal mode (a), an edge mode (b) and an internal mode (c) constructed using hierarchical basis functions.

3.3.2. Convergence pFEM

Szabó et al. [36] showed that pFEM has an exponential convergence for increasing polynomial order on problems with a smooth solution field. The geometry analysed for the shape optimization does not necessarily have a smooth solution field because there is a discrete solid-void interface. However, in section 2.5 the approximate Heaviside function was introduced to project the LSF to a geometry. A blurred boundary of intermediate densities is created around the boundary if the approximate Heaviside function is not so steep. This band improves convergence because of its smoother transition between solid and void.

3.4. Adaptive numerical integration

In the previous section the discretization was explained using higher-order elements. The governing discretized equations should also be numerically integrated. For regular FEM, the elements fit the geometry and the integration is straightforward with a Gauss quadrature scheme. For FCM, the elements do not fit the geometry because these can be intersected by the boundary. Somehow, an adaptive numerical integration method should capture the material interface within an intersected element.

An overview of different adaptive numerical integration techniques is given in this section, see figure 3.2. Generally, these methods add integration points and scale the stiffness in these points based on the amount of material. The way these methods work precisely, is discussed next.

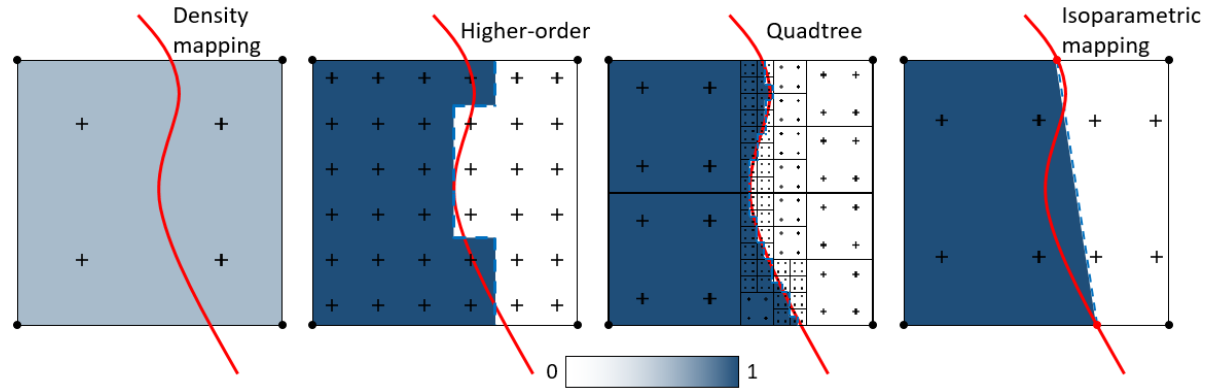


Figure 3.2: Different adaptive numerical integration techniques illustrated for fictitious domain methods. Density mapping is scaling the density with the volume fraction of the element. Higher-order integration uses more Gauss integration points than needed to integrate a certain polynomial order exactly. The quadtree refinement splits an intersected element into four integration cells, these cells are also divided in four if intersected, etc. The isoparametric mapping can be used if the precise boundary location is known.

Density mapping

This method, first used by Garcia et al. [11], is also classified as an 'Ersatz material' method. An Ersatz material method interpolates physical properties of an element to the density of the material in that element, similar to the SIMP method. The density of an intersected boundary element is equal to the area fraction of solid material. For example if a boundary element is solid for 30%, then the stiffness of the integration points scales with 0.3. A clear disadvantage is that not all information or details on the material layout is taken into account for the analysis.

Higher-order Gauss quadrature

This numerical integration procedure uses higher-order Gauss quadrature schemes. The extra integration points can be used to reflect the material distribution in an element more closely. The solid part of the domain has normal weights and the void has weights very close to zero. The weights cannot become zero because this results in bad conditioning of the stiffness matrix (Parvizian et al. [26]).

Quadtree refinement

The quadtree refinement splits an element into smaller integration cells if the element is intersected by the boundary (e.g. Parvizian et al. [27]). An exact Gauss scheme can be applied to each integration cell. Not only will the amount of integration points increase in an element, but the points are more dense close to the boundary, reflecting the material distribution in the element even better.

Isoparametric mapping

This method can only be used when (parts of) the boundary is known explicitly. Since the geometry is described by an implicit LSF, these integration methods cannot be used easily. The curved boundary can be linearised, like in figure 3.2 and also done before by Li et al. [19], and mapped to a reference element which is a commonly known method in regular FEM.

Other methods

Many more methods have been proposed to achieve a good numerical integration of the boundary elements besides the four methods shown in figure 3.2. The other methods need to know some explicit details of the boundary, give rise to sensitivity issues or the efficiency is not clear.

- Integrate triangles after triangulation of the solid part of the boundary element (e.g. Wei et al. [43])
- Equivalent polynomial function using contour integrals (e.g. Ventura [40])
- Moment fitting of implicit LSF (e.g. Müller et al. [24] and Joulaian et al. [16])

Conclusion

As shown, multiple methods exist to perform the numerical integration for methods using a fictitious domain. Several integration methods, like the ones shown in figure 3.2, were studied by Abedian et al. [1]. Their paper indicates that local quadtree refinement is a relatively economic approach to obtain accurate results. The last

benefit of quadtree refinement is the easy extension to 3D applications, where an octree refinement can be used in a similar fashion.

3.5. Example analyses

Three examples are analysed to show the performance of FCM (i.e. pFEM with quadtree integration). Convergence plots are shown as well. The convergence of FCM relies on the discretization error (pFEM) and the integration error (adaptive numerical integration using quadtree refinement). Parvizian et al. [26] already showed exponential convergence for FCM but only until the integration error became dominant.

3.5.1. Axially loaded bar with thinner mid-section

The first example is an axially loaded bar with a thinner mid-section, see figure 3.3. The length of the bar is 5

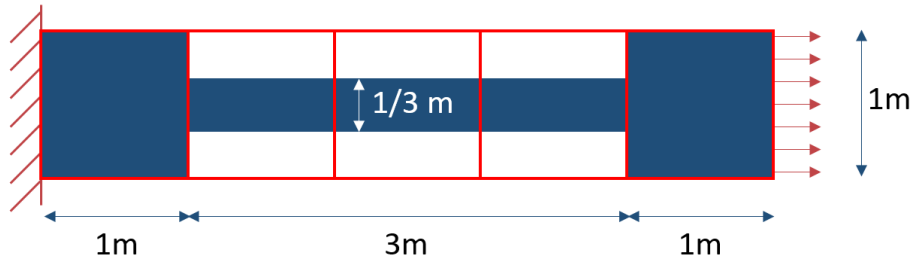


Figure 3.3: Axially loaded bar example with a thinner mid-section of 1/3 m instead of 1 m. A distributed load of 1 N/m is applied to the right edge while the left edge is completely fixed. The bar is discretized using five elements of 1 x 1 m, these are shown in red.

m and the height and thickness are 1 m, except for the mid-section where the height is 1/3 m. A distributed load of 1 N/m is applied on the right edge. The analytic solution for the displacement at the right end of the bar is:

$$u_{ex} = F_q \cdot L_y \cdot \left(\frac{2}{5} \cdot \frac{L_x}{E \cdot L_y \cdot t} + \frac{3}{5} \cdot \frac{L_x}{E \cdot (L_y/3) \cdot t} \right) = 11 \text{ m} . \quad (3.10)$$

Linear elements should be able to analyse this problem exact, if the mesh is matching the geometry. For this example a non-matching mesh is used. The extended bar is discretized using 5 equally large elements of 1 x 1 m. Different levels of quadtree refinement are tested and shown in figure 3.4. A Gauss integration scheme of 4x4 points is used for the thinner mid-section, which is higher than needed for linear elements. Different amounts of quadtree refinement are tested to examine its convergence. This particular integration procedure is chosen for this example to illustrate the 'randomness' in the convergence of the chosen adaptive numerical integration method. As can be seen in figure 3.4, the following effects occur for different quadtree refinements:

- **no quadtree refinement** - no integration points are located where material is, which means no material is modelled and the displacement becomes extremely large.
- **one quadtree refinement** - half of the integration points are located where material is, which means the mid-section is modelled as if it was 1/2 m in height instead of 1/3 m. The extended bar is modelled too stiff, $u_{FE} = 8 \text{ m}$.
- **two quadtree refinements** - a quarter of the integration points are located where material is, which means the mid-section is modelled as if it was 1/4 m in height. The extended bar responds too weak, $u_{FE} = 14 \text{ m}$.
- **more than two quadtree refinements** - eventually the mid-section is modelled as its true height of 1/3 m. However, the convergence is not a smooth descending curve but rather a random process in which not always the correct amount of integration points are captured.

From this example can be concluded that the integration error does not necessarily follow a stable or smooth convergence path to the exact answer for an increase in quadtree refinements.

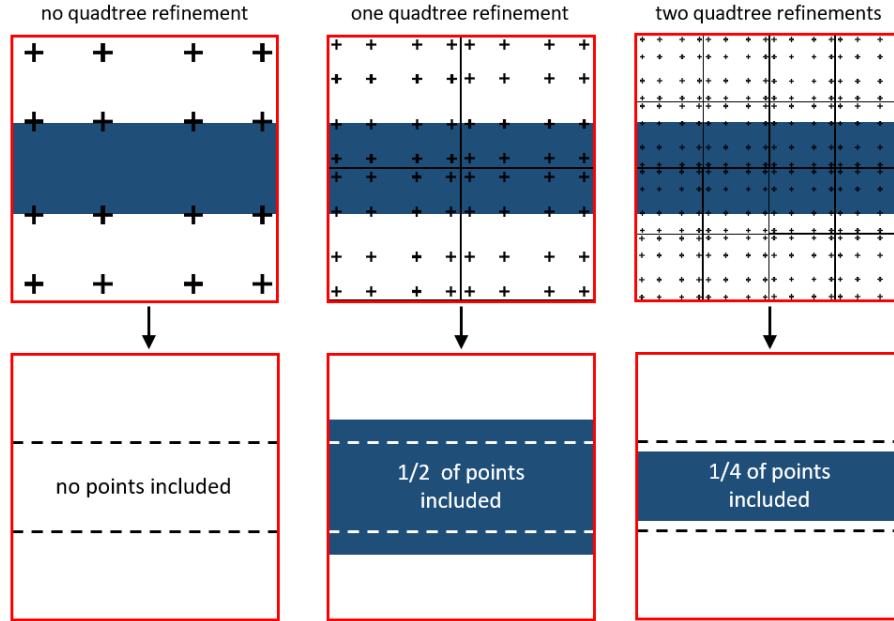


Figure 3.4: Different quadtree refinements on elements in the thinner mid-section, see figure 3.3. Each integration cell has a 4x4 Gauss integration scheme. Elements are shown in red, integration cells in black and the integration points are the black dots. The top row shows the analysis grid on the thinner mid-section and the bottom row shows the actual amount of material modelled. There is a big difference in the amount of material modelled for different amount of quadtree refinements.

3.5.2. Rectangular plate with circular hole

This example is already analysed by Parvizian et al. [26] and the convergence is exponential until the integration error becomes dominant. The same example is reproduced to show the same convergence. Furthermore, the material distribution of the plate is described implicitly by an LSE, as will be done for the geometry in the design optimization. The problem and its discretization are shown in figure 3.5.

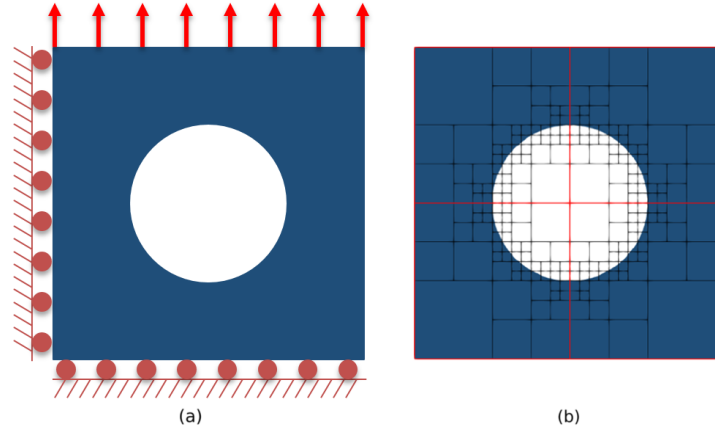


Figure 3.5: A plate of 4 mm x 4 mm with a circular hole in the middle. The diameter of the hole is 2 mm and the applied distributed load at the top is 100 MPa. The left edge is fixed in the horizontal direction and the bottom edge is fixed in the vertical direction (a). An example of the discretization of the plate using 2x2 elements in red and four quadtree refinements in black (b).

The material properties of the plate with circular hole are:

- Young's Modulus $E = 206900$ MPa;
- Poisson's ratio $\nu = 0.29$;
- Thickness $t = 1$ mm;
- Minimum density $\rho_{\min} = 1 \cdot 10^{-14}$ kg/mm³.

The relative error of the analysis is computed in the energy norm. The exact answer for the strain energy is given by Parvizian et al. [26] to be $U_{ex} = 0.7021812127$. The strain energy of the finite element approximation can be computed as follows:

$$U_{FE} = \frac{1}{2} \mathbf{u}^T \mathbf{K} \mathbf{u}, \quad (3.11)$$

where $\mathbf{u}$ is the displacement vector and $\mathbf{K}$ the stiffness matrix. The relative error in the energy norm is defined as:

$$e_r = \sqrt{\frac{|U_{ex} - U_{FE}|}{U_{ex}}}. \quad (3.12)$$

The convergence of this relative error in the energy norm is plotted for an increase in degrees of freedom (DOF) (figure 3.6a) and for an increase in quadtree refinements (figure 3.6b).

The increase in DOF can be achieved by refining the mesh (i.e. adding more elements) for a fixed polynomial order of one, this is also called h-version of the Finite Element Method (hFEM). The line 'hFEM-G2-QT0' does not have quadtree refinement whereas the line 'hFEM-G2-QT5' uses five quadtree refinements. The convergence of hFEM should be algebraic with a slope of 0.5. The line 'hFEM-G2-QT5' shows this convergence but the other line 'hFEM-G2-QT0' deteriorates eventually. The only difference between these two lines is the number of quadtree refinements (i.e. the accuracy of the integration). So, the difference must be due to an integration error.

The increase in DOF can also be achieved by increasing the polynomial order, which is the previously discussed pFEM. The convergence for pFEM should be exponential. Clearly, this is the case as can be seen for the line 'pFEM-G15-QT5' in figure 3.6a. Again, as for hFEM, an integration error appears for the line 'pFEM-G15-QT3' because this one deteriorates from line 'pFEM-G15-QT5'. The only difference between these two lines is again the number of quadtree refinements (i.e. the accuracy of the integration).

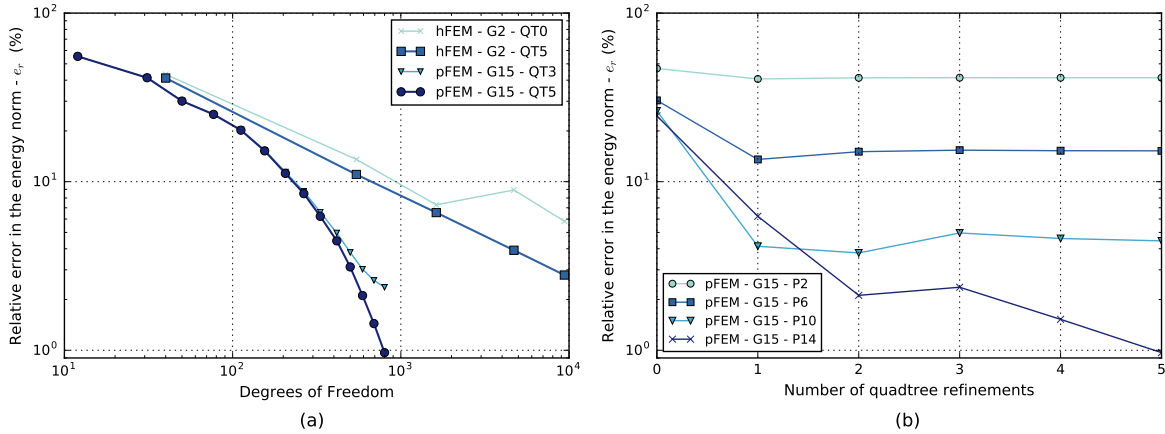


Figure 3.6: The convergence of FCM shown for the plate with circular hole example, see figure 3.5. The legend shows the type of line as: G is the number of Gauss points, QT is the number of quadtree refinement and P is the polynomial order. The relative error in the energy norm is plotted versus the degrees of freedom for either hFEM or pFEM and for different quadtree refinements on a logarithmic scale (a). The relative error in the energy norm is plotted versus the number of quadtree refinements for pFEM and different polynomial orders on a logarithmic scale (b).

The most dominant error is obviously of most interest, either the discretization or the integration error. The pFEM result for three quadtree refinements was already deteriorating for large amount of DOF as shown in figure 3.6a due to the integration error. Looking at figure 3.6b, the error for line 'pFEM-G15-P2' is basically a flat line, meaning the error does not get less for more accurate integration. This can only mean that the discretization error is dominant and the polynomial order should be increased to improve the result.

3.5.3. Geometry obtained at stage two for the MBB beam case

The previous examples show that the structural analysis converges as expected. Next, the analysis is applied to the initial geometry of the MBB beam case shown in figure 2.5. The polynomial order is increased while the numerical integration is performed using an 11x11 Gauss integration scheme and four quadtree refinements. Although the exact answer is unknown, it is clear the strain energy converges, rather slowly, to a stable

value, see figure 3.7a. Figure 3.7b shows the discretization error is dominant for this case because there is no reduction in the error for higher accuracy of the integration.

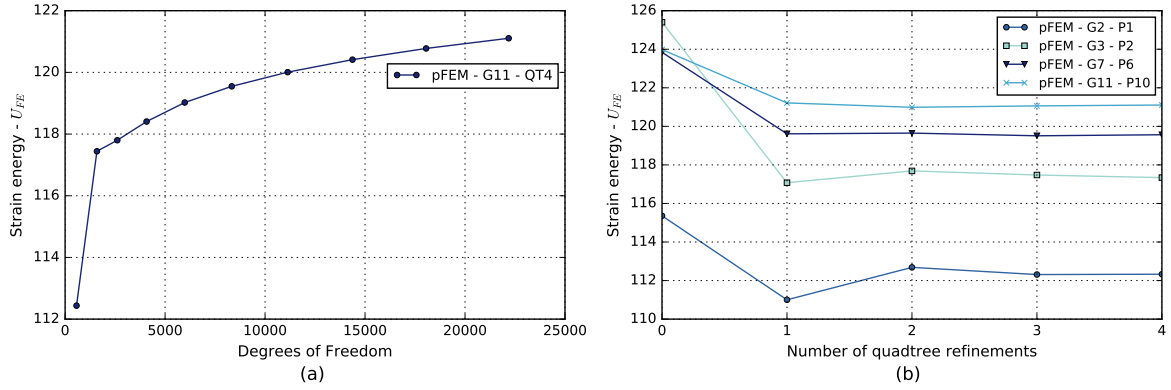


Figure 3.7: The convergence of FCM shown for the MMB beam at stage two, obtained from an LSF with an exact Heaviside function, see figure 2.5. The legend shows the type of line as: G is the number of Gauss points, QT is the number of quadtree refinement and P is the polynomial order. The strain energy is plotted versus the degrees of freedom for either hFEM or pFEM and for different quadtree refinements (a). The strain energy is plotted versus the number of quadtree refinements for pFEM and different polynomial orders (b).

The analysis of figure 3.7 uses an exact Heaviside function projection, i.e. a discrete interface, to obtain a geometry from an LSF. However, the LSF is mapped using an approximate Heaviside function (see section 2.5), which creates a blurred boundary of intermediate densities. The results using an approximate Heaviside function projection are shown in figure 3.8. The relative difference in the energy norm with the exact projection is about 1.2%, resulting in almost the same graph as shown in figure 3.7a. Figure 3.8b shows that improving the integration accuracy by increasing the number of quadtree refinements, contributes less to an improvement of the result compared to figure 3.7b. Apparently, a smoother transition from solid to void requires not so many quadtree refinements.

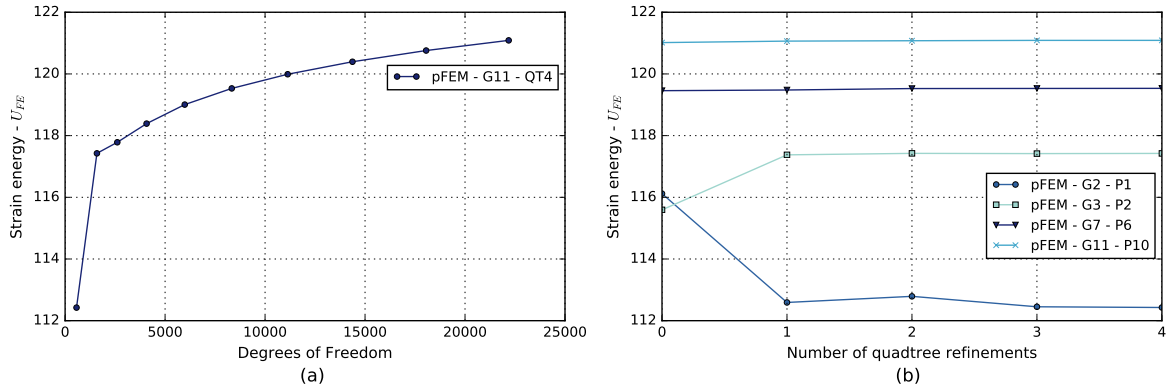


Figure 3.8: The convergence of FCM shown for the MMB beam case at stage two, obtained from an LSF with an approximate Heaviside function. The legend shows the type of line as: G is number of Gauss points, QT is number of quadtree refinement and P is the polynomial order. The strain energy is plotted versus the degrees of freedom for either hFEM or pFEM and for different quadtree refinements (a). The strain energy is plotted versus the number of quadtree refinements for pFEM and different polynomial orders (b).

4

Sensitivity analysis

The structural analysis was shown in the previous chapter. Another part of stage three is the sensitivity analysis, which is presented in this chapter. The sensitivities give the gradient-based optimizer a direction in which the boundary should move to improve the design. These sensitivities are basically first-order derivatives of the objective and the constraint with respect to the design variables. Some common methods for computing sensitivities are given first before deriving the sensitivities for the shape optimization of the post-processor.

4.1. Review of sensitivity analysis methods

Different ways exist to compute the sensitivities of the design. An overview is provided below, and more details can be found in the work of van Keulen et al. [39]. The example sensitivities given below are for a response g with respect to a design variable x .

1. Global finite differences

- Finite differences are computationally very expensive for obtaining sensitivities because an analysis of the structure has to be done for every variable x .

$$\frac{\partial g}{\partial x} \approx \frac{g(x + \Delta x) - g(x)}{\Delta x} \quad (4.1)$$

2. Discrete derivatives

- The discrete method can be used if the objective can be linked via another quantity to the variables $\mathbf{x}$. The sensitivities can then be obtained easily using the chain rule.

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + \frac{\partial g^T}{\partial \mathbf{u}} \frac{d\mathbf{u}}{dx} \quad (4.2)$$

Direct method

- The direct approach calculates the sensitivity of the state variable vector by solving a linear system for every variable x .

$$\frac{d\mathbf{u}}{dx} = \mathbf{K}^{-1} \left(\frac{\partial \mathbf{f}}{\partial x} - \frac{\partial \mathbf{K}}{\partial x} \mathbf{u} \right) \quad (4.3)$$

Another option could be to use Finite Differences to compute this term in case it is easier to implement.

$$\frac{d\mathbf{u}}{dx} \approx \mathbf{K}^{-1} \left(\frac{\mathbf{f}(x + \Delta x) - \mathbf{f}(x)}{\Delta x} - \frac{\mathbf{K}(x + \Delta x) - \mathbf{K}(x)}{\Delta x} \mathbf{u} \right) \quad (4.4)$$

Adjoint method

- The adjoint method is an improvement of the direct method if the number of variables is much higher than the number of responses. For this method one needs to solve a linear system for every response, rather than calculating $d\mathbf{u}/dx$ for every variable. First an augmented response g^* is created

$$g^* = g + \lambda (\mathbf{f} - \mathbf{K}\mathbf{u}) , \quad (4.5)$$

where λ is a Lagrange multiplier and note that the last term is always zero. The derivative of this augmented response is

$$\frac{dg^*}{dx} = \frac{dg}{dx} = \frac{\partial g}{\partial x} + \lambda \left(\frac{\partial \mathbf{f}}{\partial x} - \frac{\partial \mathbf{K}}{\partial x} \mathbf{u} \right) + \left(\frac{\partial g}{\partial \mathbf{u}} - \lambda \mathbf{K} \right) \frac{d\mathbf{u}}{dx} . \quad (4.6)$$

λ in equation 4.6 can be chosen such that the last term vanishes, making it needless to compute the sensitivity of the state variable vector. The sensitivity of a response g with respect to the variable x becomes

$$\frac{dg}{dx} = \frac{\partial g}{\partial x} + \frac{\partial g}{\partial \mathbf{u}} \mathbf{K}^{-1} \left(\frac{\partial \mathbf{f}}{\partial x} - \frac{\partial \mathbf{K}}{\partial x} \mathbf{u} \right) . \quad (4.7)$$

3. Continuum derivatives

- Continuum sensitivities are derived by differentiating the governing continuum equations directly. So, these approaches first differentiate and then discretize, where the previous methods first discretized and then differentiated. Also this class comes in a discrete and an adjoint method. Continuum derivatives can be considered more involved than the previous approaches.

4.2. Sensitivities for shape optimization

Gradient-based optimizers use the derivatives of the objective and constraint with respect to the variables to make changes to the design. The optimization problem considered in this report is minimizing the compliance (i.e. maximizing the stiffness) subjected to a volume fraction constraint.

$$\begin{aligned} & \underset{s_i}{\text{minimize}} && C(s_i) \\ & \text{subject to} && g(s_i) \leq 0. \end{aligned}$$

The objective C is

$$C = \mathbf{u}^T \mathbf{K} \mathbf{u} = \mathbf{f}^T \mathbf{u} . \quad (4.8)$$

The constraint g is

$$g = \frac{V}{L_x \cdot L_y \cdot V_{\text{frac}}} - 1 , \quad (4.9)$$

where V is the volume, L_x the size of the design domain in the horizontal direction, L_y the size of the domain in the vertical direction and V_{frac} the desired volume fraction. The volume fraction is the percentage of the design domain that can be used as material.

The derivatives or sensitivities of the compliance and the volume fraction constraint with respect the variables s_i is given below. Furthermore, a check on these sensitivities is done using a global Finite Differences procedure.

4.2.1. Sensitivity analysis objective - compliance

The derivative of the minimum compliance objective with respect to the variables (weights s_i) is desired. The chain rule of differentiation gives:

$$\frac{\partial C}{\partial s_i} = \left(\frac{\partial C}{\partial \rho_j} \frac{\partial \rho_j}{\partial \phi} \right) \cdot \frac{\partial \phi}{\partial s_i} . \quad (4.10)$$

The weights s_i influence the LSF ϕ , the LSF influences the densities ρ_j in the integration points j and the densities then influence the objective C . The partial derivative of ϕ with respect to the variables s_i is

$$\frac{\partial \phi}{\partial s_i} = \sum_{i=1}^n e^{-\left(\frac{r_i}{B}\right)^2} . \quad (4.11)$$

The density as a function of the LSF was already introduced in section 2.5 and is

$$\rho_j(\phi) = \rho_{\min} + (1 - \rho_{\min}) \cdot \left(\frac{1}{1 + e^{-\kappa \phi}} \right) . \quad (4.12)$$

The derivative of the densities ρ_j with respect to the LSF ϕ can now be computed.

$$\frac{\partial \rho_j}{\partial \phi} = (1 - \rho_{\min}) \cdot \frac{\kappa \cdot e^{\kappa \phi}}{(1 + e^{\kappa \phi})^2} \quad (4.13)$$

Finally, the partial derivative of the compliance with respect to the density in an integration point j should be determined. This is the same sensitivity as computed for SIMP problems (Bendsøe and Sigmund [5]), for which the adjoint method discussed earlier is used.

$$\frac{\partial C}{\partial \rho_j} = -p \mathbf{u}^T \rho_j^{p-1} \mathbf{K} \mathbf{u}, \quad (4.14)$$

where p is the penalization parameter of the density.

4.2.2. Sensitivity analysis constraint - volume fraction

The sensitivity of the constraint on the volume fraction, uses partly the same derivatives as the sensitivity of the compliance. The derivative of the constraint (see equation 4.9) with respect to the design variables is

$$\frac{\partial g}{\partial s_i} = \frac{\partial g}{\partial V} \cdot \left(\frac{\partial V^T}{\partial \rho_j} \frac{\partial \rho_j}{\partial \phi} \right) \cdot \frac{\partial \phi}{\partial s_i}, \quad (4.15)$$

where equation 4.9 can be easily differentiated to obtain the first term

$$\frac{\partial g}{\partial V} = \frac{1}{L_x \cdot L_y \cdot V_{\text{frac}}}. \quad (4.16)$$

The derivative of the volume with respect to the densities in the integration points (second term in equation 4.15) is equal to the weights of the Gauss integration points multiplied with the determinant of the Jacobian of the particular integration cell. The last two terms in equation 4.15 were already shown in section 4.2.1 in respectively equation 4.13 and 4.11.

4.2.3. Finite Difference check

Although the use of Finite Differences (FD) for sensitivity computation is not very efficient when a large amount of variables is used, it is very convenient for checking the sensitivities computed using the procedure shown in the previous sections. One variable is chosen and perturbed by an increment of Δs .

$$\frac{\partial C}{\partial s_i} \approx \frac{C(s_i + \Delta s) - C(s_i)}{\Delta s} \quad (4.17)$$

The approximated derivative of the compliance and the one obtained using the chain rule of equation 4.10 are shown in figure 4.1. Clearly, there is a range of step sizes Δs for which both derivatives are almost the same. From this can be concluded that the sensitivities are computed correctly. For very large steps the FD result differs but this is because of the truncation error. For very small step sizes the FD result differs because of the condition error.

4.3. Link with SIMP sensitivities

The sensitivity formulations shown in this chapter are related to minimum compliance problems while satisfying a constraint on the volume fraction. Potentially, the type of formulation used in this chapter for the sensitivities could handle all kind of problems which could also be optimized using the SIMP formulation. The shape optimization uses a combination of derivatives which are the same for every type of problem or can be directly used from the SIMP formulation. Density based methods use the derivative of the objective with respect to the density. The shape optimization of stage three also uses this partial derivative, except the density is defined in an integration point instead of the entire element. The other partial derivatives are the LSF with respect to the variables and the density with respect to the LSF, see equation 4.18.

$$\frac{\partial C}{\partial s_i} = \underbrace{\left(\frac{\partial C^T}{\partial \rho_j} \frac{\partial \rho_j}{\partial \phi} \right)}_{\text{SIMP}} \cdot \overbrace{\frac{\partial \phi}{\partial s_i}}^{\text{problem independent}} \quad (4.18)$$

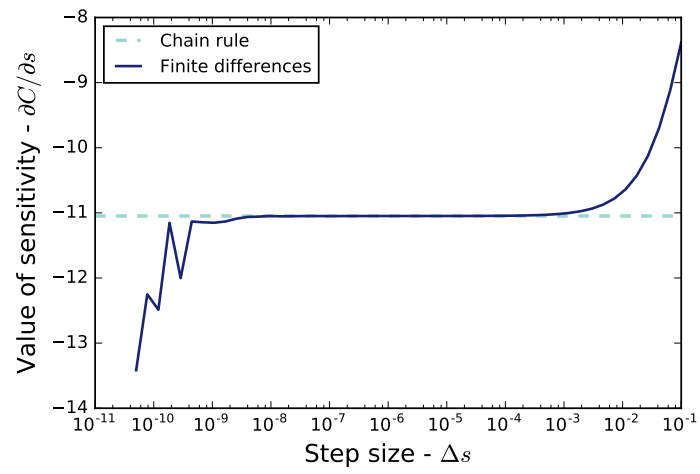


Figure 4.1: The density (black) and the derivative of the density with respect to the LSF (red) for $\kappa = 8$

5

LSF slope control

The previous two chapters discussed the structural analysis and the sensitivity analysis. Next it is advantageous to control the slope of the LSF for the shape optimization for a reason explained next. The density and its derivative with respect to ϕ are plotted in figure 5.1. The formulas were already explained in respectively section 2.5 and section 4.2.1, the density

$$\rho(\phi) = \rho_{\min} + (1 - \rho_{\min}) \cdot \left(\frac{1}{1 + e^{-\kappa\phi}} \right), \quad (5.1)$$

and its derivative with respect to the LSF

$$\frac{\partial \rho}{\partial \phi} = (1 - \rho_{\min}) \cdot \frac{\kappa \cdot e^{\kappa\phi}}{(1 + e^{\kappa\phi})^2}. \quad (5.2)$$

The density function becomes steeper for higher values of κ and the derivative then starts to look like a peak. So, the derivative of the density with respect to the LSF ends up being zero in almost the entire domain for too high values of κ , resulting in no sensitivity information for the optimization. The transition of void to solid should roughly cover a distance equal to the distance between two integration points. This way it is guaranteed to capture sensitivity values around the boundary. The value of κ should also not be too low because the density function will flatten, creating a blurred boundary with unwanted intermediate densities. Besides κ , the density is also influenced by the LSF ϕ . So, if the LSF can be controlled, then κ can be computed carefully based on the controlled LSF. This results in a density function which is not too steep and not too flat. This chapter discusses how to achieve a proper LSF slope control.

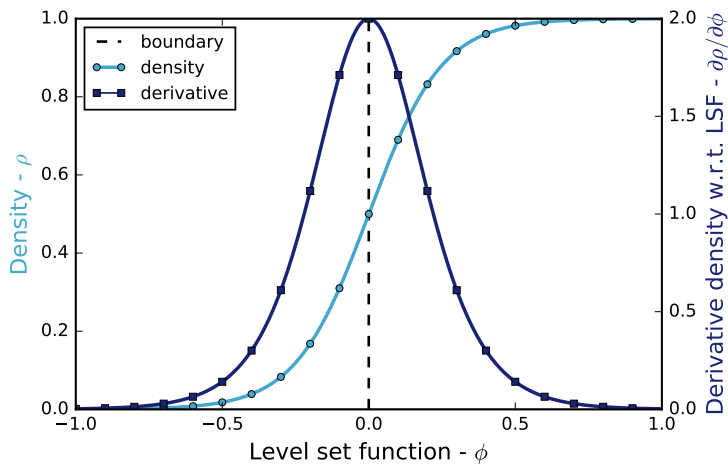


Figure 5.1: The density and its derivative with respect to the LSF plotted versus the LSF ϕ for $\kappa = 8$. As the density curve gets steeper and approaches the dotted line of the boundary, the derivative becomes zero everywhere except at $\phi = 0$, where it is infinite.

5.1. Possible slope control methods

Several procedures could be followed to control the slope of the LSF.

- The LSF can be mapped to a 'hat-function' which has a slope of 1. In literature this is known as reinitialization and for example done by Allaire et al. [2]. Reinitialization is usually done every few optimization iterations. This method is not very consistent because the slope of the LSF is not constantly controlled during the optimization.
- Another option is to impose a signed distance function, shown in equation 5.3, directly as a nonlinear constraint.

$$\|\nabla\phi\| = \sqrt{\left(\frac{\partial\phi}{\partial x}\right)^2 + \left(\frac{\partial\phi}{\partial y}\right)^2} = 1, \quad (5.3)$$

where ϕ is again the LSF and x and y are coordinates. This equation basically sets a fixed equation between the coordinates and the LSF. However, many constraints should be imposed to make sure this constraint is satisfied everywhere.

- The product of $\kappa \cdot \phi$ in equation 5.1 should always be a constant value. Another option would be to make κ a variable as well which adjusts to ϕ such that the function for the density retains its desired shape. The same problem arises as for the previous option, many constraints have to be imposed because the constraint should be satisfied everywhere.
- The penalization of intermediate density values, as used in the SIMP method (see figure 1.2), can also be used for the LSM. The design contains more intermediate densities when the optimizer chooses to flatten the LSF. Penalizing intermediate densities could solve this part of the problem. On the other hand, it will push the optimizer to make the LSF very steep. A very steep LSF could be controlled by bounding the design variables s_i .

The last option, a combination of penalization of intermediate densities and bounding the design variables, is preferred because of its consistency and efficiency.

5.2. Maximum slope RBF summation

The maximum slope of the LSF can be controlled by bounding the design variables s_i . A series of 1D RBFs and their summation are plotted in figure 5.2 for a bandwidth $B = 1$. The first four RBFs have a weight of minus one (i.e. $s = -1$) and the other four have a weight of one (i.e. $s = 1$). The summation reaches both the maximum and minimum possible value of ± 1.7726 . This results in the steepest descent possible for the summation, and thus the maximum absolute value of the slope.

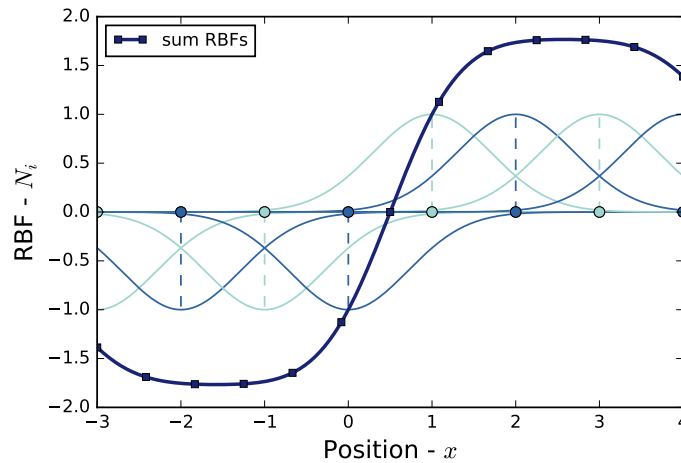


Figure 5.2: A series of 1D RBFs and their summation are plotted. The summation can be regarded as a 1D LSF. The dots on the horizontal axis correspond to the nodes to which an RBF is linked.

The derivatives of the RBFs of figure 5.2 are plotted in figure 5.3. As can be seen by the peak value, the maximum slope is about 2.2094. The maximum slope can also be computed for other smaller bandwidths resulting in the dots on the 1D curve shown in figure 5.4. These points appear to follow the formula shown below (see figure 5.4):

$$|\nabla\phi|_{\max} = 1.2464 \cdot 1.7726^D \cdot \frac{|s_i|_{\max}}{B}, \quad (5.4)$$

where D is the dimension of the problem (e.g. $D = 2$ for 2D problems) and the number 1.2464 is the value relating the maximum slope to the maximum value of $\phi_{\max} = 1.7726^D$. So, the maximum value of the summation also appears into the equation. This value increases for higher dimensional problems, which affects the maximum slope. Basically, equation 5.4 shows that the maximum slope can be controlled by controlling the maximum (absolute) value of the variables s_i .

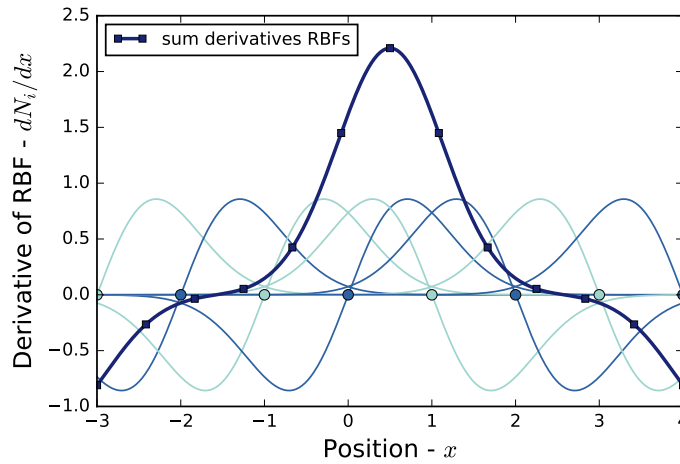


Figure 5.3: Derivatives of the RBFs, already shown in figure 5.2, with respect to the position x . The steepest slope is about 2.2094 as can be seen from the curve representing the summation of these derivatives. The summation can be regarded as the derivative of the 1D LSF.

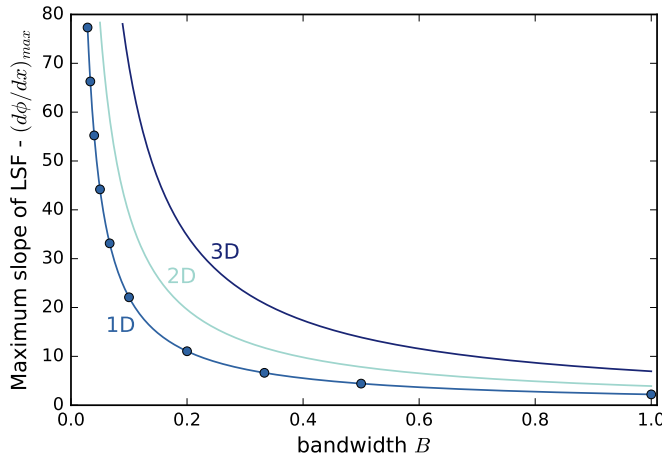


Figure 5.4: The maximum slope that could occur can be scaled by some constant over the bandwidth B as shown in equation 5.4. The dots on the 1D curve show the results from computing the maximum slope for different bandwidths.

5.2.1. Dependency maximum slope on bandwidth

The bandwidth B is equal to the distance between the nodes in the case of figure 5.2. The maximum slope can not be expressed by equation 5.4 any more if the nodal spacing and the bandwidth of the RBFs are different. The maximum value the summation of the RBFs can take is always the same (about 1.77 in 1D), if the nodal spacing and the bandwidth are the same. The slope is roughly the maximum difference in LSF divided by

the distance. The maximum possible jump in the LSF is about two times 1.77 (in 1D) and the distance is defined by the nodal spacing. Hence, the maximum slope in equation 5.4 can be described by the nodal spacing/bandwidth B .

5.3. Calculate Heaviside variable κ

Two aspects of slope control are discussed. On one hand is concluded that the influence of penalizing the intermediate densities forces the optimizer to increase the gradient at the boundary. On the other hand it became clear that the maximum value of the slope of the LSF at the boundary can be controlled by the design variables. This section shows how to compute the ideal steepness of the density function using a maximum value on the design variables.

The distance between two integration points around the boundary can be computed because the number of Gauss integration points in an integration cell and the number of quadtree refinements are known.

$$\Delta x = \frac{B}{2^q \cdot (P + 1)}, \quad (5.5)$$

where B is the bandwidth of the RBF, q is the number of quadtree refinements and P is the polynomial order of the elements (and $P + 1$ is the number of Gauss points in one direction). For the distance between the integration points it is assumed the points are distributed evenly and there is no difference between the horizontal and vertical direction.

The maximum gradient of the LSF can be determined for a certain maximum value of the design variable, e.g. $s_{i,\max} = 0.5$, using equation 5.4. The minimum LSF value at the integration points can be approximated by:

$$\phi_{\min} = |\nabla \phi|_{\max} \cdot \frac{\Delta x}{2}. \quad (5.6)$$

At last this minimum value of the LSF can be related to the derivative of the density with respect to the LSF shown in equation 5.2. A desired minimum value for this derivative should be set, for example 0.5, like done in figure 5.5. The variable κ can then be computed since it is the only variable left in equation 4.13. An iterative procedure is used to solve equation 5.2 for κ . Choosing the desired minimum value for the derivative of the density with respect to the LSF should be done with care.

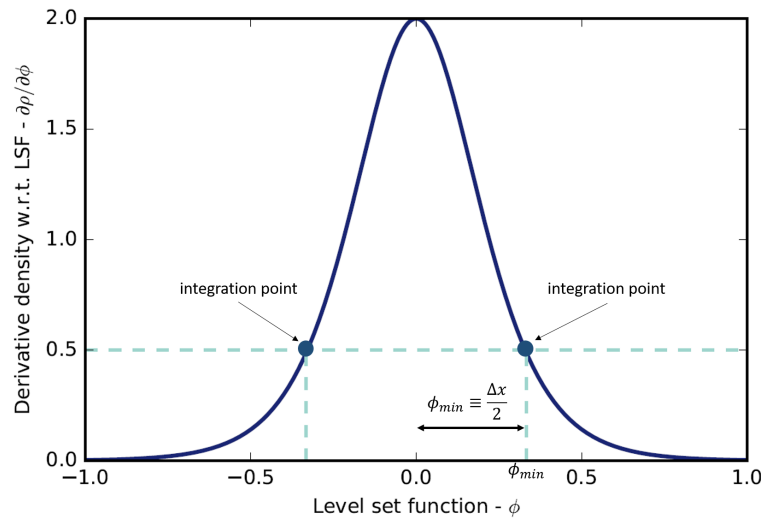


Figure 5.5: Illustration of the minimum value the derivative of the density with respect to the LSF can take at integration points. This graph shows the case for which the boundary is right in between two integration points, which is a worst case. Setting a minimum value for the derivative for these points ensures non-zero derivatives in integration points around the entire boundary of the geometry.

Algorithm 2 in appendix A shows the procedure for calculating κ , as explained in this section, in pseudo-code.

6

Shape optimization

A geometry was extracted from the topology optimized result in stage two. The shape optimization to obtain an optimized design from this extracted geometry, using the previously proposed structural analysis and sensitivity analysis, is discussed in this chapter.

6.1. Optimizers

A gradient-based optimizer uses the sensitivities from chapter 4 as an input to optimize the geometry. Several optimizers exist to evolve the design. The evolution of the LSF could be governed by a Hamilton-Jacobi equation, which can be written as:

$$\frac{\partial C}{\partial \tau} - v_n ||\nabla \phi|| = 0 . \quad (6.1)$$

This differential equation should be solved for ϕ to update the LSF. The sensitivity information can be used to obtain the velocity field v_n . This procedure is for example used by Allaire et al. [2]. Besides solving a Hamilton-Jacobi equation, other methods are Sequential Quadratic Programming (SQP), Method of Moving Asymptotes (MMA) and Interior Point Optimizer (IPOPT). Since MMA is a well established and commonly used optimizer for the SIMP method, this optimizer is chosen for the optimization in stage three.

6.1.1. Method of Moving Asymptotes

The Method of Moving Asymptotes (MMA) was introduced by Svanberg [34]. MMA is a gradient-based optimizer and is based on a special type of convex approximation. MMA is often used in topology optimization and well known for its ability to handle a large amount of design variables. MMA computes a certain change of the design variables Δs_i for every optimization iteration. The step size, i.e. how much the variables can maximally change per iteration, and the convergence tolerance are discussed next.

Move limits

The amount of change in the design variables is related to the move limit parameter α . This parameter controls how far MMA its asymptotes are apart, which influences how much the design variables can change. The value for the move limit parameter should be chosen small compared to the values commonly used for the SIMP method. Firstly because the sensitivities of the shape optimization can be locally (i.e. at the boundary) very high. Secondly, a small change in the design variables s_i can cause a large change in the geometry. For example when all the surrounding variables are zero, the one variable that is positive will have a major influence on the geometry. This combination could result in drastic shape changes. To prevent this and to stabilize the shape optimization, a small value should be chosen for α .

Convergence tolerance

The convergence of SIMP problems is determined by calculating the maximum absolute value of the update of the density variables. Several options could be checked for stage three to determine its convergence. One could check the maximum absolute value of:

- **the update of the design variables Δs_i** - the variables s_i only control the geometry implicitly and a small change in s_i does not necessarily mean a small change in the geometry. Naturally, also a large

change in s_i does not always result in a large geometry change. Roughly the amount of change in s_i and the geometry are linked, but for a convergence tolerance it should be more precise. Checking the actual density changes in the geometry is for example a more precise measurement.

- **the density change in the integration points $\Delta\rho_j$** - the change of density in the integration points ρ_j can be very large, even for small geometry changes. Especially when many integration points are close to the boundary. A small boundary change then already results in the maximum density change in an integration point located close to that boundary. Therefore, this is not a convenient measure to check the convergence of MMA.
- **the density change in an element $\Delta\rho_{el}$** - although it takes most computation time to compute this value, it is the most appropriate value to check for convergence. The density change in an element is basically an average of the density changes in the individual integration points in an element. So, it does not experience the same problems as checking the densities in the integration points directly. Furthermore, it is better comparable with the convergence of the SIMP method in stage one.

So, the maximum absolute change of the density in the elements is determined to check the convergence of stage three. The shape optimization has converged if this value becomes smaller than a chosen convergence tolerance ε .

$$\varepsilon_{\text{LSM}} < |\Delta\rho_{el}|_{\max} \quad (6.2)$$

6.2. Procedure stage three

An example of the proposed shape optimization method is given in this section. Along with this example, further details of the shape optimization of stage three are explained. The cantilever beam problem, as shown by Allaire et al. [2], is mimicked as an example. The cantilever beam is shown in figure 6.1 and its properties are listed below.

- Material properties:
 - Young's modulus: $E = 1$
 - Poisson's ratio: $\nu = 0.3$
- Design domain:
 - Width: $L_x = 2$;
 - Height: $L_y = 1$;
 - Elements in horizontal direction: $n_{el,x} = 80$;
 - Elements in vertical direction: $n_{el,y} = 40$;
 - Volume fraction: $V_{\text{frac}} = 0.4$.
- Boundary conditions:
 - Vertical point load on the middle of the right edge: $F = -1$;
 - Left edge fixed in both the horizontal and vertical direction.

The settings for the shape optimization are:

- Polynomial order analysis: $P = 2$;
- Number of quadtree refinements: $q = 1$;
- Maximum absolute value design variables: $|s_i|_{\max} = 0.5$;
- Penalization intermediate densities: $p_{\text{LSM}} = 3$;
- Desired minimum value of derivative of the density with respect to the LSF in an integration point: $(\partial\rho/\partial\phi)_{\min} = 0.5$;
- Move limit parameter: $\alpha_{\text{LSM}} = 0.20$;

- Convergence tolerance: $\epsilon_{\text{LSM}} = 0.02$.

A small value for the move limit parameter of $\alpha_{\text{LSM}} = 0.20$ is chosen such that the maximum change of 0.20 in the design variables does not result in too drastic shape changes. A downside of a smaller value for α is the slower convergence because more steps are needed to reach the same point.



Figure 6.1: The design requirements for the cantilever beam. A load is applied on the middle of the right edge and the left side is fully clamped. The aspect ratio of the design space is 2:1 and 80x40 elements are used to discretize the structural analysis.

6.2.1. Initial geometry

The starting point for the shape optimization is the geometry extracted at stage two for the post-processing method. For this example, an initial geometry is constructed by constructing an LSF having certain negative parts, see figure 6.2a. The corresponding geometry is shown as well in figure 6.2b and consists of holes where the LSF is negative.

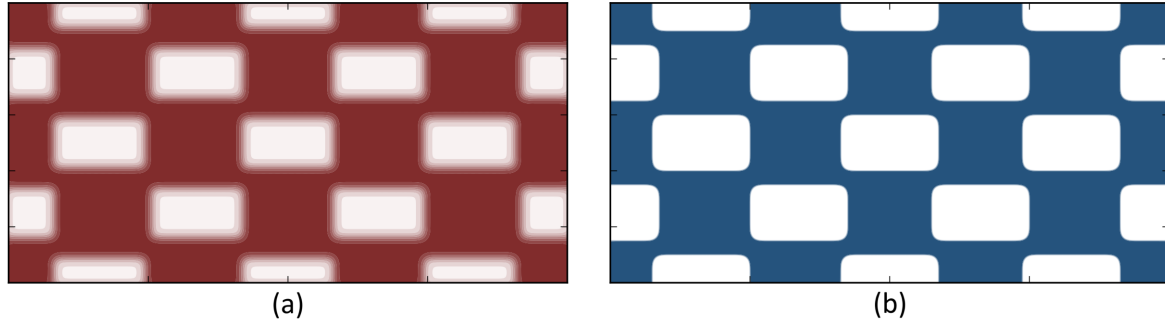


Figure 6.2: The initial LSF constructed by putting adding certain amount of negative parts (a). The corresponding initial geometry is obtained by mapping the LSF with an approximate Heaviside function, resulting in a geometry consisting of holes where the LSF is negative (b).

6.2.2. Quadtree integration band

The use of quadtree integration cells was already introduced in section 3.4 as a part of the structural analysis. There are two reasons to keep a quadtree grid from one iteration to the other for the shape optimization of stage three. One reason is the computational time it takes to build a new grid. The second reason is more important. By changing the integration grid, the optimization problem changes. Therefore, the sensitivities computed in section 4.2 are not consistent from one iteration to the other. As a solution, not only elements intersected by the boundary are refined but also elements that are close to the boundary. This creates an integration band in which the boundary can move during the shape optimization iterations. Any elements that lie outside the integration band are deleted from the analysis to save computation time. Algorithm 3 in appendix A shows the procedure for creating the quadtree integration band in pseudo-code.

The use of a quadtree integration band is most useful if the shape changes are small. For the post-processing method this is the case because the extracted geometry of stage two is quite close to the final design. Figure 6.3 shows the geometry of the cantilever beam after 10 shape optimization iterations with a matching quadtree integration band.

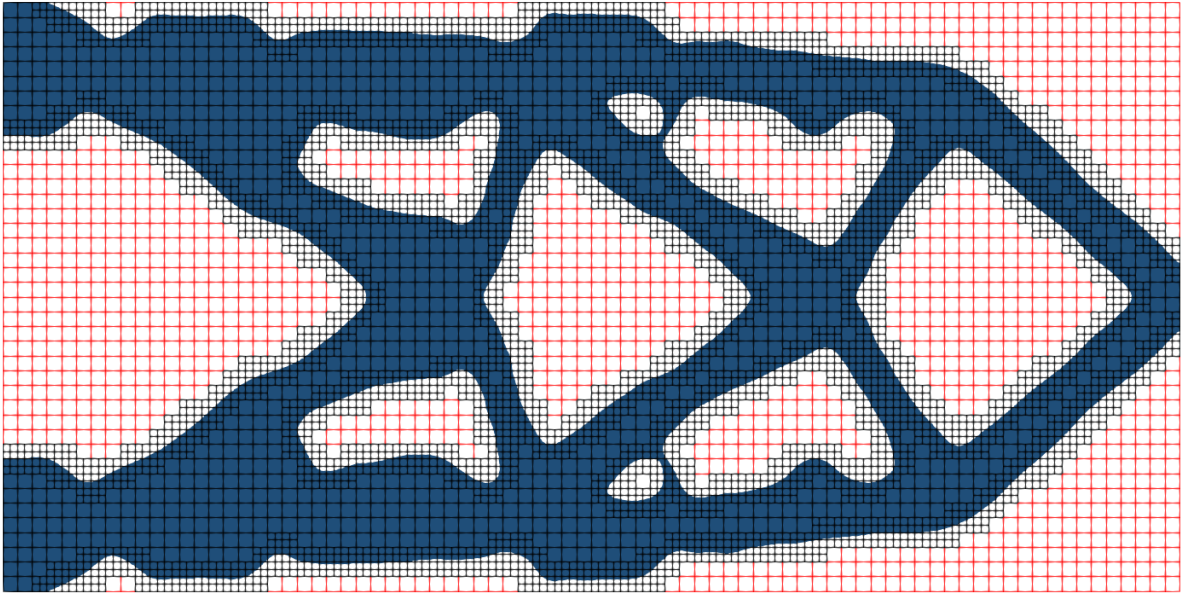


Figure 6.3: The quadtree integration band as defined on the geometry after ten shape optimization iterations. The number of quadtree refinements $q = 1$. The deleted elements are shown in red and the integration cells are shown in black.

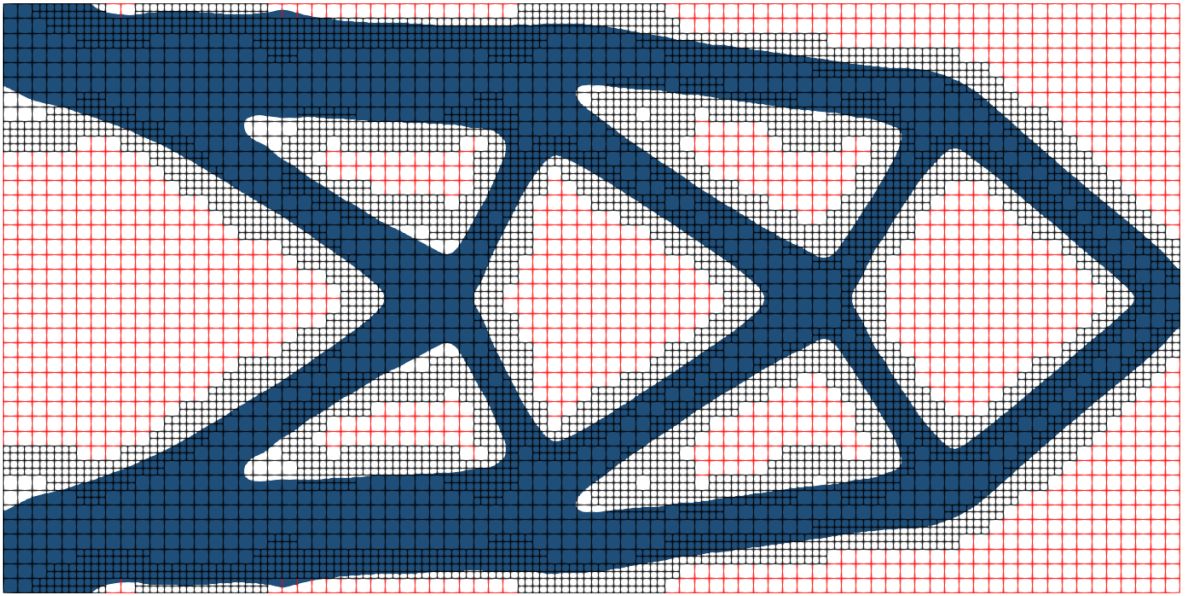


Figure 6.4: Mismatch of the geometry after twenty iterations with the quadtree integration band which was constructed based on the geometry after ten iterations (see figure 6.3). The number of quadtree refinements $q = 1$. The deleted elements are shown in red and the integration cells are shown in black.

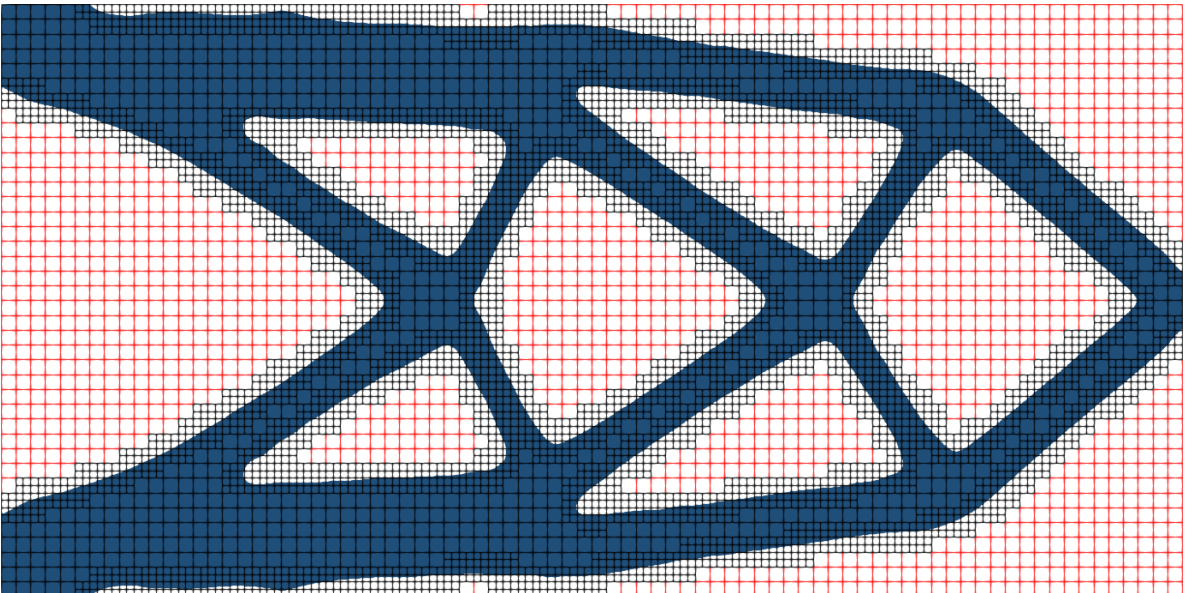


Figure 6.5: Reinitialization of the quadtree integration band on the geometry after twenty shape optimization iterations. The number of quadtree refinements $q = 1$. The deleted elements are shown in red and the integration cells are shown in black.

Boundary location and intersected elements

The geometry is described implicitly by an LSF. It is therefore not straightforward to determine where the boundary of the geometry is and to know where to apply the quadtree refinement. The LSF values at the corner points of a cell are evaluated and checked for a sign change (negative is void, positive is solid). The cell is intersected if the LSF in one or more corner points has a different sign than the LSF in the other corner points. A boundary approximation can be obtained by creating a set of boundary points. These points are defined as the centers of all these intersected cells. This boundary approximation can then be used to create the quadtree integration band by determining whether a cell is within a certain distance from one of the points of the boundary approximation.

6.2.3. Reinitialization

The width of the integration band is set to one element size around the boundary, i.e. half an element on each side of the boundary. It could happen that during the shape optimization the boundary moves outside the integration band creating a mismatch, see figure 6.4.

In that case, a reinitialization of the structural analysis is needed. The quadtree integration band is reinitialized and previously deleted elements can reappear in the structural analysis, see figure 6.5. Ideally, the reinitialization is done as soon as the boundary moves against the outer cells of the integration band. Unfortunately, it requires a lot of computational time to determine the precise location of the boundary and would be a cumbersome procedure to do every iteration of the shape optimization. For this reason the reinitialization is done after a fixed amount of iterations. The number of reinitialization cycles $c_{ri} = 5$ and the maximum number of iterations per cycle $n_{max} = 10$ for the cantilever example in this chapter.

6.2.4. Result

The optimization problem considered in this report is minimizing the compliance (i.e. maximizing the stiffness) subjected to a volume fraction constraint, see section 4.2. The overview is repeated below:

$$\begin{aligned} & \underset{s_i}{\text{minimize}} && C(s_i) \\ & \text{subject to} && g(s_i) \leq 0. \end{aligned}$$

where the objective C is

$$C = \mathbf{f}^T \mathbf{u}, \quad (6.3)$$

and the constraint g is

$$g = \frac{V}{L_x \cdot L_y \cdot V_{frac}} - 1. \quad (6.4)$$

The final optimized design for the cantilever beam is shown in figure 6.6. Obtaining this optimized result takes all 50 iterations, so five reinitialization cycles times ten iterations. The time it takes is about 6.5 minutes.

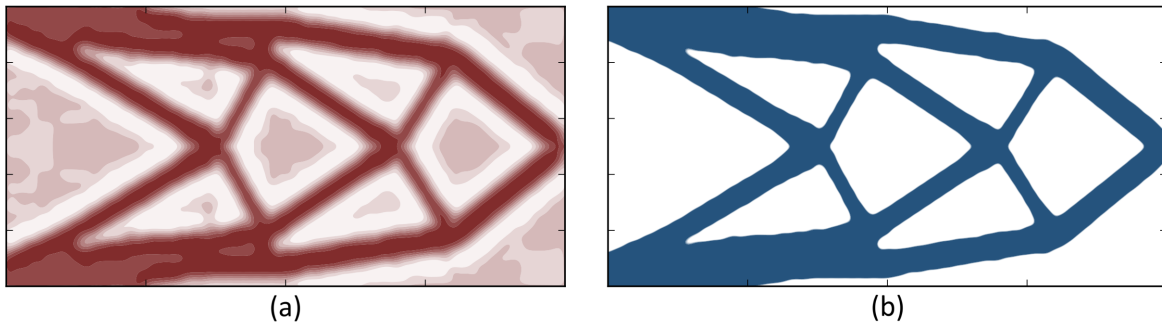


Figure 6.6: The LSF after the shape optimization for the cantilever beam problem (a) and the corresponding optimized geometry by mapping the LSF with an approximate Heaviside function (b).

6.3. Final design to CAD

The optimized geometry obtained at stage three should be translated into a common CAD format. The use of other open source programs is very convenient for this purpose. The density of the domain can be sampled

at a certain amount of pixels in the domain. The amount of pixels should be chosen such that a sufficient resolution of the geometry is obtained. These pixels can be written to a .vtk-file, a file format that can be used in Paraview. Paraview is a post-processing visualization tool. This program is also capable of thresholding the density image and converting it into an .x3d-file. These file types can for example be opened in MeshLab or Blender, open source CAD programs. Besides that, these programs can construct .stl-files of the geometry which makes it possible to 3D-print.

Case studies

The implemented method for post-processing topology optimized results is tested for several cases. The main inconvenience of topology optimized results using the SIMP method are the elements with intermediate densities and the jagged boundaries. These two properties are for example compared between the three stages of the design optimization process. All the properties that will be compared are listed below.

- **Crispness** - the amount of intermediate densities of a design can be 'measured' using the following formula introduced by Sigmund [32],

$$M_{nd} = \frac{\sum_{e=1}^n 4\rho_e(1 - \rho_e)}{n} \cdot 100\%, \quad (7.1)$$

where ρ_e is the element density and n is the number of elements. The crispness of the designs described by projecting an LSF is measured by sampling the density in small pixels. Then n is the number of pixels and ρ_e is the density in a pixel.

- **Smoothness** - this property is hard to quantify so will be judged qualitatively (bad, poor, sufficient, good, excellent).
- **Compliance** - this is the objective of the optimization and should at least improve from stage two to three. The compliances are computed using the penalization of intermediate densities. The compliance of the topology optimized result is not compared with stage two and three because the analyses between those stages differ too much.
- **Iterations** - the number of iterations every stage takes until convergence or if the maximum amount of iterations is reached. The number of iterations per reinitialization cycle are given as well for stage three.
- **Computation time** - the time every stage takes will be measured to give an indication of the computation time.

Besides comparing the results obtained for the three stages of the structural design optimization, the performance of the shape optimization is also of interest. Stage three is tested for the following parameters/settings.

- **Accuracy structural analysis** - the polynomial order and the number of quadtree refinements can be increased or decreased. Changing the accuracy of the structural analysis probably has a large influence on the computation time.
- **Penalization intermediate densities** - the amount of intermediate densities could be reduced by increasing the penalization parameter p_{LSM} . However, it could be that the shape optimization shows a worse convergence.
- **Move limit parameter** - the maximum update of the design variables every shape optimization iteration is controlled by the move limit parameter α_{LSM} of MMA. This parameter influences how fast and how stable the shape optimization converges.

Instead of comparing compliance values for the performance study of stage three, the strain energy of the final design is computed. This is done using the same analysis settings for a good comparison.

- The polynomial order: $P = 5$.
- Number of quadtree refinements: $q = 1$.
- Penalization of intermediate densities: $p_{\text{LSM}} = 1$.
- The Heaviside parameter: $\kappa = 1000$, which creates a very steep density function resulting in almost no intermediate densities.

The smoothness property is ignored for the performance study because changing the polynomial order, the number of quadtree refinements, the penalization of intermediate densities or the move limit parameter does not influence the smoothness of the design. The smoothness only depends on the LSF, which is always smooth and independent of these parameters.

7.1. General optimization settings case studies

The optimization problem considered in this report is minimizing the compliance (i.e. maximizing the stiffness) subjected to a volume fraction constraint, see section 4.2. The overview is repeated below:

$$\begin{aligned} & \underset{s_i}{\text{minimize}} && C(s_i) \\ & \text{subject to} && g(s_i) \leq 0. \end{aligned}$$

where the objective C is

$$C = \mathbf{f}^T \mathbf{u}, \quad (7.2)$$

and the constraint g is

$$g = \frac{V}{L_x \cdot L_y \cdot V_{\text{frac}}} - 1. \quad (7.3)$$

At first, all cases are optimized using precisely the same optimization settings for stage one and three. The topology optimization of stage one is done using the Python implementation of the 99-line MATLAB code of Sigmund [31]. Stage two is independent of parameters that should be chosen in advance.

- Stage one - Topology optimization
 - Penalization intermediate densities: $p_{\text{SIMP}} = 3$;
 - Filter radius: $r = 1.5$ elements;
 - Maximum iterations: $n_{\text{max}} = 100$;
 - Move limit parameter: $\alpha_{\text{SIMP}} = 1.0$.
- Stage three - Shape optimization
 - Polynomial order analysis: $P = 2$;
 - Number of quadtree refinements: $q = 1$;
 - Maximum absolute value design variables: $|s_i|_{\text{max}} = 0.5$;
 - Penalization intermediate densities: $p_{\text{LSM}} = 3$;
 - Desired minimum value of derivative of the density with respect to the LSF in an integration point: $(\partial\rho/\partial\phi)_{\text{min}} = 0.5$;
 - Number of reinitialization cycles and maximum iterations per cycle: $c_{\text{ri}} = 2$ and $n_{\text{max}} = 5 + c_{\text{ri}} \cdot 5$;
 - Move limit parameter: $\alpha_{\text{LSM}} = 0.10$.

The geometry is more likely to go outside the quadtree integration band at the beginning of stage three because the shape changes are relatively large in the first iterations. So, the maximum number of iterations per reinitialization cycle n_{max} is smallest at first and increases for every cycle c_{ri} .

7.2. MBB beam

The MBB beam was already shown in the introduction of this report and is the first test case to show the method for post-processing topology optimized results. The following list provides the details specific to this MBB beam problem.

- Design domain:
 - Width: $L_x = 1$;
 - Height: $L_y = 1/3$;
 - Elements in horizontal direction: $n_{el,x} = 30$;
 - Elements in vertical direction: $n_{el,y} = 10$;
 - Volume fraction: $V_{frac} = 0.4$.
- Boundary conditions:
 - Vertical point load on top right corner: $F = -1$;
 - Right bottom corner fixed in vertical direction;
 - Left edge fixed in horizontal direction.
- Convergence tolerances:
 - $\epsilon_{SIMP} = \epsilon_{LSM} = 0.02$

7.2.1. Results MBB beam

The results for the MBB beam case are shown in figure 7.1. Table 7.1 compares the properties of the results obtained from the three different stages. The crispness improves by a factor 5.5 from stage one to three and the smoothness of the final design is excellent due to the LSF description. The shape optimization in stage three reduces the compliance by 10.5% compared to stage two. Furthermore, the post-processor was slightly faster than the topology optimization in stage one.

Table 7.1: The results for different properties of the design stages for the MBB beam case. The iterations for stage three are given as a total and between the brackets the number of iterations before and after the reinitialization.

Properties	MBB beam Stage 1 Figure 7.1b	MBB beam Stage 2 Figure 7.1d	MBB beam Stage 3 Figure 7.1f
Intermediate densities (%)	43.6	14.2	8.0
Smoothness	Bad	Excellent	Excellent
Compliance	356.6	264.9	237.2
Iterations	71	-	17 (5,10,2)
Computation time (s)	8.7	0.4	7.9

7.2.2. Performance stage three for MBB beam

In this paragraph the performance of the shape optimization is addressed. At the start of chapter 7 it was already mentioned that different settings for the shape optimization are altered to see the influence on the results. The performance of stage three can be inspected via the properties of the final design. The results are shown in appendix B, in tables B.1 till B.4.

Polynomial order

The shape optimization using a polynomial order $P = 5$ did not converge. Hence, the maximum amount of iterations is used and the strain energy is very high. The shape changes got too large at a certain point in the shape optimization, resulting in a member to vanish. Usually, at a place where is no material, there are no sensitivities. So, it is hard for the shape optimization to make a member reappear again. A cure would be to use a lower move limit parameter, preventing a member from disappearing in the first place.

Increasing the polynomial order from $P = 1$ to $P = 4$ improves the crispness of the design by a factor 2.2 but computation time increases by a factor 5. For the lowest polynomial order $P = 1$, the final design was about 1.7% worse than the best obtained design for $P = 3$ and $P = 4$.

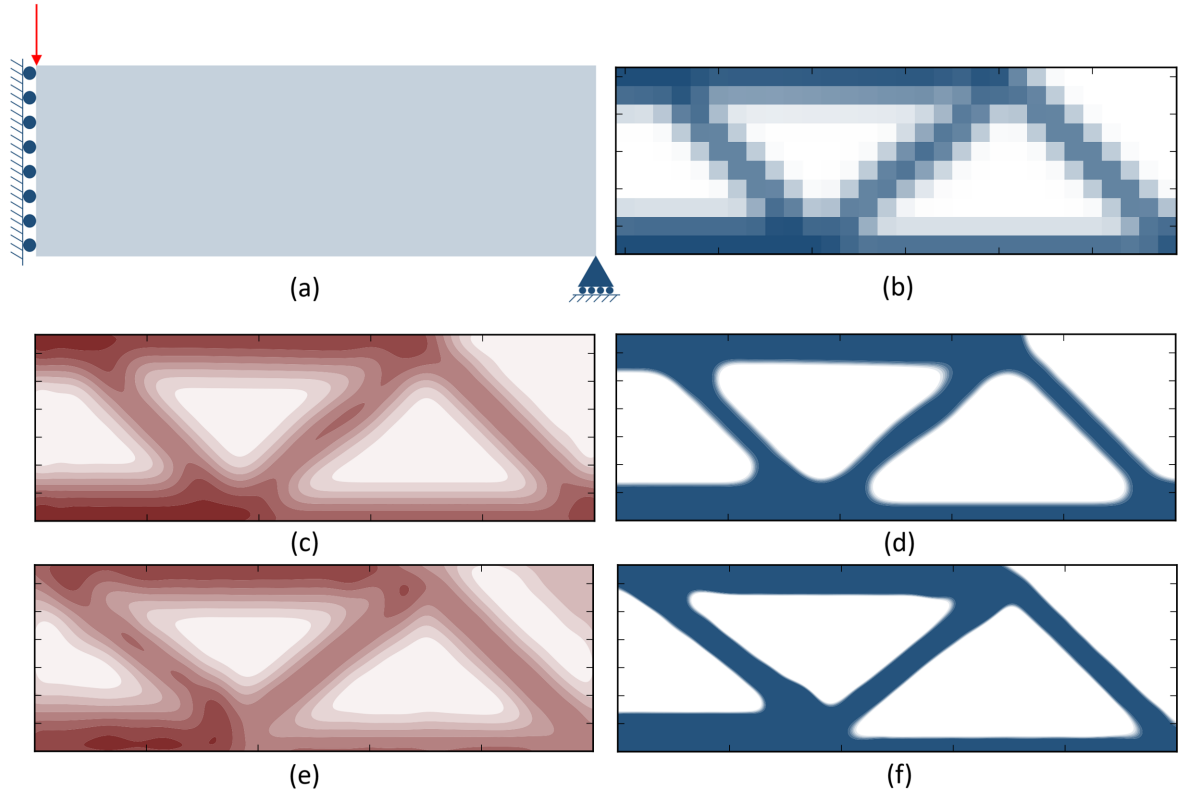


Figure 7.1: The entire design optimization process for the MMB beam case. The design requirements (a), the topology optimized result using the SIMP method (b), the LSF after stage two (c), the geometry of stage two by mapping the LSF with an approximate Heaviside function (d), the optimized LSF after stage three (e) and the final optimized design (f). Please note, the colorscales of the two LSFs (c) and (e) are different.

Number of quadtree refinements

Increasing the number of quadtree refinements from $q = 0$ to $q = 3$ improves the crispness by a factor of 7.3 whereas the computation time increases by a factor 6.9. For $q = 2$ and $q = 3$ the strain energies are the lowest and for $q = 0$ the highest, with a difference of 1.2% between the two.

Penalization of intermediate densities

The shape optimization using $p_{\text{LSM}} = 1$ did not converge. A shape change that was too large occurred during one of the iterations, resulting in a member being deleted. Hence, the high strain energy and the use of the maximum amount of iterations.

Increasing the penalization shows an expected improvement in crispness. The crispness improves by a factor 1.5 between $p_{\text{LSM}} = 2$ and $p_{\text{LSM}} = 5$ without taking too much extra time for the computations. On the other hand, the best design in terms of strain energy is obtained using $p_{\text{LSM}} = 2$.

Move limit parameter

The convergence started to get unstable and bad for a value of $\alpha_{\text{LSM}} = 0.28$ and higher. Larger values resulted in too drastic shape changes during the optimization, resulting in members being removed which could (almost) not reappear again. A value of 0.28 can be regarded as the critical move limit value for the MBB beam case.

Choosing the move limit parameter lower than the 'critical' value, results in final designs having almost the same strain energy value (all within 1%). However, the amount of intermediate densities can be reduced by almost a factor 2 for similar computation times.

7.2.3. Check sensitivities

The LSF slope control, introduced in chapter 5, can be checked for the final design. The derivative of the density with respect to the LSF should not be zero in the integration points at the boundary. Figure 7.2 shows this derivative for the final design of the MBB beam. Clearly, the derivative is non-zero at the boundary of the

structure. Furthermore, a closer look at the boundary reveals that there are always some integration points containing non-zero derivative values around the entire boundary.

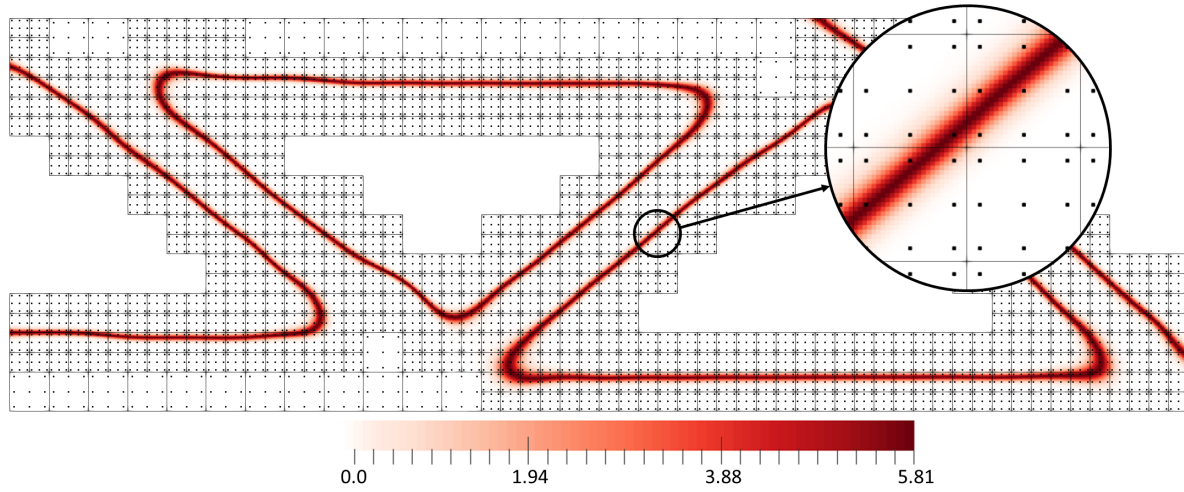


Figure 7.2: The derivative of the density with respect to the LSF ($\partial\rho/\partial\phi$) in the entire design domain of the final design of the MBB beam. The derivative is only non-zero at the boundaries of the design. The black cells are the integration cells and the dots inside are the integration points. At the boundary there are always a few integration points positioned where the derivative is non-zero.

7.3. Cantilever beam

The following list provides the details specific to this cantilever beam problem.

- Design domain:
 - Width: $L_x = 1$;
 - Height: $L_y = 1/3$;
 - Elements in horizontal direction: $n_{el,x} = 48$;
 - Elements in vertical direction: $n_{el,y} = 16$;
 - Volume fraction: $V_{frac} = 0.4$.
- Boundary conditions:
 - Vertical point load on the middle of the right edge: $F = -1$;
 - Left edge fixed in both the horizontal and vertical direction.
- Convergence tolerances:
 - $\epsilon_{SIMP} = \epsilon_{LSM} = 0.03$

7.3.1. Results cantilever beam

The results are shown in figure 7.3. Table 7.2 compares the properties of the designs obtained at the three different stages. Just as for the MBB beam case, the crispness improves a lot for the cantilever beam, namely by a factor 5.3 from stage one to three. Also the smoothness improves because of the use of an LSF. The shape optimization in stage three reduces the compliance by 11.4% compared to stage two.

7.3.2. Performance stage three for cantilever beam

In this paragraph the performance of the shape optimization is addressed. At the start of chapter 7 it was already mentioned that different settings for the shape optimization were going to be altered to see the influence on the results. The performance of stage three can be inspected via the properties of the final design. The results are shown in appendix B, in tables B.5 till B.8.

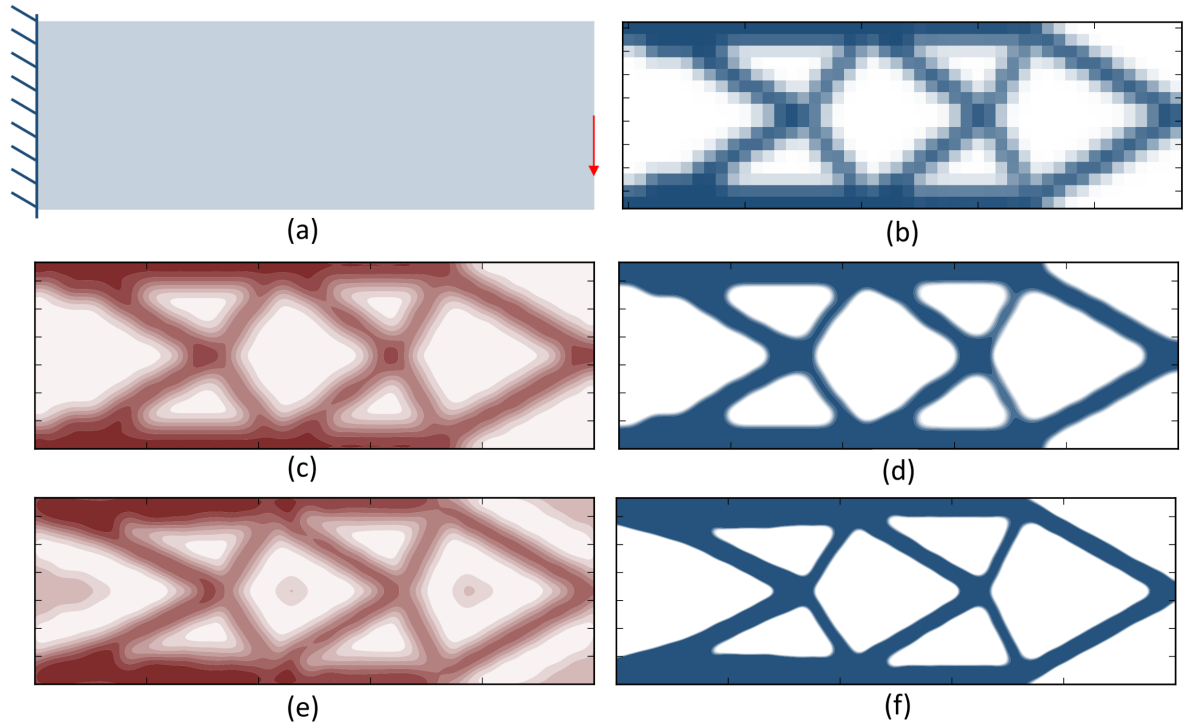


Figure 7.3: The entire design optimization process for the cantilever beam case. The design requirements (a), the topology optimized result using the SIMP method (b), the LSF after stage two (c), the geometry of stage two by mapping the LSF with an approximate Heaviside function (d), the optimized LSF after stage three (e) and the final optimized design (f). Please note, the colorscales of the two LSFs (c) and (e) are different.

Polynomial order

Increasing the polynomial order from $P = 1$ to $P = 5$ improves the crispness of the design by a factor 2.7 but computation time increases by a factor 12. For the lowest polynomial order $P = 1$, the final design was about 1.9% worse than the best obtained design for $P = 5$.

Number of quadtree refinements

Increasing the number of quadtree refinements from $q = 0$ to $q = 3$ improves the crispness by a factor of 10.8 whereas the computation time increases by a factor 6.6. For $q = 2$ the strain energy is the lowest and for $q = 0$ the highest, with a difference of 2.1% between the two.

Penalization of intermediate densities

The shape optimization using $p_{\text{LSM}} = 1$ did not converge. A shape change that was too large occurred during one of the iterations, resulting in a member being deleted. Hence, the high strain energy and the use of the maximum amount of iterations.

Increasing the penalization shows an expected improvement in crispness. The crispness improved by a factor 1.5 between $p_{\text{LSM}} = 2$ and $p_{\text{LSM}} = 5$ without taking too much extra time for the computations. On the other hand, the best design in terms of strain energy is obtained using $p_{\text{LSM}} = 2$.

Move limit parameter

The convergence started to get unstable and bad for a value of $\alpha_{\text{LSM}} = 0.28$ and higher. Larger values resulted in too drastic shape changes during the optimization, resulting in members being removed which could (almost) not reappear again. A value of 0.28 can be regarded as the critical move limit value for the MBB beam case.

The amount of intermediate densities can be reduced by more than a factor 2 for similar computation times, if a move limit is chosen closer to the 'critical' value. The cantilever beam case showed a very fast convergence for the smallest tested move limit of $\alpha_{\text{LSM}} = 0.05$. However, this resulted in a relatively poorly performing design.

Table 7.2: The results for different properties of the design stages for the cantilever beam case. The iterations for stage three are given as a total and between the brackets the number of iterations before and after the reinitialization.

Properties	Cantilever beam Stage 1 Figure 7.3b	Cantilever beam Stage 2 Figure 7.3d	Cantilever beam Stage 3 Figure 7.3f
Intermediate densities (%)	38.2	12.5	7.2
Smoothness	Bad	Excellent	Excellent
Compliance	292.5	236.1	209.1
Iterations	99	-	17 (5,10,2)
Computation time (s)	30	1.5	23

7.4. Bridge with distributed load

The following list provides the details specific to this bridge with distributed load problem.

- Design domain:
 - Width: $L_x = 1$;
 - Height: $L_y = 0.5$;
the bottom row of elements are not part of the design domain for the topology optimization and are always solid
 - Elements in horizontal direction: $n_{el,x} = 50$;
 - Elements in vertical direction: $n_{el,y} = 25$;
 - Volume fraction: $V_{frac} = 0.4$.
- Boundary conditions:
 - Vertical distributed load on the bottom edge: $F_q = -1$;
 - Left and right bottom corners fixed in both the horizontal and vertical direction;
 - Another two points on the left and right, at $y = 0.06$, fixed in the both the horizontal and vertical direction.
- Convergence tolerances:
 - $\epsilon_{SIMP} = \epsilon_{LSM} = 0.03$

7.4.1. Results bridge with distributed load

The results are shown in figure 7.4. Table 7.3 compares the properties of the results obtained from the three different stages. Just as for the previous cases, the crispness improves a lot for the bridge with distributed load, namely by a factor 5.6 from stage one to three. Also the smoothness improves because of the use of an LSF. The shape optimization in stage three reduces the compliance by 7.9% compared to stage two.

Table 7.3: The results for different properties of the design stages for the bridge with distributed load case. The iterations for stage three are given as a total and between the brackets the number of iterations before and after the reinitialization.

Properties	Bridge Stage 1 Figure 7.4b	Bridge Stage 2 Figure 7.4d	Bridge Stage 3 Figure 7.4f
Intermediate densities (%)	25.8	7.6	4.6
Smoothness	Bad	Excellent	Excellent
Compliance	4.22	3.82	3.52
Iterations	79	-	17 (5,10,2)
Computation time (s)	13	2.5	35

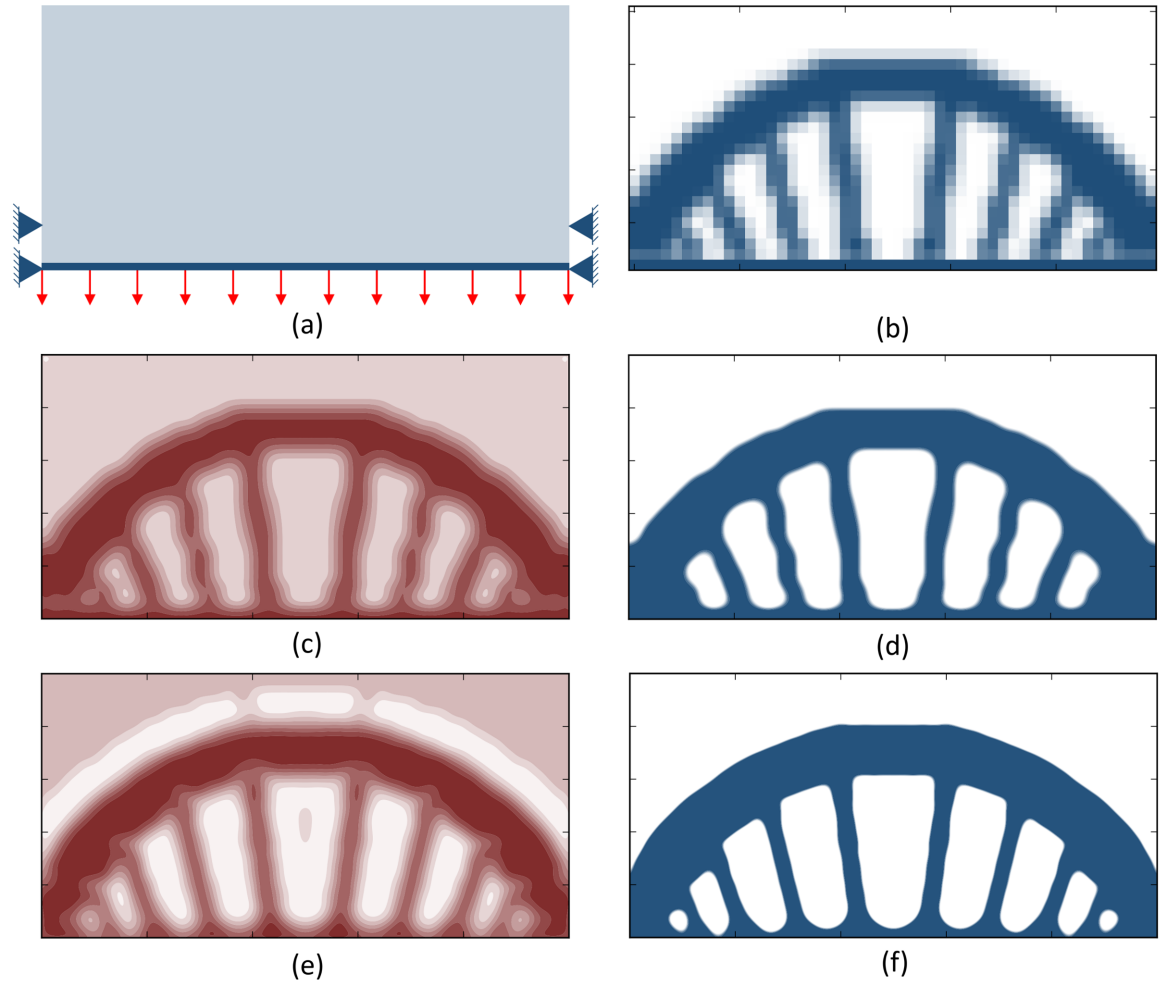


Figure 7.4: The entire design optimization process for the bridge with distributed load case. The design requirements (a), the topology optimized result using the SIMP method (b), the LSF after stage two (c), the geometry of stage two by mapping the LSF with an approximate Heaviside function (d), the optimized LSF after stage three (e) and the final optimized design (f). Please note, the colorscales of the two LSFs in (c) and (e) are different.

7.4.2. Performance stage three for bridge with distributed load

In this paragraph the performance of the shape optimization is addressed. At the start of chapter 7 it was already mentioned that different settings for the shape optimization were altered to see the influence on the results. The performance of stage three can be inspected via the properties of the final design. The results are shown in appendix B, in tables B.9 till B.12.

Polynomial order

Increasing the polynomial order from $P = 1$ to $P = 5$ improves the crispness of the design by a factor 2.9 but computation time increases by a factor 14.2. For the lowest polynomial order of $P = 1$, the final design is about 0.5% worse than the best obtained design for $P = 5$.

Number of quadtree refinements

Increasing the number of quadtree refinements from $q = 0$ to $q = 3$ improves the crispness by a factor of 13.5 whereas the computation time increases by a factor 4.5. For $q = 1$ the strain energy is the lowest and for $q = 0$ the highest, with a difference of 0.5% between the two.

Penalization of intermediate densities

The shape optimization using $p_{\text{LSM}} = 1$ did not converge. A shape change that was too large occurred during one of the iterations, resulting in a member being deleted. Hence, the high strain energy and the use of the maximum amount of iterations.

Increasing the penalization shows an expected improvement in crispness. The crispness improved by a factor

1.5 between $p_{\text{LSM}} = 2$ and $p_{\text{LSM}} = 5$ without taking too much extra time for the computations. On the other hand, the best design in terms of strain energy is obtained using $p_{\text{LSM}} = 2$.

Move limit parameter

The convergence started to get unstable and bad for a value $\alpha_{\text{LSM}} = 0.40$ and higher. Larger values resulted in too drastic shape changes during the optimization, resulting in members being removed which could (almost) not reappear again. A value of 0.40 can be regarded as the critical move limit value for the bridge case. Choosing the move limit parameter lower than the 'critical' value, results in final designs having almost the same strain energy value (all within 0.5%). However, the amount of intermediate densities can be reduced by almost a factor 2 for similar computation times.

7.4.3. Topological change

The results for the bridge with distributed load case showed a different topology between stage two and three (c.f. figure 7.4d and figure 7.4f). Two new holes appeared in the final design. A closer look at the sensitivities of the extracted geometry at stage two explains how the topological change could occur. The sensitivities at the position where the new holes emerged are actually non-zero as can be seen in figure 7.5.

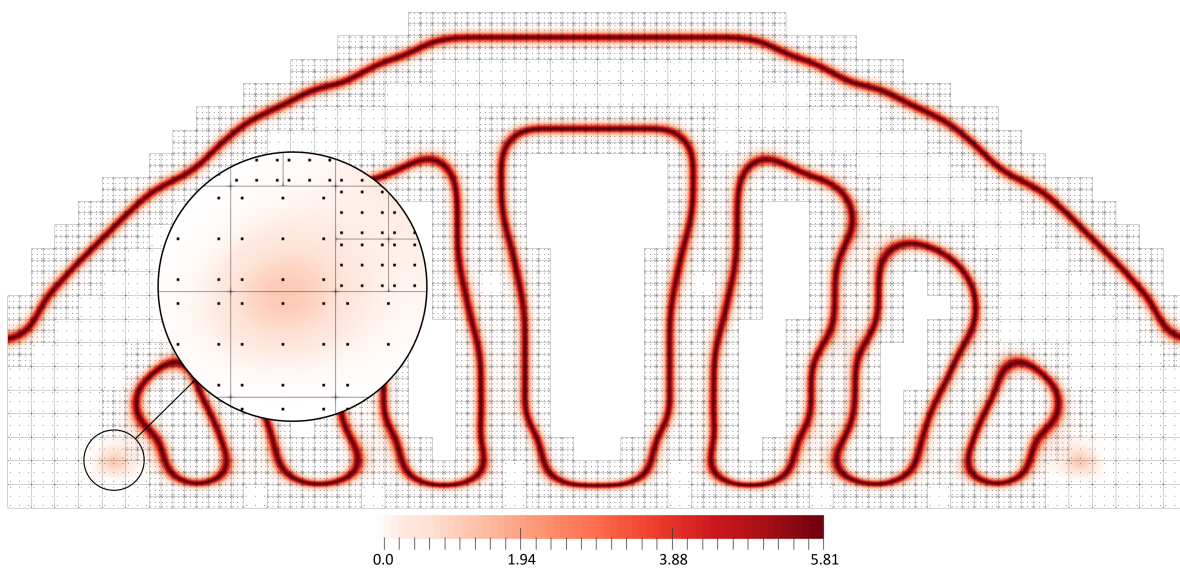


Figure 7.5: The derivative of the density with respect to the LSF ($\partial\rho/\partial\psi$) of the extracted geometry of the bridge with distributed load. The derivative is non-zero close to the bottom corners, allowing a change in the geometry at that region. The black squares are the integration cells and the dots inside are the integration points.

7.5. Conclusions case studies

Many conclusions can be drawn from the case studies and the corresponding performance studies. The conclusions for the post-processor are listed first.

- **Crispness** - the post-processor decreases the amount of intermediate densities between the topology optimized result and the final design with an average factor of 5.5. A fully crisp geometry cannot be forced because then sensitivity analysis difficulties occur.
- **Smoothness** - the post-processor provides smooth final designs due to the use of a smooth LSF as an implicit description of the geometry.
- **Compliance** - the compliance reduces by an average 10% from stage two to three due to the shape optimization. This is a significant reduction, showing the extracted geometry at stage two is not yet an optimized design. The compliance values of the final design and the topology optimized result are not compared because the structural analyses of both differ too much.
- **Iterations** - the shape optimization uses less iterations than the topology optimization to reach the same convergence tolerance. This makes sense because the starting design for the shape optimization

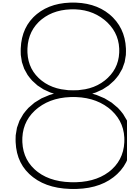
is a lot closer to an optimized result than for the topology optimization. Because of this, the size of the problem can increase from stage one to stage three (more DOF and more integration points), while the computation time of the shape optimization is in the same order of magnitude as the topology optimization.

Many simulations converge in the second iteration of the last reinitialization cycle, so after 17 (5,10,2) iterations. Further research is needed to determine why this is. It could be that steps are generally smaller in the first iterations after a reinitialization because MMA is initializing during its first two optimization iterations. If the steps are small enough, the convergence tolerance will be reached right away.

- **Computation time** - the computation time is strongly problem dependent. Out of the three cases, the post-processor is one time slower, one time equally fast and one time faster than the topology optimization in stage one. A large part, more than 10%, of the computation time is spent on creating quadtree integration bands. A big improvement would be to fasten this process.

The conclusions for the performance studies are listed next.

- **Polynomial order structural analysis** - an increase in the polynomial order for the structural analysis of stage three does not improve the final design very much in terms of the strain energy. Almost all final designs are within an average of 1.5% from each other for different polynomial orders. The only polynomial increase worth the increase in computation time is from $P = 1$ to $P = 2$. This can also be expected from section 3.5.3, where an increase from $P = 1$ to $P = 2$ shows the largest absolute improvement in the strain energy versus the increase in DOF.
- **Number of quadtree refinements structural analysis** - an increase in the number of quadtree refinements for the structural analysis of stage three does not improve the final design very much in terms of the strain energy. All results are within an average of 1.5% from each other. This can be expected for increasing the number of quadtree refinements. The example studies done on FCM, see section 3.5.3, already showed that two or more quadtree refinements do not contribute to a better structural analysis. Hence, the shape optimization is not very likely to do better either.
A higher accuracy of the structural analysis shows a strong reduction in the intermediate densities. This reduction is because the steepness of the approximate Heaviside is based on the maximum distance between two integration points around the boundary. An higher accuracy of the structural analysis results in more integration points, which results in a steeper approximate Heaviside projection. Furthermore, increasing the number of quadtree refinements shows a better reduction relative to an increase in computation time than increasing the polynomial order for this purpose.
- **Penalization of intermediate densities** - penalization shows to be an efficient method to reduce the amount of intermediate densities. It does not affect the computation time too much but it does reduce the amount of intermediate densities by an average factor 1.5 between $p_{LSM} = 2$ and $p_{LSM} = 5$.
The strain energy was very high for final designs using $p_{LSM} = 1$. This has two reasons. Firstly, the structural analysis, calculating the final performance, uses a steep approximate Heaviside projection of the LSF, creating a geometry which barely contains intermediate densities. The result using $p_{LSM} = 1$ contained relatively much intermediate densities resulting in some members becoming extremely thin when a steep projection is used. Secondly, the shape optimization for $p_{LSM} = 1$ does not converge properly because it uses its maximum amount of iterations.
The penalization and the LSF maximum slope control are a contradiction. Penalization causes the LSF to be as steep as possible, while only a certain configuration of RBFs allows for the maximum slope. The effect of penalization on the optimizer is that it not only looks for an optimal design but also a design which allows the steepest slope around the entire boundary, resulting in less design freedom. Therefore, it makes sense that lower penalization values could end up in better designs in terms of strain energy. The final designs obtained using $p_{LSM} = 2$ performed best in terms of the strain energy.
- **Move limit parameter** - higher move limit values show better final results in terms of crispness and strain energy for comparable computation times. However, a too large move limit parameter ends up in bad and unstable convergence. Unfortunately, it is hard to determine a priori whether a move limit parameter is too large.



Conclusions and recommendations

The scope of this MSc. Thesis is to come up with a post-processing method for topology optimized results which should obtain smooth and crisp geometries, resulting in a complete structural design optimization process. Topology optimization commonly uses a density based method called SIMP. The main two drawbacks of the SIMP method are the intermediate densities and the jagged boundaries in the optimized geometry. The post-processor should get rid of these features. Furthermore, it should conveniently couple the three stages of the structural design optimization process.

8.1. Goals post-processing method

Several goals were set in the introduction of this MSc. Thesis for the method of post-processing topology optimized results. A conclusion is drawn for every goal.

8.1.1. Usable final geometry

The final geometry should be of direct use. This means the final design should be: crisp (i.e. no intermediate densities), smooth (i.e. no jagged boundaries) and convertible into a CAD file format.

Crispness

The final design still contains a small amount of intermediate densities. A fully crisp geometry cannot be forced because then sensitivity analysis difficulties occur. This makes the use of a blurred boundary containing intermediate densities inevitable.

Penalization of intermediate densities is used to create designs which are as crisp as possible. Intermediate densities occur at places where the LSF is close to zero. A steep gradient (i.e. a steep transition from void to solid) around the zero-level contour minimizes the amount of intermediate densities. The optimizer pushes the LSF to a steep gradient when penalization is used. The intermediate densities reduce an average factor of 5.5 between the topology optimized result and the final design. An even better improvement of the crispness was obtained by using a higher penalization, higher accuracy of the structural analysis or a higher move limit parameter.

Smoothness

The summation of RBFs results in a smooth LSF. The actual geometry is obtained by projecting the LSF to a material distribution, which is smooth as well.

Computer Aided Design

The material distribution of the geometry is not explicitly known, but only implicitly via an LSF. The design domain can be divided in a lot of pixels and the density can be sampled in the middle of each pixel. This results in a detailed geometry if a sufficient resolution of the pixels is chosen. Using a combination of open source programs such pixel descriptions can be easily converted into an .stl-file, which is suitable for importing into CAD environments.

8.1.2. Simple and all-round tool

The implementation should be straightforward and result in a fully integrated structural design optimization process.

Model and result stage one

The structural analysis uses FCM, which is a fictitious domain method. This allows for the direct use of the mesh used for the topology optimization.

The convergence of the optimizer is determined in a similar fashion for both the topology and the shape optimization. The density change in an element is determined for both. The optimizer has converged when the maximum absolute elemental density change is below a certain threshold.

Potentially, the proposed post-processing method can handle all kind of problems which can also be optimized using the SIMP method. The shape optimization uses a combination of derivatives which are either for every type of problem the same or can be directly used from the SIMP formulation. This report only presents compliance related optimization problems but it should be easy to extend the post-processor to other optimization problems.

99-line MATLAB code

The Python version of the 99-line MATLAB code is used for stage one. The element densities obtained from this code are directly used by the post-processor by translating them to nodal densities. These nodal densities are then used to set up an LSF.

Extension to 3D

This MSc. Thesis does not contain 3D case studies but at least in theory the proposed post-processor is extendible to 3D. First of all, the implicit geometry description using an LSF can also be used in 3D by using the 3D version of the Gaussian RBF. Also the fictitious domain method for the structural analysis can be extended by using octree integration refinements instead of quadtree. Other than adaptive numerical integration, pFEM can also be extended to 3D.

8.1.3. Fast

The goal is that the post-processing is done in a reasonable amount of time. It should not take longer than the topology optimization in stage one. The speed of the post-processor relative to the topology optimization seems strongly dependent on the problem. Out of the three cases, the post-processor is one time slower, one time equally fast and one time faster than the topology optimization in stage one. So, this goal is partly met. The shape optimization uses less iterations than the topology optimization to reach the same convergence tolerance. This makes sense because the starting design for the shape optimization is a lot closer to an optimized result than for the topology optimization. Because of this, the size of the problem can increase from stage one to stage three (more DOF and more integration points), while the computation time of the shape optimization is in the same order of magnitude as the topology optimization.

8.2. Other findings

Along with the development of a post-processing method for topology optimized results, other findings are made as well. Some conclusions can be drawn besides the conclusions regarding the goals set for the post-processor.

8.2.1. Structural analysis

The structural analysis during stage three is done using a combination of pFEM and adaptive integration using quadtree refinements. This combination is also known as FCM. The use of more than one quadtree refinement is shown to be mostly unnecessary to obtain more accurate results because the discretization error is already dominant for one quadtree refinement. Especially when the approximate Heaviside function is used to project the LSF to densities. This creates a smoother transition from solid to void, which apparently requires a less accurate numerical integration.

The convergence of pFEM is exponential, meaning that the largest absolute improvement of the structural analysis is reached by increasing the polynomial order from $P = 1$ to $P = 2$. For the shape optimization only this increase in polynomial order is worth the increase in computation time.

8.2.2. LSF slope control

The sensitivities of the objective and constraint are needed for the shape optimization of stage three. A critical partial derivative is the derivative of the density with respect to the LSF. This derivative becomes infinite at the boundary and zero elsewhere if the gradient of the LSF is too steep. The use of penalization pushes the LSF to a steep gradient to minimize the amount of intermediate densities. Therefore, a slope control of the LSF was

introduced by bounding the design variables s_i . Eventually, using the knowledge of the maximum possible gradient of the LSF, the parameter κ can be calculated. This method for slope control is proven to provide non-zero values for the derivative of the density with respect to the LSF at the integration points around the boundary for the final designs.

8.2.3. Penalization vs. design freedom

The penalization of intermediate densities in combination with a maximum slope control of the LSF appears to limit the design freedom of the shape optimization. The strain energy is mostly lower for final designs obtained with lower penalization factors, so those designs perform better. The steepest possible gradient of the LSF is only possible for a specific summation of RBFs. A high penalization pushes the design to a shape which allows this specific summation of RBFs such that the intermediate densities are minimized. The use of more quadtree refinements or a higher move limit parameter show to be a good alternative to obtain designs with less intermediate densities.

8.3. Recommendations

This section provides recommendations for further research and discusses potential improvements of the proposed post-processor.

8.3.1. Extension to 3D

Although the actual theory and methods behind the post-processor are extendible to 3D, the actual code for post-processing 3D problems should still be written. Also case studies in 3D could bring new insights or problems that did not appear for the 2D cases.

8.3.2. Quadtree integration band

On average more than 10% of the computation time of the shape optimization is spent on creating a quadtree integration band. Improving this process could make the post-processor significantly faster.

Instead of taking centers of integration cells as a boundary approximation, the derivative of the density with respect to the LSF in the integration points can be used. All integration points having a relatively high value for this derivative are part of the boundary approximation. The advantage is that these values are computed anyway. Furthermore, checking the sign change of the LSF at the corner points is not a solid proof whether an integration cell is intersected. Examples could be thought of for which the integration cell is intersected by the boundary but there is not a sign change of the LSF at the corner points. Checking values in the integration points does not experience this problem.

8.3.3. Convergence behaviour shape optimization

More research is needed in the behaviour of the post-processor for different values for the move limit parameter, the convergence tolerance and the maximum amount of iterations before a reinitialization is done.

A few test simulations show that the move limit parameter should be chosen smaller for stable convergence if the accuracy of the analysis is increased. So, a shape optimization using a polynomial order $P = 2$ and $q = 1$ quadtree refinements, has a stable convergence for higher move limit parameters than when P or q are increased.

Many shape optimizations converge in the second iteration of the last reinitialization cycle, so after 17 (5,10,2) iterations. Further research is needed to determine why this is. It could be that steps are generally smaller in the first iterations after a reinitialization because MMA is initializing during its first two optimization iterations. If the steps are small enough, the convergence tolerance will be reached right away. If this is the case, the so called 'asyinit' parameter can have a major role in this behaviour. This parameter influences the distance between the asymptotes in the first two iterations.

8.3.4. RAM usage

The program is constructed such that all evaluations needed at the level of the integration points are only done once, e.g. calculating the RBFs values and the stiffness matrices at the integration points. The advantage is the computational speed of the program because calculations are not done more than absolutely needed. A clear disadvantage is the high use of RAM because a lot of information needs to be stored. A more efficient handling or evaluation of the LSF in the integration points could be an improvement for the post-processor.

8.3.5. Minimum feature size

Slope control is introduced to make sure the integration points at the boundary capture a non-zero sensitivity. The slope control is a combination of penalization of intermediate densities and bounding the design variables s_i . This way the LSF is prevented from becoming respectively too flat and too steep. This behaviour actually prevents members from becoming too small as well. A summation of several RBFs leads to a maximum value of the LSF and therefore a maximum possible gradient. This also implies that a relatively small or thin member (i.e. roughly smaller than the span of one RBF), has LSF values close to zero. LSF values close to zero are associated to intermediate densities. The use of penalization makes sure it is not beneficial to have such a member.

In case minimum feature size constraint are to be implemented, research could be done on the precise effects of penalization on the feature size.

8.3.6. Stress constraints

Stresses can be evaluated when FCM is used by for example evaluating the stresses at the corner points of the quadtree integration cells, as can be seen in figure 8.1. This also means that stress constraints could be implemented as well for the shape optimization. A point of attention would be how to deal with stress in the void part of the domain. The stresses in the void part are high, so these stresses should be scaled somehow. It is very likely that the way the SIMP method deals with this problem can also be applied to the post-processor.

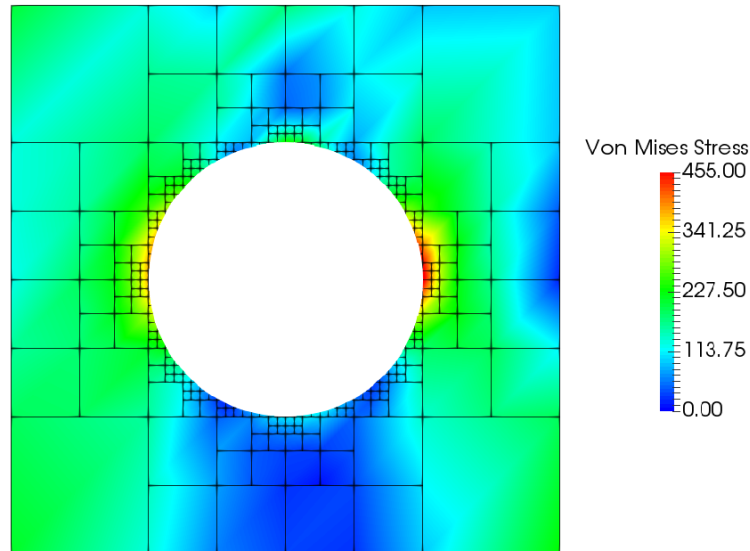
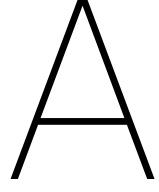


Figure 8.1: The Von Mises stress (MPa) in the rectangular plate with circular hole, a problem discussed in section 3.5.



Algorithms

This appendix shows pseudo-code of the main algorithms implemented in Python.

Algorithm 1 Geometry extraction - stage two

LINK DENSITIES TO LSF IN A MATRIX EQUATION

```
1: for each node  $i$  do
2:    $\rho_i$  = average density of elements using node  $i$ 
3:   for each RBF  $j$  do
4:      $A_{ij} = e^{-(r_j/B)^2}$  where  $r_j = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$  ▷ value RBF  $j$  at location node  $i$ 
5:   end for
6: end for
7: solve  $A_{ij} s_{j,\rho_i} = \rho_i$  for  $s_{j,\rho_i}$ 
```

SHIFT LSF ZERO-CONTOUR

```
8:  $\phi_{th,0} = 1 - V_{\text{frac}}$ 
9: while  $|\varepsilon| > 1 \cdot 10^{-4}$  and  $n < 6$  do
10:   compute area:  $V$ 
11:   compute error:  $\varepsilon = V_{\text{frac}} - V / (L_x \cdot L_y)$ 
12:   update threshold:  $\phi_{th,n} = \phi_{th,n-1} - \varepsilon$ 
13:    $n = n + 1$ 
14: end while
15: solve  $A_{ij} s_{j,th} = \phi_{th,n}$  for  $s_{j,th}$ 
16:  $s_j = s_{j,\rho_i} - s_{j,th}$ 
```

PROJECTING LSF TO DENSITIES

```
17: for each point in design domain  $p$  do
18:    $\rho_p = 0$ 
19:   for each RBF  $j$  do
20:      $\rho(p) = \rho(p) + e^{-(r_j/B)^2}$  where  $r_j = \sqrt{(x_p - x_j)^2 + (y_p - y_j)^2}$ 
21:   end for
22: end for
```

Algorithm 2 LSF slope control

COMPUTE MINIMUM VALUE LSF

- 1: maximum gradient LSF: $|\nabla\phi|_{\max} = 1.24637 \cdot 1.77264^2 \cdot (|s_i|_{\max}/B)$
- 2: distance integration points: $\Delta x = B/(2^q \cdot (P+1))$
- 3: lowest value LSF at integration points: $\phi_{\min} = \nabla\phi_{\max} \cdot (\Delta x/2)$

 COMPUTE κ ITERATIVELY

- 4: desired min derivative: $\partial\rho_{\min} = 0.5$
 - 5: initialize: $\kappa = 1000$
 - 6: **while** $\varepsilon > 1 \cdot 10^{-4}$ **do**
 - 7: compute derivative: $\partial\rho = \kappa e^{(\kappa\phi_{\min})} / (1 + e^{(\kappa\phi_{\min})})^2$
 - 8: compute difference: $\varepsilon = \partial\rho_{\min} - \partial\rho$
 - 9: update κ : $\kappa = \kappa - \varepsilon$
 - 10: **end while**
-

Algorithm 3 Creation quadtree integration band

DETERMINE IF CELL IS INTERSECTED

```

1: procedure INTERSECT(cell)
2:   for each cell corner  $c$  do
3:     evaluate LSF at  $(x_c, y_c)$ :  $\phi_c$ 
4:   end for
5:   if sign change  $\phi_c$  then                                ▷ if corner points have different sign LSF
6:     return True
7:   else
8:     return False
9:   end if
10: end procedure

```

CREATE BOUNDARY APPROXIMATION USING PREVIOUSLY OBTAINED CELLS

```

11: for each cell  $c$  do
12:   if INTERSECT(cell) then
13:      $cb_x = \text{average}(x_c)$                                 ▷ coordinates boundary are centers of intersected cells
14:      $cb_y = \text{average}(y_c)$ 
15:   end if
16: end for

```

DISTANCE ELEMENT TO APPROXIMATED BOUNDARY POINTS

```

17: procedure DISTANCE-CB(cell,cb)
18:   if  $\min\left(\sqrt{(x_c - cb_x)^2 + (y_c - cb_y)^2}\right) < B/2$  then    ▷ quadtree if element within half bandwidth distance
19:     return True
20:   else
21:     return False
22:   end if
23: end procedure

```

CREATE QUADTREE INTEGRATION BAND

```

24: for each element  $el$  do
25:   for each quadtree refinement  $q$  do
26:     for each (sub-)cell  $c$  do
27:       if DISTANCE-CB(cell,cb) then
28:         split cell( $c$ ) in 4 sub-cells
29:       end if
30:     end for
31:     update  $c$  to check new sub-cells for intersection
32:   end for
33:   LSF left bottom corner  $el$ :  $\phi_{el}$ 
34:   if  $\phi_{el} < 0$  and size(cell( $el$ )) == 1 then                                ▷ if element is void and not refined
35:     delete element  $el$ 
36:   end if
37: end for

```

B

Performance stage three

This appendix contains results of different case studies shown in chapter 6. The performance of stage three, the shape optimization, is tested by changing the polynomial order of the structural analysis, the number of quadtree refinements of the numerical integration, the penalization of intermediate densities or the move limit parameter of MMA.

MBB beam

Compared to the settings given in section 7.1 for stage three, the results of the MBB beam case for different properties of the final design are given in:

- table B.1 for different polynomial orders;
- table B.2 for different number of quadtree refinements;
- table B.3 for different penalization factors;
- table B.4 for different move limit parameters.

Cantilever beam

Compared to the settings given in section 7.1 for stage three, the results of the cantilever beam case for different properties of the final design are given in:

- table B.5 for different polynomial orders;
- table B.6 for different number of quadtree refinements;
- table B.7 for different penalization factors;
- table B.8 for different move limit parameters.

Bridge with distributed load

Compared to the settings given in section 7.1 for stage three, the results of the bridge with distributed load case for different properties of the final design are given in:

- table B.9 for different polynomial orders;
- table B.10 for different number of quadtree refinements;
- table B.11 for different penalization factors;
- table B.12 for different move limit parameters.

Table B.1: The results for different properties of the final design after stage three of the MBB beam for different polynomial orders P . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Polynomial order – MBB beam	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$P = 1$	11.4	114.7	17 (5,10,2)	5.3
$P = 2$	8.0	113.6	17 (5,10,2)	7.9
$P = 3$	6.1	112.7	19 (5,10,4)	14
$P = 4$	5.1	112.7	19 (5,10,4)	26
$P = 5$	5.0	16896	30 (5,10,15)	79

Table B.2: The results for different properties of the final design after stage three of the MBB beam for different number of quadtree refinements q . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Quadtree refinements – MBB beam	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$q = 0$	16.8	114.2	17 (5,10,2)	7.5
$q = 1$	8.0	113.6	17 (5,10,2)	7.9
$q = 2$	4.4	112.8	20 (5,10,5)	15
$q = 3$	2.3	112.8	20 (5,10,5)	37

Table B.3: The results for different properties of the final design after stage three of the MBB beam for different penalization factors p_{LSM} . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Penalization – MBB beam	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$p_{\text{LSM}} = 1$	29.9	139.6	30 (5,10,15)	12
$p_{\text{LSM}} = 2$	10.3	112.0	17 (5,10,2)	8.0
$p_{\text{LSM}} = 3$	8.0	113.6	17 (5,10,2)	7.9
$p_{\text{LSM}} = 4$	7.1	114.5	17 (5,10,2)	8.0
$p_{\text{LSM}} = 5$	6.5	115.4	19 (5,10,4)	8.2

Table B.4: The results for different properties of the final design after stage three of the MBB beam for different move limit parameters α_{LSM} . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Move limit – MBB beam	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$\alpha_{\text{LSM}} = 0.05$	10.0	113.9	17 (5,10,2)	8.1
$\alpha_{\text{LSM}} = 0.10$	8.0	113.6	17 (5,10,2)	7.9
$\alpha_{\text{LSM}} = 0.15$	6.8	113.4	19 (5,10,4)	8.1
$\alpha_{\text{LSM}} = 0.20$	6.3	113.1	21 (5,10,6)	8.8
$\alpha_{\text{LSM}} = 0.28$	5.4	113.0	30 (5,10,15)	11.4

Table B.5: The results for different properties of the final design after stage three of the cantilever beam for different polynomial orders P . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Polynomial order – Cantilever beam	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$P = 1$	10.4	101.0	17 (5,10,2)	14
$P = 2$	7.2	99.7	17 (5,10,2)	23
$P = 3$	5.6	99.3	17 (5,10,2)	42
$P = 4$	4.6	99.4	17 (5,10,2)	87
$P = 5$	3.9	99.1	17 (5,10,2)	169

Table B.6: The results for different properties of the final design after stage three of the cantilever beam for different number of quadtree refinements q . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Quadtree refinements – Cantilever beam	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$q = 0$	22.7	101.5	7 (5,2,-)	14
$q = 1$	7.2	99.7	17 (5,10,2)	23
$q = 2$	3.9	99.4	17 (5,10,2)	37
$q = 3$	2.1	98.9	17 (5,10,2)	93

Table B.7: The results for different properties of the final design after stage three of the cantilever beam for different penalization factors p_{LSM} . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Penalization – Cantilever beam	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$p_{\text{LSM}} = 1$	27.7	782	30 (5,10,15)	33
$p_{\text{LSM}} = 2$	9.0	98.7	17 (5,10,2)	23
$p_{\text{LSM}} = 3$	7.2	99.7	17 (5,10,2)	23
$p_{\text{LSM}} = 4$	6.4	100.5	17 (5,10,2)	24
$p_{\text{LSM}} = 5$	6.0	101.0	17 (5,10,2)	23

Table B.8: The results for different properties of the final design after stage three of the cantilever beam for different move limit parameters α_{LSM} . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Move limit – Cantilever beam	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$\alpha_{\text{LSM}} = 0.05$	11.0	102.3	7 (5,2,-)	12
$\alpha_{\text{LSM}} = 0.10$	7.2	99.7	17 (5,10,2)	23
$\alpha_{\text{LSM}} = 0.15$	6.3	99.4	18 (5,10,3)	23
$\alpha_{\text{LSM}} = 0.20$	5.9	99.2	18 (5,10,3)	24
$\alpha_{\text{LSM}} = 0.28$	4.9	99.2	30 (5,10,15)	22

Table B.9: The results for different properties of the final design after stage three of the bridge with distributed load for different polynomial orders P . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Polynomial order – Bridge	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$P = 1$	7.0	1.824	15 (5,10,-)	19
$P = 2$	4.6	1.818	17 (5,10,2)	35
$P = 3$	3.5	1.816	17 (5,10,2)	77
$P = 4$	2.8	1.816	17 (5,10,2)	167
$P = 5$	2.4	1.814	17 (5,10,2)	270

Table B.10: The results for different properties of the final design after stage three of the bridge with distributed load for different number of quadtree refinements q . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Quadtree refinements – Bridge	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$q = 0$	14.8	1.827	7 (5,2,-)	28
$q = 1$	4.6	1.818	17 (5,10,2)	35
$q = 2$	2.6	1.820	13 (5,8,-)	40
$q = 3$	1.1	1.822	19 (5,10,4)	125

Table B.11: The results for different properties of the final design after stage three of the bridge with distributed load for different penalization factors p_{LSM} . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Penalization – Bridge	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$p_{\text{LSM}} = 1$	23.5	160713	30 (5,10,15)	51
$p_{\text{LSM}} = 2$	5.7	1.813	17 (5,10,2)	36
$p_{\text{LSM}} = 3$	4.6	1.818	17 (5,10,2)	35
$p_{\text{LSM}} = 4$	4.1	1.821	17 (5,10,2)	36
$p_{\text{LSM}} = 5$	3.8	1.826	18 (5,10,3)	36

Table B.12: The results for different properties of the final design after stage three of the bridge with distributed load for different move limit parameters α_{LSM} . The iterations are given as a total and between the brackets the number of iterations before and after the reinitialization.

Move limit – Bridge	Intermediate densities (%)	Strain energy	Iterations	Computation time (s)
$\alpha_{\text{LSM}} = 0.05$	5.6	1.822	17 (5,10,2)	36
$\alpha_{\text{LSM}} = 0.10$	4.6	1.818	17 (5,10,2)	35
$\alpha_{\text{LSM}} = 0.15$	3.9	1.817	18 (5,10,3)	36
$\alpha_{\text{LSM}} = 0.20$	3.6	1.816	17 (5,10,2)	37
$\alpha_{\text{LSM}} = 0.40$	3.0	1.813	27 (5,10,7)	42

Bibliography

- [1] A. Abedian, J. Parvizian, A. Düster, H. Khademyzadeh, and E. Rank. Performance of different integration schemes in facing discontinuities in the finite cell method. *International Journal of Computational Methods*, 10(3):1–24, 2013. doi: 10.1142/S0219876213500023.
- [2] G. Allaire, F. Jouve, and A.M. Toader. Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1):363–393, 2004. doi: 10.1016/j.jcp.2003.09.032.
- [3] M. P. Bendsøe. Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4): 193–202, 1989. doi: 10.1007/BF01650949.
- [4] M. P. Bendsøe and N. Kikuchi. Generating optimal topologies in structural design using a homogenisation method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197–224, 1988.
- [5] M. P. Bendsøe and O. Sigmund. *Topology optimization: theory, methods and applications*. Springer, Berlin, 2003.
- [6] E. Cervera and J. Trevelyan. Evolutionary structural optimisation based on boundary representation of NURBS. Part II: 3D algorithms. *Computers and Structures*, 83(23):1917–1929, 2005. doi: 10.1016/j.compstruc.2005.02.017.
- [7] S. Cho and S. H. Ha. Isogeometric shape design optimization: Exact geometry and enhanced sensitivity. *Structural and Multidisciplinary Optimization*, 38(1):53–70, 2009. doi: 10.1007/s00158-008-0266-z.
- [8] M. Dauge, A. Düster, and E. Rank. Theoretical and Numerical Investigation of the Finite Cell Method. *Journal of Scientific Computing*, 65(3):1039–1064, 2015. doi: 10.1007/s10915-015-9997-3.
- [9] J. D. Deaton and R. V. Grandhi. A survey of structural and multidisciplinary continuum topology optimization: Post 2000. *Structural and Multidisciplinary Optimization*, 49(1):1–38, 2014. doi: 10.1007/s00158-013-0956-z.
- [10] T. P. Fries and T. Belytschko. The extended/generalized finite element method: An overview of the method and its applications. *International Journal for Numerical Methods in Engineering*, 84(3):253–304, 2010. doi: 10.1002/nme.
- [11] M. J. Garcia, M. A. Henao, and O. E. Ruiz. Fixed grid finite element analysis for 3D structural problems. *International Journal for Computational Methods*, 2(4):569–586, 2005.
- [12] J. K. Guest, J. H. Prévost, and T. Belytschko. Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering*, 61:238–254, 2004. doi: 10.1002/nme.1064.
- [13] M. H. Hsu and Y. L. Hsu. Interpreting three-dimensional structural topology optimization results. *Computers and Structures*, 83(4):327–337, 2005. doi: 10.1016/j.compstruc.2004.09.005.
- [14] Y. L. Hsu, M. S. Hsu, and C. T. Chen. Interpreting results from topology optimization using density contours. *Computers & Structures*, 79(10):1049–1058, 2001. doi: 10.1016/S0045-7949(00)00194-2.
- [15] T. J. R. Hughes, J. A. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39): 4135–4195, 2005. doi: 10.1016/j.cma.2004.10.008.
- [16] M. Joulaiian, S. Hubrich, and A. Düster. Numerical integration of discontinuities on arbitrary domains based on moment fitting. *Computational Mechanics*, 57(6):979–999, 2016. doi: 10.1007/s00466-016-1273-3.

- [17] P. Kumar and A. Saxena. On topology optimization with embedded boundary resolution and smoothing. *Structural and Multidisciplinary Optimization*, 52(6):1135–1159, 2015. doi: 10.1007/s00158-015-1272-6.
- [18] C. Le, T. Bruns, and D. Tortorelli. A gradient-based, parameter-free approach to shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 200(9):985–996, 2011. doi: 10.1016/j.cma.2010.10.004.
- [19] L. Li, M. Y. Wang, and P. Wei. XFEM schemes for level set based structural optimization. *Frontiers of Mechanical Engineering*, 7(4):335–356, 2012. doi: 10.1007/s11465-012-0351-2.
- [20] C. Y. Lin and L. S. Chao. Automated image interpretation for integrated topology and shape optimization. *Structural and Multidisciplinary Optimization*, 20(2):125–137, 2000. doi: 10.1007/s001580050144.
- [21] A. Mandhyan, G. Srivastava, and S. Krishnamoorthi. A novel method for prediction of truss geometry from topology optimization. *Engineering with Computers*, 33(1):95–106, 2017. doi: 10.1007/s00366-016-0474-x.
- [22] A. L. Marsan and D. Dutta. Construction of a surface model and layered manufacturing data from 3D homogenization output. *Journal of Mechanical Design*, 118:412–418, 1996.
- [23] K. Maute and E. Ramm. Adaptive topology optimization. *Structural and Multidisciplinary Optimization*, 10(2):100–112, 1995. doi: 10.1007/BF01743537.
- [24] B. Müller, F. Kummer, and M. Oberlack. Highly accurate surface and volume integration on implicit domains by means of moment-fitting. *International Journal for Numerical Methods in Engineering*, 96(8):512–528, 2013. doi: 10.1002/nme.4569.
- [25] P. Y. Papalambros and M. Chirehdast. An integrated environment for structural configuration design, 1990.
- [26] J. Parvizian, A. Düster, and E. Rank. Finite cell method. *Computational Mechanics*, 41(1):121–133, 2007. doi: 10.1007/s00466-007-0173-y.
- [27] J. Parvizian, A. Düster, and E. Rank. Topology optimization using the finite cell method. *Optimization and Engineering*, 13(1):57–78, 2012. doi: 10.1007/s11081-011-9159-x.
- [28] Y. D. Seo, H. J. Kim, and S. K. Youn. Shape optimization and its extension to topological design based on isogeometric analysis. *International Journal of Solids and Structures*, 47(11):1618–1640, 2010. doi: 10.1016/j.ijsolstr.2010.03.004.
- [29] J. A. Sethian and A. Wiegmann. Structural boundary design via level set and immersed interface methods. *Journal of Computational Physics*, 163(2):489–528, 2000. doi: 10.1006/jcph.2000.6581.
- [30] A. Sharma, H. Villanueva, and K. Maute. On shape sensitivities with heaviside-enriched XFEM. *Structural and Multidisciplinary Optimization*, 55(2):385–408, 2017. doi: 10.1007/s00158-016-1640-x.
- [31] O. Sigmund. A 99 line topology optimization code written in matlab. *Structural and Multidisciplinary Optimization*, 21(2):120–127, 2001. doi: 10.1007/s001580050176.
- [32] O. Sigmund. Morphology-based black and white filters for topology optimization. *Structural and Multidisciplinary Optimization*, 33(4):401–424, 2007. doi: 10.1007/s00158-006-0087-x.
- [33] M. Stolpe and K. Svanberg. On the trajectories of penalization methods for topology optimization. *Structural and Multidisciplinary Optimization*, 21(2):128–139, 2001. doi: 10.1007/s001580050177.
- [34] K. Svanberg. The method of moving asymptotes - a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373, 1987.
- [35] B. Szabo and I. Babuska. *Finite Element Analysis*. John Wiley & Sons, 1991.
- [36] B. Szabó, A. Düster, and E. Rank. The p-version of the finite element method. *Encyclopedia of computational mechanics*, 2004.

- [37] P. S. Tang and K. H. Chang. Integration of topology and shape optimization for design of structural components. *Structural and Multidisciplinary Optimization*, 22(1):65–82, 2001. doi: 10.1007/PL00013282.
- [38] N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48(3):437–472, 2013. doi: 10.1007/s00158-013-0912-y.
- [39] F. van Keulen, R. T. Haftka, and N. H. Kim. Review of options for structural design sensitivity analysis. Part 1: Linear systems. *Computer Methods in Applied Mechanics and Engineering*, 194(30):3213–3243, 2005. doi: 10.1016/j.cma.2005.02.002.
- [40] G. Ventura. On the elimination of quadrature subcells for discontinuous functions in the eXtended Finite-Element Method. *International Journal for Numerical Methods in Engineering*, 66(5):761–795, 2006. doi: 10.1002/nme.1570.
- [41] W. A. Wall, M. A. Frenzel, and C. Cyron. Isogeometric structural shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 197(33):2976–2988, 2008. doi: 10.1016/j.cma.2008.01.025.
- [42] M. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1):227–246, 2003.
- [43] P. Wei, M. Y. Wang, and X. Xing. A study on X-FEM in continuum structural optimization using a level set model. *Computer-Aided Design*, 42(8):708–719, 2010. doi: 10.1016/j.cad.2009.12.001.
- [44] G. Yi and N. H. Kim. Identifying boundaries of topology optimization results using basic parametric features. *Structural and Multidisciplinary Optimization*, 55(5):1641–1654, 2017. doi: 10.1007/s00158-016-1597-9.
- [45] M. Zhou and G. I. N. Rozvany. The COC algorithm, Part II: Topological, geometrical and generalized shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 89(1-3):309–336, 1991. doi: 10.1016/0045-7825(91)90046-9.