Combined Shape and Topology Optimization

Christiansen, Asger Nyman; Bærentzen, Jakob Andreas; Sigmund, Ole

Publication date:
2015

Document Version
Publisher's PDF, also known as Version of record

Citation (APA):
Christiansen, A. N., Bærentzen, J. A., & Sigmund, O. (2015). Combined Shape and Topology Optimization. Kgs. Lyngby: Technical University of Denmark (DTU). (DTU Compute PHD-2014; No. 350).

Combined
Shape and Topology
Optimization

Asger Nyman Christiansen

Kongens Lyngby 2014
PhD-2014-350

Technical University of Denmark
Department of Applied Mathematics and Computer Science
Richard Petersens Plads, building 324,
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk PhD-2014-350

Summary (English)

Shape and topology optimization seeks to compute the optimal shape and topology of a structure such that one or more properties, for example stiffness, balance or volume, are improved. The goal of the thesis is to develop a method for shape and topology optimization which uses the Deformable Simplicial Complex (DSC) method. Consequently, we present a novel method which combines current shape and topology optimization methods. This method represents the surface of the structure explicitly and discretizes the structure into non-overlapping elements, i.e. a simplicial complex. An explicit surface representation usually limits the optimization to minor shape changes. However, the DSC method uses a single explicit representation and still allows for large shape and topology changes. It does so by constantly applying a set of mesh operations during deformations of the structure. Using an explicit instead of an implicit representation gives rise to several advantages including straightforward modeling of the surface, improved scalability and ability to optimize multiple materials.

This dissertation describes the essential parts of the novel method for combined shape and topology optimization. This includes the structural analysis in Chapter 2, the optimization in Chapter 3 and the Deformable Simplicial Complex method in Chapter 4. Finally, four applications of the developed method are presented in the included papers and summarized in Chapter 5.

Summary (Danish)

Form- og topologioptimering forsøger at beregne den optimale form og topologi af en struktur så en eller flere egenskaber, for eksempel stivhed, balance eller volumen, bliver forbedret. Målet med afhandlingen er at udvikle en metode til form- og topologioptimering som anvender Deformable Simplicial Complex (DSC) metoden. Denne afhandling præsenterer derfor en ny metode der kombinerer tidligere form- og topologioptimerings metoder. Metoden repræsenterer overfladen af strukturen eksplicit og diskretiserer strukturen i elementer der ikke overlapper. En eksplicit repræsentation af overfladen begrænser normalt optimeringen til små formændringer men DSC metoden, der andvender en enkelt eksplicit repræsentation, giver mulighed for store formændringer og endda topologiændringer. Den opnår dette ved konstant at anvende et sæt af mesh operationer samtidigt med at strukturen bliver deformeret. En eksplicit i stedet for en implicit repræsentation giver anledning til en række fordele, f.eks. er det ligetil at modellere overfladen, metoden skalerer bedre og optimering med flere materialer er muligt.

Afhandlingen beskriver de væsentlige dele af den nye metode til kombineret form- og topologioptimering. Dette omfatter den strukturelle analyse i kapitel 2, optimeringen i kapitel 3 og Deformable Simplicial Complex metoden i kapitel 4. Derudover er fire anvendelser af denne nye metode præsenteret i de medfølgende artikler og sammenfattet i kapitel 5.

Preface

This thesis was prepared at the Department of Applied Mathematics and Computer Science at the Technical University of Denmark (DTU) in partial fulfillment of the requirements for acquiring the Ph.D. degree in engineering.

The thesis consists of a summary report and a collection of three published scientific papers and one paper currently under review. The work was carried out between 2011 and 2014.

Lyngby, 01-November-2014

A. Ch

Asger Nyman Christiansen

List of publications

Included papers

Paper A Christiansen, A. N., Nobel-Jørgensen, M., Aage, N., Sigmund, O., and Bærentzen, J. A. (2014b). Topology optimization using an explicit interface representation. *Structural and Multidisciplinary Optimization*, 49(3):387–399

Paper B Christiansen, A. N., Bærentzen, J. A., Nobel-Jørgensen, M., Aage, N., and Sigmund, O. (2014a). Combined shape and topology optimization of 3D structures. *Computers & Graphics*, 46(2015):25 – 35. Shape Modeling International 2014

Paper C Christiansen, A. N., Tortorelli, D. A., Aage, N., and Sigmund, O. (2014d). Combined shape and topology optimization for minimization of maximal von Mises stress. (Submitted)

Paper D Christiansen, A. N., Schmidt, R., and Bærentzen, J. A. (2014c). Automatic balancing of 3D models. *Computer-Aided Design*, 58(2015):236 – 241. Solid and Physical Modeling 2014

Additional publications

- Christiansen, A. N., Carstensen, J. M., Møller, F., and Nielsen, A. A. (2012). Monitoring the change in colour of meat: A comparison of traditional and kernel-based orthogonal transformations. *Journal of Spectral Imaging*, 3(1)

- Christiansen, A. N. and Bærentzen, J. A. (2013). Generic graph grammar: A simple grammar for generic procedural modelling. In *Proceedings of the 28th Spring Conference on Computer Graphics*, SCCG '12, pages 85–92, New York, NY, USA. ACM

- Christiansen, A. N., Nobel-Jørgensen, M., Bærentzen, J. A., Aage, N., and Sigmund, O. (2013). Topology optimization using an explicit interface representation. Presented at the 10th World Congress on Structural and Multidisciplinary Optimization (WCSMO-10)

- Nobel-Jørgensen, M., Christiansen, A. N., Bærentzen, J. A., Aage, N., and Sigmund, O. (2013). Improving topology optimization using games. Presented at the 10th World Congress on Structural and Multidisciplinary Optimization (WCSMO-10)

- Dahl, V. A., Christiansen, A. N., and Bærentzen, J. A. (2014). Multiphase image segmentation using the deformable simplicial complex method. In *Proceedings of the 22nd International Conference on Pattern Recognition*

- Christiansen, A. N., Tortorelli, D. A., Aage, N., and Sigmund, O. (2014e). Combined shape and topology optimization for minimization of von Mises stress. Presented at the 4th International Conference on Engineering Optimization (EngOpt 2014)

- Nobel-Jørgensen, M., Aage, N., Christiansen, A. N., Igarashi, T., Bærentzen, J. A., and Sigmund, O. (2014). 3D interactive topology optimization on hand-held devices. (Submitted)

Acknowledgments

First and foremost, I would like to thank my two main supervisors, Andreas Bærentzen and Ole Sigmund, for their knowledge and technical advise. Good ideas come only from the brightest minds and both Andreas and Ole have plenty. Furthermore, I would like to express my gratitude to my co-supervisor Niels Aage for patiently teaching me everything I needed to know about topology optimization. I am also thankful for the collaboration and discussions with my co-authors Morten Nobel-Jørgensen, Vedrana A. Dahl, Daniel A. Tortorelli, Ryan Schmidt, Flemming Møller, Jens Michael Carstensen and Allan A. Nielsen.

I am grateful for the financial support from the Villum foundation who made this project possible through the grant 'NextTop'. I would also like to express my gratitude to the Otto Mønsted foundation for supporting my study financially on several occasions. Furthermore, I am grateful to the Technical University of Denmark for the financial support of the project and for all the damn good coffee.

I would like to thank my colleagues, the members of the Image Analysis and Computer Graphics group, for all the interesting discussions and fun times. A special thanks goes to Morten for being the best colleague I could have wished for.

Finally, I am deeply grateful for the support that my friends and family have given me though good times and bad times and I would especially like to thank my girlfriend Annette.

So long, and thanks for all the fish.

List of Symbols

DSC	Deformable Simplicial Complex
FEM	Finite Element Method
FEA	Finite Element Analysis
SED	Strain energy density
CST	Constant strain triangles
LST	Linear strain triangles
f	Objective function
g	Global constraint function
n	Design node
c	Control point
e	Element (triangle in 2D and tetrahedron in 3D)
$\boldsymbol{p}_n$	Position of node n
$\boldsymbol{p}$	Positions of all nodes assembled in a vector $\boldsymbol{p} = [\dots, \boldsymbol{p}_n^T, \dots]^T$
m_e	Material of element e
$\boldsymbol{m}$	Materials of all elements assembled in a vector $\boldsymbol{m} = [\dots, m_e, \dots]^T$
$\boldsymbol{f}_c$	Local load vector of control point c
$\boldsymbol{f}$	Global load vector
$\boldsymbol{K}_e$	Local stiffness matrix of element e
$\boldsymbol{K}$	Global stiffness matrix

$\boldsymbol{E}_e$	Constitutive matrix of element e
$\boldsymbol{B}_e$	Strain-displacement matrix of element e
$\boldsymbol{u}_c$	Local displacement vector of control point c
$\boldsymbol{u}$	Global displacement vector
$\underset{e}{\mathbb{A}}$	Finite element assembly operator
s_e	Strain energy density (SED) of element e
$\boldsymbol{\epsilon}_e$	Strain of element e
$\boldsymbol{\sigma}_e$	Stress of element e
σ_e	Von Mises stress of element e

Note that the mathematical notation may deviate in the included papers.

Contents

Chapter 1

Introduction

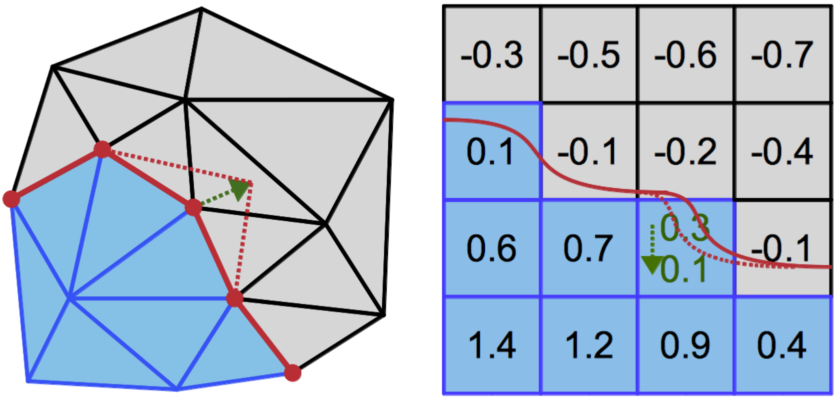
This dissertation introduces a novel method for shape and topology optimization.

This chapter first introduces the notion of shape and topology optimization along with examples of its use. Then, the general ideas of the proposed method for combined shape and topology optimization are explained and motivated. Finally, previous methods are described briefly before an overview of the thesis is presented along with the contributions, i.e. the advantages of the presented method.

1.1 Motivation

Shape and topology optimization is the discipline of computing the optimal shape and topology of a structure with respect to some desired effect (Bendsøe and Kikuchi, 1988; Bendsøe and Sigmund, 2003). It can be used to solve a wide variety of design problems arising when producing such diverse products as cars, houses, computer chips and antennas. The manufacturers are often concerned with finding the stiffest structure, the lightest structure which does not break, the structure with the highest cooling effect, etc.

Existing methods include approaches to optimize the shape of a structure where its surface is represented explicitly. However, these methods cannot accommodate large shape deformations and topology changes. Consequently, a variety of topology optimization methods exists e.g. the density method (Bendsøe, 1989;



-0.3	-0.5	-0.6	-0.7
0.1	-0.1	-0.2	-0.4
0.6	0.7	0.3 0.1	-0.1
1.4	1.2	0.9	0.4

(a) Explicit representation

(b) Implicit representation

Figure 1.1: The explicit representation represents the object surface by a piecewise linear curve (red) and the shape is deformed by changing the positions of the surface nodes. Consequently, there is a direct connection between changes to the representation and the shape of the object. The implicit representation represents the surface by the zero level set (red) of a continuous function defined on a fixed grid. When the values of the function are changed, the surface is deformed implicitly. Note that the explicitly represented surface divides the internal elements (blue) from the external elements (gray), whereas the implicitly represented surface does not.

Bendsøe and Sigmund, 2003) and level set method (Wang et al., 2003; Allaire et al., 2004). Topology optimization methods can change the topology in addition to the shape but they represent the surface of the structure implicitly. Figure 1.1 illustrates the difference between implicit and explicit representations. The goal of this work is to combine the shape and topology optimization approaches into one unified framework which represents the surface explicitly and is able to accommodate large shape and topology changes.

Methods which only accommodate minor shape changes have the disadvantage that the optimization has to start close to the optimal structure to be able to reach optimum. On the other hand, if large shape and topology changes are possible, the optimization can start far away from optimum and still reach it. Consequently, the result is less dependent on the initial shape. However, one should note that also topology optimization approaches can end up in a local minimum.

An explicit representation of the surface is in many cases essential to solve shape and topology optimization problems. For example, the problem of optimizing the shape of an antenna is difficult using an implicit representation since the efficiency of the antenna depends on its smoothness (Aage et al., 2011). Another example is when modeling pressure, heat, fluid or anything where energy is transferred through the surface. In these cases, it is necessary to model the surface which is ambiguous and difficult to do when using an implicit representation. Furthermore, an explicit representation is always necessary to retrieve the final design. Consequently, methods using an implicit representation need to convert the implicit to an explicit representation, which results in loss of precision. Some approaches use both an implicit and explicit representation, however, again, that results in loss of precision when converting. Furthermore, it is inconvenient to switch between two representations and it increases the complexity of the methods.

This thesis will show that by using the Deformable Simplicial Complex (DSC) method (Misztal et al., 2010a; Misztal and Bærentzen, 2012), it is possible to combine shape and topology optimization. In other words, it is possible to represent the surface explicitly and accommodate large shape and topology changes during the optimization. An example of such an optimization is seen in Figure 1.2. In addition, we will show that this approach has several advantages which are direct consequences of the explicit representation. These advantages will be elaborated in Section 1.3 together with an overview of the thesis content. However, first, a brief overview of previous shape and topology optimization approaches are presented.

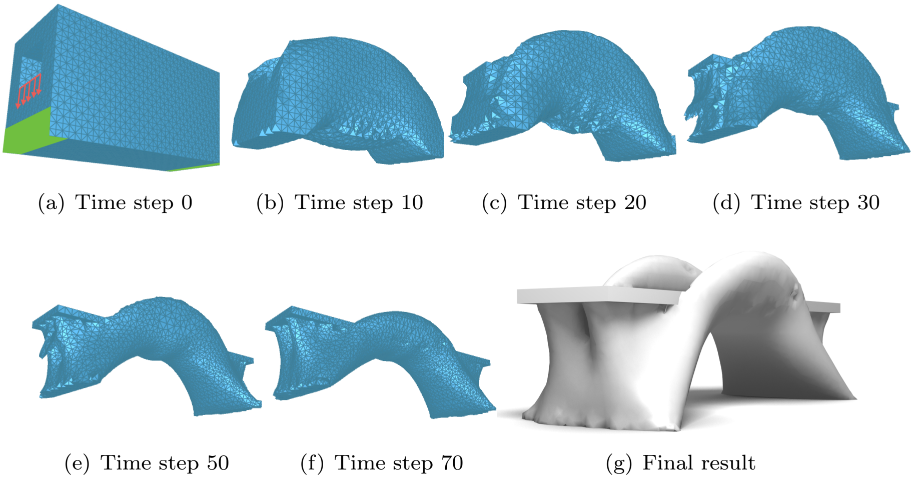

(a) Time step 0
(b) Time step 10
(c) Time step 20
(d) Time step 30
(e) Time step 50
(f) Time step 70
(g) Final result

Figure 1.2: Combined shape and topology optimization of a problem which is to connect the two river banks by a structure that is as stiff as possible while limiting the amount of material. The optimization is initialized as depicted in Figure 1.2(a). Furthermore, a few iterations of the method are depicted along with the result which took 68 min to compute on a laptop.

1.2 Previous work

Currently, three types of methods are used to solve shape and topology optimization problems: Eulerian methods, Lagrangian methods and hybrids between these. The main difference between these three approaches is the shape representation. For a more elaborate comparison of shape and topology optimization approaches see e.g. Sigmund and Maute (2013).

Eulerian methods

Eulerian methods are primarily represented by the density method (Bendsøe, 1989; Bendsøe and Sigmund, 2003) and the level set method (Osher and Fedkiw, 2002; Wang et al., 2003; Allaire et al., 2004). For the density method, the design variables are the density (the amount of material) of each element in a fixed grid. This approach implicitly represents the surface between material and void by the boundary between elements with high and low material density. For the level set method, the design variables are the values of a level set function defined on the nodes of a fixed grid. The surface is then represented by the zero level set of the level set function as illustrated in Figure 1.1. The Eulerian methods smoothly capture topology changes and, consequently, these methods are widely used for topology optimization.

Lagrangian methods

Lagrangian methods parametrize the surface explicitly and apply the optimization directly to the surface (Ding, 1986; Mohammadi and Pironneau, 2001; Le et al., 2011; Arnout et al., 2012). This means that the design variables are the positions of the nodes of the surface. It is therefore easy to track the surface with these methods but difficult to change the topology. Consequently, they have been used for shape optimization and often in combination with a parametrisation for example splines (Zhang et al., 1995). Furthermore, some have used Lagrangian methods for topology optimization by applying an expensive remeshing scheme (Eschenauer et al., 1994; Maute and Ramm, 1995; Kim et al., 2008). We also propose to use a Lagrangian method for topology optimization but without the expensive remeshing and we will therefore learn from previous approaches. For example, we will apply the theory of topological derivatives (Sokolowski and Zochowski, 1999; Céa et al., 2000; Feijóo et al., 2003) as also used by Eschenauer et al. (1994).

Hybrid methods

Notice, that an advantage of a Lagrangian method is a disadvantage of an Eulerian method and vice versa. Several research teams have therefore tried to combine these methods to use the appropriate method where suited. This has resulted in some complex, but useful, hybrid methods for example the particle level-set method (Enright et al., 2002) and the split-and-merge method (Wojtan et al., 2009). These ideas has also been utilised within the field of topology optimization. For example by combining a deformable mesh with the level set method (Chen et al., 2007; Ha and Cho, 2008; Allaire et al., 2011; Yamasaki et al., 2011; Xia et al., 2012; Allaire et al., 2013b) or in the finite point method (Oñate et al., 1999).

1.3 Contributions and thesis overview

The subject of the next chapter, Chapter 2, is the structural analysis, i.e. how the properties of a structure (stress, strain, volume, etc.) are computed for example by using the finite element method (FEM). To compute these properties, the structure has to be discretized into finite-sized elements. Here, the inside and outside of the structure are discretized into a simplicial complex, i.e. a triangular mesh in 2D and a tetrahedral mesh in 3D. Furthermore, each element in the simplicial complex is labeled with a material, e.g. *void* or *solid*. The surface of the structure is then represented by the line pieces (in 2D) or triangles (in 3D) which are sandwiched between two elements labeled *void* and *solid* respectively. Consequently, the surface of the structure is represented explicitly and, furthermore, embedded in the simplicial complex. Therefore, only one representation is used for both representing the shape and for discretizing the structure.

The FEAs are computationally very heavy for evaluating mechanical properties of the structure such as stiffness or stress. Therefore, using as few elements as possible to accurately represent the structure is essential to achieve an efficient method. A big advantage of the presented approach is therefore that the elements are labeled either *void* or *solid* as opposed to Eulerian methods. Consequently, void elements can be eliminated from the analysis and hence reduce the computation time drastically.

Chapter 3 will describe how the structure is iteratively optimized based on the structural analysis described in Chapter 2. We will use a novel gradient-based optimization scheme which alternates between a discrete and a continuous optimization step. The discrete optimization step relabels elements for example from *solid* to *void*. It is computationally very heavy to find the optimal solution to this discrete optimization. Consequently, the relabeling strategy is heuristic even though it is based on the theory of topological derivatives (Eschenauer et al., 1994; Sokolowski and Zochowski, 1999; Céa et al., 2000; Feijóo et al., 2003). There is therefore no guarantee to find an optimal solution in this step. However, the discrete step is followed by a continuous optimization step which performs a standard parameter-free shape optimization (Ding, 1986; Mohammadi and Pironneau, 2001; Le et al., 2011; Arnout et al., 2012) using the gradient-based optimization algorithm Method of Moving Asymptotes (MMA) (Svanberg, 1987). These two steps are then iterated until convergence.

The two optimization steps complement each other well. First, the continuous optimization step is guaranteed to converge to an optimized solution and, since the discrete step has less effect as the optimization progresses, the proposed optimization procedure converges. Furthermore, the discrete optimization step is global in the sense that it can change both the shape and topology anywhere on the structure. Finally, it is essential to be able to create holes inside the structure. The continuous optimization step, on the other hand, is local and can only deform the shape of the structure slightly in each step. However, if two surfaces collide, this step will merge the two surfaces and it can therefore close holes and collapse thin beams. In summary, the discrete step makes the structure converge fast to an approximate solution, whereas the continuous step deforms the structure slowly to an accurate result.

Chapter 4 describes the Deformable Simplicial Complex (DSC) method (Misztal et al., 2010a; Misztal and Bærentzen, 2012) and the strategy for deforming the structure based on the results of the optimization schemes from Chapter 3. The DSC method is able to deform a structure by moving the nodes on the surface and relabeling elements. While the surface is deformed, a set of mesh operations are performed to ensure that the elements in the simplicial complex do not degenerate. If the mesh contains degenerate or close to degenerate elements the FEA would produce large errors and the analysis would no longer be valid. In addition, the DSC method can change the topology using the same mesh operations. If two surfaces collide, the method automatically merge the structures at the point of impact. Finally, the DSC method can control the detail level of the simplicial complex, i.e. how fine or coarse different parts of the structure is discretized. The mesh operations that ensure high-quality elements, change the topology and control the level of detail are only applied where necessary which is often near the surface. Consequently, the DSC method is much faster than a complete remeshing.

The mesh adaptivity inherent to the DSC method implies that large elements can be used far from the surface to coarsely discretize the inside of the object thereby reducing the number of elements in the FEA. Oppositely, small elements can be used near the surface to get a finely discretized surface, thereby resulting in an optimized structure with fine details. The mesh adaptivity can also be utilized by increasing the resolution as the optimization converges. We show that this can be used with great success in 3D to speed up the optimization.

A final advantage of the DSC method is its capability to represent multiple materials in a simple manner as opposed to e.g. the level set method (Allaire et al., 2013a). Instead of labeling each element with *void* or *solid*, the set of labels could be extended to include *steel*, *concrete*, *plastic* etc. In this case, the DSC method automatically creates surfaces between elements with different labels. Furthermore, if two objects with different materials collide, the objects are not merged but kept separate by a surface. We will only take advantage of this in a few applications, however, it is an important feature of DSC which should be further utilized in the future.

Chapter 5 briefly summarizes four applications of the proposed method. For more details, the reader is referred to the four included papers in the appendix. The first two applications minimizes compliance of 2D (Paper A) and 3D (Paper B) structures respectively. Next, the maximal von Mises stress of 2D structures is minimized (Paper C). The final application automatically balances 3D models using a variant of the proposed scheme (Paper D).

Finally, the thesis is concluded in Chapter 6. Here, the advantages and disadvantages of the combined shape and topology optimization approach are summarized. Furthermore, an outlook is provided.

Chapter 2

Structural analysis

We want to change the shape and topology of a structure such that one of its properties, e.g. strain, stress, center of gravity or volume, is optimized. Hence, we need to both represent the shape of the structure and evaluate its structural properties. To do this, we discretize the entire design domain, not just the structure, into a simplicial complex. The shape of the structure is explicitly embedded in the simplicial complex whereas its solid elements are used for the structural analysis, i.e. to evaluate its structural properties. In addition, we will use the finite element method (FEM) to estimate some of the structural properties, e.g. stress and strain.

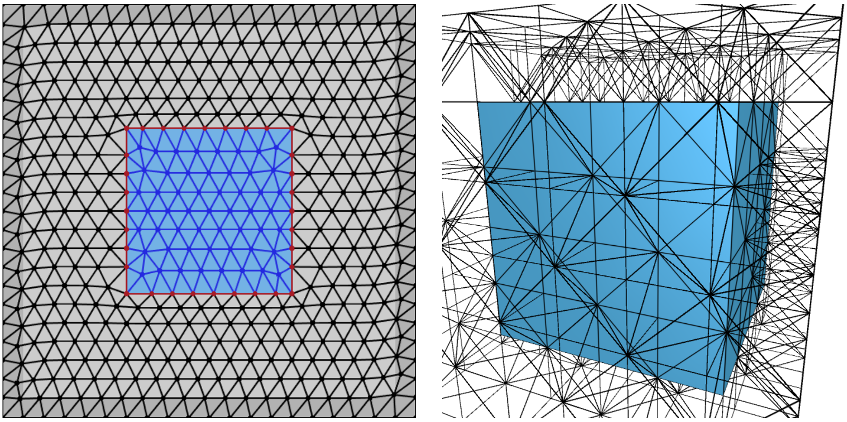
In this chapter, the discretization (the simplicial complex), the problem formulation and the finite element analysis (FEA) are described. Furthermore, a few examples of structural properties are presented.

2.1 Discretization and problem formulation

A simplicial complex discretizes a domain into *triangular elements* in 2D and *tetrahedral elements* in 3D[1] as illustrated in Figure 2.1. Furthermore, the elements do not overlap and any point in the discretized domain is either inside an element or on the boundary between elements. Finally, all elements e are labeled with a material m_e (for example air, plastic or steel) with associated material parameters *density* ρ_e, *Young's modulus* E_e and *Poisson's ratio* ν_e. For simplicity, we will illustrate the concept with only two labels; *void* and *solid*.

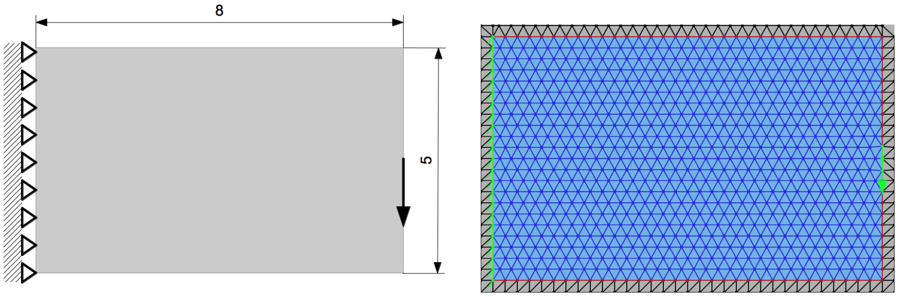
The structural surface is represented explicitly by the collection of line pieces in 2D and triangles in 3D that are sandwiched between two elements labeled with different materials. Furthermore, the shape and topology of the structure can be described by the positions $\boldsymbol{p} = [\ldots, \boldsymbol{p}_n^T, \ldots]^T$ of the *nodes* n and the material labels $\boldsymbol{m} = [\ldots, m_e, \ldots]^T$ of the *elements* e. In addition to the shape representation, we will use the simplicial complex discretization to evaluate the structural properties.

[1] In 3D, the tetrahedral mesh generator TetGen (Si, 2013) generates the initial mesh.


(a) 2D

(b) 3D

Figure 2.1: A square/cube discretized by a simplicial complex in 2D/3D. The design domain is completely filled with non-overlapping triangular/tetrahedral elements. The surface of the structure is then represented by the line pieces/triangles sandwiched between an element labeled *void* and an element labeled *solid*.



(a)

(b)

Figure 2.2: The cantilever beam problem and the discretization of this problem into a simplicial complex. On the left, the arrow indicates a load, the triangles indicates supports and the gray rectangle is the design domain. On the right, the design domain is completely filled with turquoise triangular elements and the structural surface is depicted in red. Furthermore, the bright green nodes with a slanted line are supported nodes whereas the bright green arrows represent loads applied to the structure.

Figure 2.2 illustrates how to model and discretize a structural optimization problem exemplified by the cantilever beam problem. First, the *design domain* specifies the space in which we want to find the optimal structure. Next, the *loads* specify the magnitude, direction and position of external forces applied to the surface of the structure. Finally, the *supports* can be interpreted as the structure being attached with infinitely strong glue to for example a wall or the ground.

2.2 Finite element analysis

Structural properties such as stress and strain are continuous functions defined at every point of the structure. Consequently, we estimate these properties using the FEM (Cook et al., 2007) as opposed to e.g. volume and center of gravity. We consider structures in static equilibrium exerted to a *force field*. Solving dynamic problems is therefore subject for future work. For the structure to be in static equilibrium, the acceleration exerted by the force field has to be countered by an equal and oppositely directed force (Newton's first and second law). The structure exerts such a force when it is displaced from its original equilibrium by the force field. The displacement at every point of the structure is called the *displacement field* which can be found by solving a Poisson problem. The solution to this problem, and thereby the displacement field, can be approximated using the FEM.

The discretization described in Section 2.1 is used by the FEM to approximate the displacement field. A set of *control points* indexed by c are associated with each element e. Furthermore, a set of polynomial basis functions are defined across each element and assembled in the matrix $\boldsymbol{N}_e(\boldsymbol{p}, \boldsymbol{x})$. The basis functions depend on the shape of the element, i.e. the vertex positions contained in $\boldsymbol{p}$, and are functions of the position $\boldsymbol{x} = [x, y, z]^T$. The number of control points and basis functions are dependent on both the accuracy of the approximation and the dimensionality. In Section 2.2.5, we consider linear basis functions in 2D where the control points are placed at the vertices of the triangular elements. Furthermore, quadratic basis functions in 2D are considered in Section 2.2.6. Here, the control points are placed both at the vertices and at the center of the edges.

We can model the displacement field across element e by using its basis functions $\boldsymbol{N}_e(\boldsymbol{p}, \boldsymbol{x})$ to weight the displacements at the element control points $\boldsymbol{u}_e$. Consequently, the displacement $\boldsymbol{u}(\boldsymbol{x})$ at any interior point $\boldsymbol{x}$ on element e is approximated by

$$\boldsymbol{u}(\boldsymbol{x}) \approx \boldsymbol{N}_e(\boldsymbol{p}, \boldsymbol{x})\boldsymbol{u}_e \tag{2.1}$$

2.2.1 Strain and stress

We can now compute the strain and stress at any point $\boldsymbol{x}$ on the structure by using the approximation for the displacement field in Equation 2.1.

The strain $\boldsymbol{\epsilon}(\boldsymbol{x})$ at position $\boldsymbol{x}$ which is inside element e is the change in the displacement field at $\boldsymbol{x}$. Consequently,

$$\boldsymbol{\epsilon}(\boldsymbol{x}) \approx \boldsymbol{d}_{\boldsymbol{x}}\boldsymbol{N}_e(\boldsymbol{p}, \boldsymbol{x})\boldsymbol{u}_e = \boldsymbol{B}_e(\boldsymbol{p}, \boldsymbol{x})\boldsymbol{u}_e \tag{2.2}$$

The *strain-displacement matrix* $\boldsymbol{B}_e(\boldsymbol{p}, \boldsymbol{x})$ therefore describes the relation between the strain and the displacement. Furthermore, as an example, the differential operator in 2D is

$$\boldsymbol{d}_{\boldsymbol{x}} = \begin{bmatrix} \frac{d}{dx} & 0 \\ 0 & \frac{d}{dy} \\ \frac{d}{dy} & \frac{d}{dx} \end{bmatrix} \tag{2.3}$$

The stress $\boldsymbol{\sigma}(\boldsymbol{x})$ at position $\boldsymbol{x}$ inside element e is related to strain by Hooke's law for continuous media

$$\boldsymbol{\sigma}(\boldsymbol{x}) = \boldsymbol{E}_e(\boldsymbol{m})\boldsymbol{\epsilon}(\boldsymbol{x}) \approx \boldsymbol{E}_e(\boldsymbol{m})\boldsymbol{B}_e(\boldsymbol{p}, \boldsymbol{x})\boldsymbol{u}_e \tag{2.4}$$

where $\boldsymbol{E}_e(\boldsymbol{m})$ is the *constitutive matrix* consisting of elastic constants. For example, in the 2D case, for isotropic materials and under the plane stress conditions

$$\boldsymbol{E}_e(\boldsymbol{m}) = \frac{E_e}{1-\nu_e^2}\begin{bmatrix} 1 & \nu_e & 0 \\ \nu_e & 1 & 0 \\ 0 & 0 & \frac{1-\nu_e}{2} \end{bmatrix} \tag{2.5}$$

where E_e is Young's modulus and ν_e is Poisson's ratio for material m_e associated with element e.

2.2.2 Static equilibrium equations

Using the approximation to the continuous displacement field in Equation 2.1, the weak formulation of the Poisson problem takes the discrete form

$$\boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p}) \tag{2.6}$$

which is called the *static equilibrium equations*. These equations are used to estimate the *global displacement vector* $\boldsymbol{u} = [\ldots, \boldsymbol{u}_c^T, \ldots]^T$ and thereby the displacement $\boldsymbol{u}_c$ at every control point c (illustrated in Figure 2.3). Furthermore, $\boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p})$ is the *global force vector* which is a discrete approximation to the force field and described in Section 2.2.3. Finally, $\boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p})$ is the *global stiffness matrix* which is computed as described in Section 2.2.4. We will impose Dirichlet boundary conditions to ensure a unique solution to the Poisson problem. Consequently, the stiffness matrix is altered such that the equations involving the supported control points c are $\boldsymbol{u}_c = \boldsymbol{0}$.

The number of unknowns in the equilibrium equations is proportional to the number of unsupported control points and is therefore often huge. Consequently, estimating the global displacement vector $\boldsymbol{u}$ is computationally expensive. However, the global stiffness matrix is sparse and we can therefore use the fast sparse solver *CHOLMOD*[2] which is a part of the *SuiteSparse* library (Davis et al., 2013).

[2] CHOLMOD is the default solver for sparse symmetric positive definite linear systems in MATLAB.

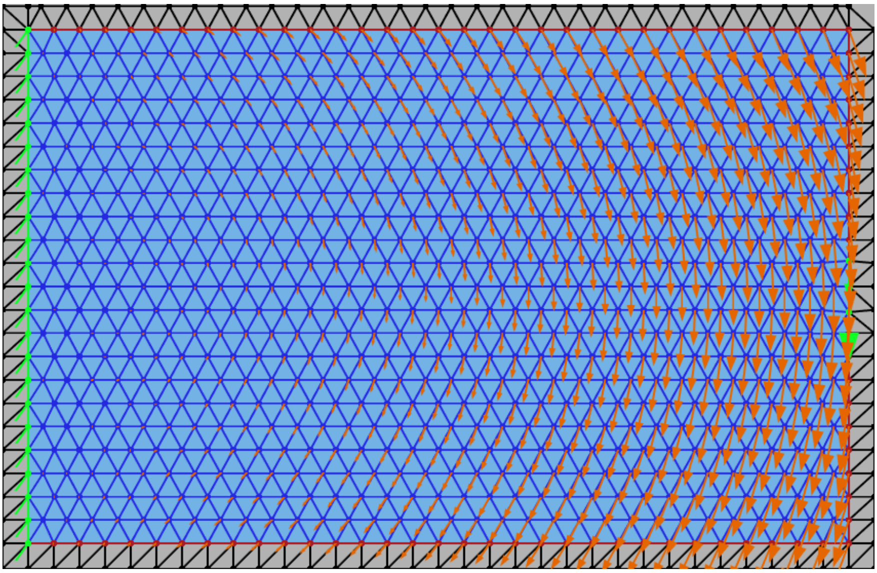


Figure 2.3: The orange arrows illustrate the displacements at the control points as estimated by the FEA for the cantilever beam problem.

2.2.3 Force vector

External forces are distributed across the surface area such that $\boldsymbol{f}_c$ is the force vector at control point c. Furthermore, in some cases, we will model internal forces at control point c due to gravity

$$\boldsymbol{w}_c(\boldsymbol{m}, \boldsymbol{p}) = \boldsymbol{g} \cdot \sum_{e \in c} a_e \cdot \rho_e \cdot V_e(\boldsymbol{p}) \tag{2.7}$$

Here, $\boldsymbol{g}$ is the gravitational acceleration vector and ρ_e and V_e are the density and volume of element e respectively. Furthermore, the scale factors a_e are computed by a mass lumping scheme such that

$$\sum_{c \in e} \boldsymbol{w}_c(\boldsymbol{m}, \boldsymbol{p}) = \boldsymbol{g} \cdot \rho_e \cdot V_e(\boldsymbol{p}) \tag{2.8}$$

The external and internal forces are assembled in the global force vector, i.e.

$$\boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p}) = [\ldots, \boldsymbol{f}_c^T + \boldsymbol{w}_c(\boldsymbol{m}, \boldsymbol{p})^T, \ldots]^T \tag{2.9}$$

Note that $\boldsymbol{f}$ is a function of the positions $\boldsymbol{p}$ and materials $\boldsymbol{m}$ only when gravity is taken into account.

2.2.4 Stiffness matrix

The global stiffness matrix $\boldsymbol{K}(\boldsymbol{m},\boldsymbol{p})$ is assembled by

$$\boldsymbol{K}(\boldsymbol{m},\boldsymbol{p}) = \underset{e}{\mathbb{A}}\,\boldsymbol{K}_e(\boldsymbol{m},\boldsymbol{p}) \tag{2.10}$$

where $\underset{e}{\mathbb{A}}$ is the FE assembly operator and $\boldsymbol{K}_e(\boldsymbol{m},\boldsymbol{p})$ is the *local stiffness matrix* of element e which can be computed by integrating across the element volume V_e

$$\boldsymbol{K}_e(\boldsymbol{m},\boldsymbol{p}) = \int_{V_e} \boldsymbol{B}_e(\boldsymbol{p},\boldsymbol{x})^T \boldsymbol{E}_e(\boldsymbol{m}) \boldsymbol{B}_e(\boldsymbol{p},\boldsymbol{x})\ \partial(x,y,z) \tag{2.11}$$

In the following, we illustrate the computation of the local stiffness matrix in 2D when assuming an isotropic material model and that the thickness t is constant across the design domain. Consequently, in this case, Equation 2.11 reduces to an integral across the area A_e of element e

$$\boldsymbol{K}_e(\boldsymbol{m},\boldsymbol{p}) = t \cdot \int_{A_e} \boldsymbol{B}_e(\boldsymbol{p},\boldsymbol{x})^T \boldsymbol{E}_e(\boldsymbol{m}) \boldsymbol{B}_e(\boldsymbol{p},\boldsymbol{x})\ \partial(x,y) \tag{2.12}$$

Instead of using custom basis functions for each triangular element, which is computationally expensive, we will use a mapping. Consequently, a point $\boldsymbol{x} = [x,y]^T$ on the interior of an element is mapped into coordinates $\boldsymbol{a} = [a,b]^T$ as illustrated in Figure 2.4. The mapping between the differential operators is described by the Jacobian matrix $\boldsymbol{J}_e(\boldsymbol{p})$

$$\begin{bmatrix} \frac{d}{da} \\ \frac{d}{db} \end{bmatrix} = \boldsymbol{J}_e(\boldsymbol{p}) \begin{bmatrix} \frac{d}{dx} \\ \frac{d}{dy} \end{bmatrix} \Leftrightarrow \begin{bmatrix} \frac{d}{da} \\ \frac{d}{db} \end{bmatrix} = \begin{bmatrix} \frac{dx}{da} & \frac{dy}{da} \\ \frac{dx}{db} & \frac{dy}{db} \end{bmatrix} \begin{bmatrix} \frac{d}{dx} \\ \frac{d}{dy} \end{bmatrix} \tag{2.13}$$

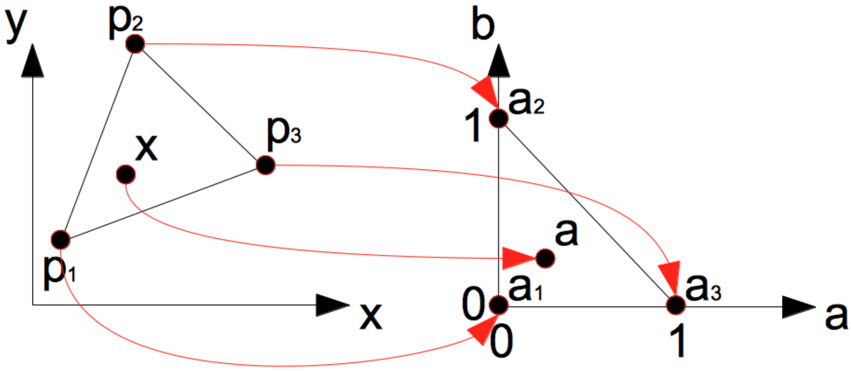


Figure 2.4: The mapping from coordinates $\boldsymbol{x} = [x,y]^T$ to $\boldsymbol{a} = [a,b]^T$.

Since we use straight sided triangles the Jacobian matrix is given by

$$\boldsymbol{J}_e(\boldsymbol{p}) = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \end{bmatrix} \tag{2.14}$$

This change of variable leads to the following expression for the local stiffness matrix

$$\boldsymbol{K}_e(\boldsymbol{m}, \boldsymbol{p}) = t \cdot |\boldsymbol{J}_e(\boldsymbol{p})| \cdot \int_{A_e} \boldsymbol{B}_e(\boldsymbol{p}, \boldsymbol{a})^T \boldsymbol{E}_e(\boldsymbol{m}) \boldsymbol{B}_e(\boldsymbol{p}, \boldsymbol{a}) \ \partial(a, b) \tag{2.15}$$

where

$$\begin{aligned} \boldsymbol{B}_e(\boldsymbol{p}, \boldsymbol{a}) = & \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{J}_e(\boldsymbol{p})^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{J}_e(\boldsymbol{p})^{-1} \end{bmatrix} \begin{bmatrix} \frac{d}{da} & 0 \\ \frac{d}{db} & 0 \\ 0 & \frac{d}{da} \\ 0 & \frac{d}{db} \end{bmatrix} \boldsymbol{N}(\boldsymbol{a}) \\ = & \boldsymbol{L} \boldsymbol{G}_e(\boldsymbol{p}) \boldsymbol{d_a} \boldsymbol{N}(\boldsymbol{a}) \end{aligned} \tag{2.16}$$

2.2.5 Linear basis functions

First, we consider the case where linear basis functions are used for interpolating across triangular elements. In this case, one control point c is associated with each vertex of the triangular element e. Therefore, the positions of the control points associated with element e are

$$\boldsymbol{p}_e = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \end{bmatrix} \tag{2.17}$$

and the linear basis functions are

$$\begin{aligned} N_1(a, b) &= 1 - a - b \\ N_2(a, b) &= a \\ N_3(a, b) &= b \end{aligned} \tag{2.18}$$

These basis functions are assembled in the matrix

$$\boldsymbol{N}(\boldsymbol{a}) = \begin{bmatrix} 1 - a - b & 0 & a & 0 & b & 0 \\ 0 & 1 - a - b & 0 & a & 0 & b \end{bmatrix} \tag{2.19}$$

and, when we apply the differential operator $\boldsymbol{d_a}$ to this matrix, we get

$$\boldsymbol{d_a N}(\boldsymbol{a}) = \begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.20}$$

It is evident that this matrix is not dependent on $\boldsymbol{a}$ and thereby $\boldsymbol{B}_e$ is not dependent on $\boldsymbol{a}$ either. The expression for the strain-displacement matrix becomes

$$\boldsymbol{B}_e(\boldsymbol{p}) = \frac{1}{|\boldsymbol{J}_e(\boldsymbol{p})|} \begin{bmatrix} y_2 - y_3 & 0 & y_3 - y_1 & 0 & y_1 - y_2 & 0 \\ 0 & x_3 - x_2 & 0 & x_1 - x_3 & 0 & x_2 - x_1 \\ x_3 - x_2 & y_2 - y_3 & x_1 - x_3 & y_3 - y_1 & x_2 - x_1 & y_1 - y_2 \end{bmatrix} \tag{2.21}$$

From Equation 2.3 we see that the strain is constant across each triangle which is the reason why these elements are called *Constant Strain Triangles* (CST).

Finally, the expression for the local stiffness matrix becomes

$$\begin{aligned} \boldsymbol{K}_e(\boldsymbol{m}, \boldsymbol{p}) &= t \cdot |\boldsymbol{J}_e(\boldsymbol{p})| \cdot \boldsymbol{B}_e(\boldsymbol{p})^T \boldsymbol{E}_e(\boldsymbol{m}) \boldsymbol{B}_e(\boldsymbol{p}) \int_{A_e} 1 \; \partial(a, b) \\ &= t \cdot A_e(\boldsymbol{p}) \cdot \boldsymbol{B}_e(\boldsymbol{p})^T \boldsymbol{E}_e(\boldsymbol{m}) \boldsymbol{B}_e(\boldsymbol{p}) \end{aligned} \tag{2.22}$$

since $|\boldsymbol{J}_e(\boldsymbol{p})| = 2 \cdot A_e(\boldsymbol{p})$ and $\int_{A_e} 1 \; \partial(a, b) = \frac{1}{2}$.

2.2.6 Quadratic basis functions

Next, we use quadratic basis functions and triangular elements which has 6 associated control points, one at each vertex and one at the center of each edge. Therefore the positions of the control points associated with element e are

$$\boldsymbol{p}_e = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ x_4 \\ y_4 \\ x_5 \\ y_5 \\ x_6 \\ y_6 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \\ (x_1 + x_2)/2 \\ (y_1 + y_2)/2 \\ (x_2 + x_3)/2 \\ (y_2 + y_3)/2 \\ (x_3 + x_1)/2 \\ (y_3 + y_1)/2 \end{bmatrix} \tag{2.23}$$

and the six basis functions are

$$\begin{aligned}N_1(a,b) &= 2(1-a-b)^2-(1-a-b)\\ N_2(a,b) &= 2a^2-a\\ N_3(a,b) &= 2b^2-b\\ N_4(a,b) &= 4(1-a-b)a\\ N_5(a,b) &= 4ab\\ N_6(a,b) &= 4b(1-a-b)\end{aligned} \tag{2.24}$$

Again, we can arrange these basis functions in a shape function matrix $\boldsymbol{N}(\boldsymbol{a})$ and use Equation 2.16 to compute the strain-displacement matrix. The expressions become relatively big and are therefore omitted but the derivations are similar to the case where we use linear basis functions. However, here, when applying the differential operator $\boldsymbol{d_a}$ on $\boldsymbol{N}(\boldsymbol{a})$, we get a linear dependency on $\boldsymbol{a}$ and, consequently, these elements are called *Linear Strain Triangles* (LST). Gaussian quadrature integration is applied to get an exact estimation of the local stiffness matrix

$$\begin{aligned}\boldsymbol{K}_e(\boldsymbol{m},\boldsymbol{p}) &= t\cdot|\boldsymbol{J}_e(\boldsymbol{p})|\cdot\int_{A_e}\boldsymbol{B}_e(\boldsymbol{p},\boldsymbol{a})^T\boldsymbol{E}_e(\boldsymbol{m})\boldsymbol{B}_e(\boldsymbol{p},\boldsymbol{a})\ \partial(a,b)\\ &= t\cdot A_e(\boldsymbol{p})\cdot\sum_{i=1}^{G_n}w_i\boldsymbol{B}_e(\boldsymbol{p},\boldsymbol{a}_i)^T\boldsymbol{E}_e(\boldsymbol{m})\boldsymbol{B}_e(\boldsymbol{p},\boldsymbol{a}_i)\end{aligned} \tag{2.25}$$

where we use the $G_n = 3$ location weight pairs

$$\begin{aligned}(a_1,b_1,w_1) &= (1/2,0,1/3)\\ (a_2,b_2,w_2) &= (0,1/2,1/3)\\ (a_3,b_3,w_3) &= (1/2,1/2,1/3)\end{aligned} \tag{2.26}$$

2.3 Structural properties

We now show four examples of structural properties which can be computed. The first two, volume and center of gravity, do not require a FEA whereas the following two, strain energy density and von Mises stress, do.

Example 2.1 (Volume)
The volume V of the structure is computed by

$$V(\boldsymbol{m},\boldsymbol{p}) = \sum_e H(m_e)\cdot V_e(\boldsymbol{p}) \tag{2.27}$$

where

$$H(m_e) = \begin{cases} 1, & \textit{if } m_e \textit{ is solid} \\ 0, & \textit{if } m_e \textit{ is void} \end{cases} \tag{2.28}$$

Example 2.2 (Center of gravity)

The center of gravity $\boldsymbol{c}_g$ of the structure is computed by

$$\boldsymbol{c}_g(\boldsymbol{m}, \boldsymbol{p}) = \frac{1}{\sum_e \rho_e \cdot V_e(\boldsymbol{p})} \sum_e \rho_e \cdot V_e(\boldsymbol{p}) \cdot \boldsymbol{c}_e(\boldsymbol{p}) \tag{2.29}$$

where $\boldsymbol{c}_e(\boldsymbol{p})$ is the barycenter of element e. Note that void elements do not contribute to the center of gravity since they have zero density, i.e. $\rho_e = 0$.

Example 2.3 (Strain energy density)

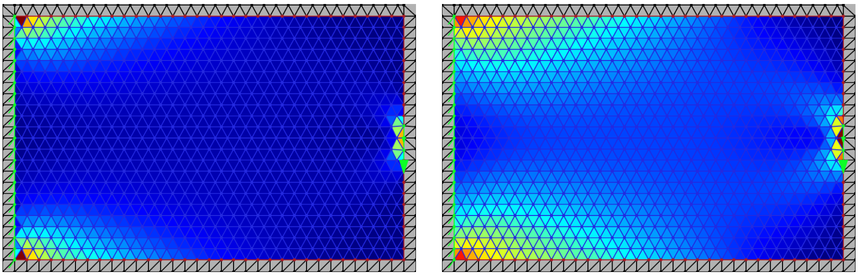
The strain energy density (SED) s_e of an element e is

$$s_e(\boldsymbol{m}, \boldsymbol{p}) = \boldsymbol{u}_e^T \boldsymbol{K}_e(\boldsymbol{m}, \boldsymbol{p}) \boldsymbol{u}_e \tag{2.30}$$

where the displacements $\boldsymbol{u}_e$ are estimated by solving Equation 2.6. A visualization of the strain energy density as approximated by the FEA, i.e. the element strain energy densities, can be seen in Figure 2.5(a).

Example 2.4 (von Mises stress)

To get a single value which reflects the stress in element e, we compute the von Mises stress evaluated at the centroid of the element, i.e. for $\boldsymbol{a} = \frac{\boldsymbol{1}}{\boldsymbol{3}}$. Under



(a) Strain energy density

(b) von Mises stress

Figure 2.5: The strain energy density and the von Mises stress of the structure for the cantilever beam problem as estimated by the FEA. The 'jet' colormap is used where the values are scaled such that blue is low and red is high.

the plane stress conditions the centroidal von Mises stress (depicted in Figure 2.5(b)) is

$$\begin{aligned}\sigma_e(\boldsymbol{m},\boldsymbol{p}) &= \sqrt{\boldsymbol{\sigma}_e(\boldsymbol{m},\boldsymbol{p})^T\boldsymbol{A}\boldsymbol{\sigma}_e(\boldsymbol{m},\boldsymbol{p})} \\ &= \sqrt{(\boldsymbol{E}_e(\boldsymbol{m})\boldsymbol{B}_e(\boldsymbol{p})\boldsymbol{u}_e)^T\boldsymbol{A}\boldsymbol{E}_e(\boldsymbol{m})\boldsymbol{B}_e(\boldsymbol{p})\boldsymbol{u}_e} \\ &= \sqrt{\boldsymbol{u}_e^T\boldsymbol{B}_e(\boldsymbol{p})^T\boldsymbol{E}_e(\boldsymbol{m})\boldsymbol{A}\boldsymbol{E}_e(\boldsymbol{m})\boldsymbol{B}_e(\boldsymbol{p})\boldsymbol{u}_e}\end{aligned} \tag{2.31}$$

where the matrix $\boldsymbol{A}$ *in 2D is*

$$\boldsymbol{A} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \tag{2.32}$$

CHAPTER 3

Optimization

The structural analysis makes it possible to evaluate structural properties. Consequently, objective and constraint functions can now be defined based on these properties. A few examples of such functions are given in the beginning of this chapter. Furthermore, regularization of the optimization problem is considered. Next, we describe how an objective function can be minimized subject to global constraints using a gradient-based optimization procedure. The optimization procedure consists of alternating between two steps; a discrete and a continuous optimization step. These steps are then iterated until convergence. Using a gradient-based optimization procedure implies computing the derivatives of the objective and constraint functions with respect to the design variables. This will be a main part of the chapter.

3.1 Objective and constraint functions

The objective function f is a function of the shape and topology of the structure. Hence, the *design variables* – the parameters that we change to optimize the objective function – are the node positions $\boldsymbol{p} = [\ldots, \boldsymbol{p}_n^T, \ldots]^T$ and the element materials $\boldsymbol{m} = [\ldots, m_e, \ldots]^T$. The objective function hereby takes the form $f(\boldsymbol{m}, \boldsymbol{p})$.

In the following, three examples of objective functions – compliance, sum of von Mises stress and maximal von Mises stress – are given.

Example 3.1 (Compliance)

To minimize compliance, or equivalently maximize stiffness, the objective function is

$$f(\boldsymbol{m}, \boldsymbol{p}) = \sum_e s_e(\boldsymbol{m}, \boldsymbol{p}) = \sum_e \boldsymbol{u}_e^T \boldsymbol{K}_e(\boldsymbol{m}, \boldsymbol{p}) \boldsymbol{u}_e = \boldsymbol{u}^T \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p}) \boldsymbol{u} \tag{3.1}$$

Example 3.2 (Sum of von Mises stress)

Another possible objective is the sum of the von Mises stress

$$f(\boldsymbol{m}, \boldsymbol{p}) = \sum_e \sigma_e(\boldsymbol{m}, \boldsymbol{p}) \tag{3.2}$$

Example 3.3 (Maximal von Mises stress)

Finally, minimizing the maximal von Mises stress, thereby penalizing large stress concentrations, is also a possibility

$$f(\boldsymbol{m}, \boldsymbol{p}) = \max(\ldots, \sigma_e(\boldsymbol{m}, \boldsymbol{p}), \ldots) \tag{3.3}$$

However, the max-function is not smooth or differentiable and the p-norm approximation (Duysinx and Sigmund, 1998) is therefore used instead

$$\max(\ldots, \sigma_e(\boldsymbol{m}, \boldsymbol{p}), \ldots) \approx \left(\sum_e \sigma_e(\boldsymbol{m}, \boldsymbol{p})^p \right)^{\frac{1}{p}} \tag{3.4}$$

Note that all of these examples are functions of the structural properties estimated by the FEA in Chapter 2. Consequently, to evaluate these objectives, the equilibrium equations need to be solved.

In addition to the objective function, q global constraints $g_i(\boldsymbol{m}, \boldsymbol{p})$, $i = 1, \ldots, q$ are applied such that

$$g_i(\boldsymbol{m}, \boldsymbol{p}) \leq 0, \; i = 1, \ldots, q \tag{3.5}$$

Note that the objective function and any of the global constraints can be interchanged. A typical example of a global constraint used in topology optimization, which we will use extensively, is the volume constraint.

Example 3.4 (Constraint: Volume)
The material volume constraint

$$g_V(\boldsymbol{m}, \boldsymbol{p}) = \frac{V(\boldsymbol{m}, \boldsymbol{p})}{V^*} - 1 \tag{3.6}$$

is a global constraint that limits the amount of material that is at the optimization algorithms disposal. Furthermore, V^* is the imposed maximum volume of material.

3.2 Regularization

Topology optimization problems possibly have no optimal solution, contain local optima and are subjected to numerical instabilities. In general, the problems are non-convex and, therefore, there is no guarantee to find the optimal (or any) solution. There are two types of approaches to deal with this; relaxation and regularization.

Relaxation

Relaxation enlarges the set of admissible solutions. In topology optimization, this means that structures which are not manufacturable are allowed to be able to smoothly go from one manufacturable structure to another more optimal manufacturable structure. Of course this makes structures that are not manufacturable valid solutions which is remedied by penalising these unwanted structures.

Regularization

Regularization is the opposite approach to relaxation since it reduces the set of admissible solutions. It does so by adding constraints for example on the smoothness or thickness of the structure. These constraints are also known as regularizing terms and are widely used in engineering problems to be able to solve hard optimization problems. The downside of this approach is that it may exclude the optimal structure from the set of admissible solutions.

We use regularization in our approach since we rather will exclude some optimal solutions that are not feasible to manufacture (e.g. because of very thin parts of the structure) than allowing non-physical structures.

In some applications, we will apply a global perimeter constraint to limit thin beams and tiny details.

Example 3.5 (Constraint: Perimeter)
The global perimeter constraint

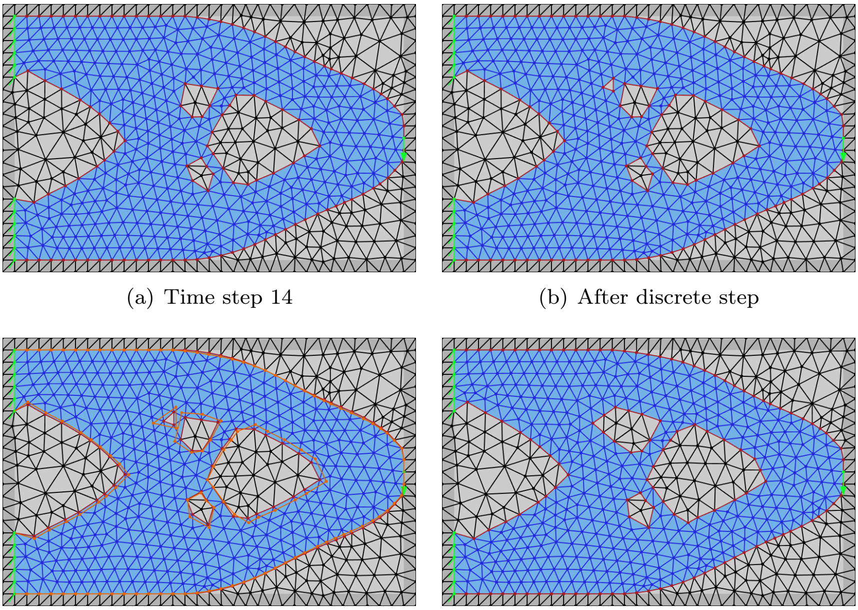
$$g_P(\boldsymbol{m}, \boldsymbol{p}) = \frac{P(\boldsymbol{m}, \boldsymbol{p})}{P^*} - 1 \tag{3.7}$$

limits the total area of the surface of the structure. Here, $P(\boldsymbol{m}, \boldsymbol{p})$ is the total surface area and P^* is the imposed maximum perimeter.

Furthermore, sometimes a jagged surface is perceived as optimal by the optimization procedure. This is a well-known problem (Ding, 1986; Mohammadi and Pironneau, 2001) and, consequently, several solutions exist. When minimizing compliance in Paper A, we find that using quadratic basis functions instead of linear for the FEA solves this problem. However, when minimizing the maximal von Mises stress, this is not the case and regularization is therefore necessary. Consequently, a Gaussian smoothing filter is applied as described in Paper C.

3.3 Optimization procedure

The optimization procedure consists of alternating between a discrete and a continuous optimization step as illustrated in Figure 3.1. These two steps are iterated until convergence, i.e. until changes between consecutive time steps are small.



(a) Time step 14

(b) After discrete step

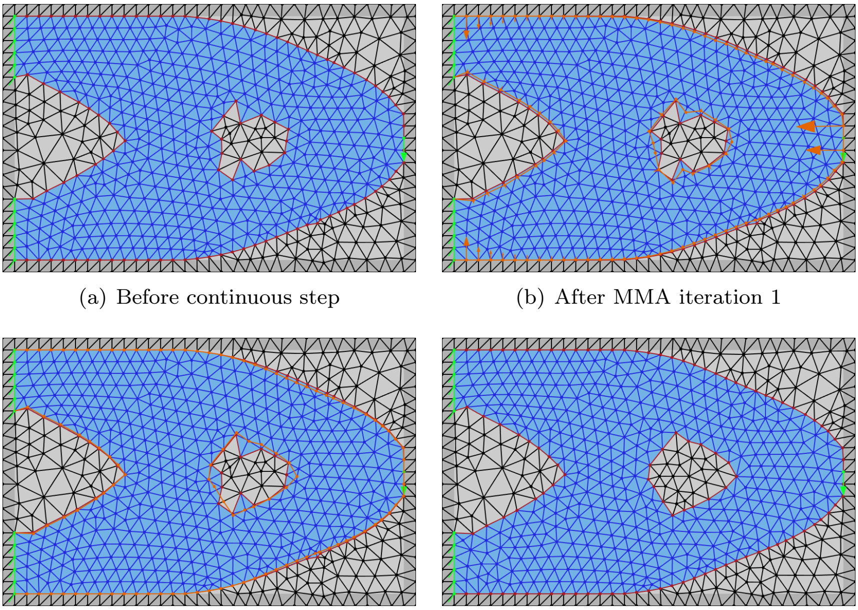
(c) Optimized shape computed

(d) After continuous step

Figure 3.1: One time step of the optimization method. First, the discrete step relabels elements from solid to void using a heuristic optimization step based on discrete derivatives. Secondly, the continuous optimization step computes a more optimal shape using MMA and uses the DSC method to deform the structure to this new shape. These two steps are iterated until convergence.

3.3.1 Continuous optimization step

The continuous optimization step, illustrated in Figure 3.2, is inspired by non-parametric shape optimization procedures (Ding, 1986; Mohammadi and Pironneau, 2001; Bucur and Buttazzo, 2006; Le et al., 2011; Arnout et al., 2012). Consequently, the positions of the *design nodes* (contained in the vector $\boldsymbol{p} = [\ldots, \boldsymbol{p}_n^T, \ldots]^T$) are changed such that the objective function is minimized subject to global constraints. The design nodes are the surface nodes which are not supported (except in Paper C) and not subjected to loads. Since the design nodes are moved by the DSC method, the continuous optimization step will change the topology of a structure if two surfaces collide. Consequently, this step may close holes and collapse thin beams but is not able to introduce holes inside the structure. Finally, the materials $\boldsymbol{m}$ are not changed during this step.



(a) Before continuous step

(b) After MMA iteration 1

(c) After MMA iteration 4 (convergence)

(d) After continuous step

Figure 3.2: A continuous optimization step computes improved positions for the design nodes using MMA which, in this case, converged in 4 iterations. The design nodes are then moved from their original positions (red) to the more optimal positions (orange) using the DSC method. Furthermore, the orange arrows indicate the continuous gradients for the design nodes. Note that this step can change both the shape and topology of the structure.

Moving a design node in the tangent direction will not change the surface much. Consequently, to reduce the number of design variables a change of variable is applied such that one design variable x_n is associated with each design node n. The relation between the improved position $\boldsymbol{p}_n(x_n)$, the original position $\boldsymbol{p}_n^0$ and the design variable x_n is given by

$$\boldsymbol{p}_n(x_n) = \boldsymbol{p}_n^0 + x_n \cdot \boldsymbol{n}_n \tag{3.8}$$

where $\boldsymbol{n}_n$ is the normal to design node n. Furthermore, the design variables are assembled in a vector $\boldsymbol{x} = [\ldots, x_n, \ldots]^T$ (which should not be confused with the position $\boldsymbol{x}$ from Chapter 2).

To estimate the optimized design variables $\boldsymbol{x}^* = [\ldots, x_n^*, \ldots]^T$, a smooth non-linear optimization problem is solved

$$\begin{aligned} \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} &: f(\boldsymbol{x}) \\ subject\ to &: g_i(\boldsymbol{x}) \leq 0,\ i = 1, \ldots, q \\ &: \boldsymbol{K}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{x}) \\ &: \boldsymbol{x}_{min} \leq \boldsymbol{x} \leq \boldsymbol{x}_{max} \end{aligned} \tag{3.9}$$

Here, the design domain is enforced via the move limits on the design variables $\boldsymbol{x}_{min}$ and $\boldsymbol{x}_{max}$. Generally, the move limits restrict the search for the optimal position of the design variables to a small neighbourhood around each variable. The neighbourhood is chosen such that elements do not degenerate during the optimization and it therefore depends on the discretization. This implies that the shape only changes slightly at each continuous optimization step which is remedied by applying this step, and the discrete optimization step, repeatedly.

Several optimization algorithms exist to solve this type of smooth, non-linear optimization problems. In this work, the iterative gradient based method *Method of Moving Asymptotes (MMA)* (Svanberg, 1987) has been applied. When an optimized design variable x_n^*, and thereby an optimized position $\boldsymbol{p}_n^* = \boldsymbol{p}_n(x_n^*)$, has been computed for each design node n, the nodes are moved by the DSC method as described in Chapter 4.

Continuous derivatives

To be able to use a gradient-based optimization algorithm to solve Equation 3.9, we need to compute the derivatives of the objective and constraint functions with respect to each design variable x_n. We will compute these derivatives based on standard analytical theory (Choi and Kim, 2010).

First, the chain rule is used to accommodate for the change of variable

$$\frac{\partial f(\boldsymbol{x})}{\partial x_n} = \frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\frac{\partial \boldsymbol{p}_n(x_n)}{\partial x_n} = \frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} \cdot \boldsymbol{n}_n \tag{3.10}$$

and analogous for the derivative of constraints $\frac{\partial g_i(\boldsymbol{x})}{\partial x_n}$.

Next, the continuous derivatives $\frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n}$ and $\frac{\partial g_i(\boldsymbol{p})}{\partial \boldsymbol{p}_n}$ (exemplified for compliance in Figure 3.2) are computed. For all constraints considered here, computing their continuous derivatives $\frac{\partial g_i(\boldsymbol{p})}{\partial \boldsymbol{p}_n}$ are straightforward. However, often, the objective is a function of the displacement vector $\boldsymbol{u}$ which is indirectly dependent on the design variables $\boldsymbol{p}$ through the equilibrium equations. This means that the expression for the continuous derivative contains the term $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{p}_n}$ which has to be approximated numerically through evaluations of the equilibrium equations. Computing $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{p}_n}$ for all design variables is therefore computationally very expensive. Consequently, we will use the adjoint variable method (Pironneau, 1982; Christensen and Klarbring, 2008) which results in an expression for the continuous derivative without $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{p}_n}$.

The idea of the adjoint variable method is to alter the objective function by adding the term $\boldsymbol{\lambda}^T(\boldsymbol{K}(\boldsymbol{p})\boldsymbol{u} - \boldsymbol{f}(\boldsymbol{p}))$ where $\boldsymbol{\lambda}$ is a vector of Lagrange multipliers. Since this term is zero, due to static equilibrium (Equation 2.6), it will not alter the objective function. Consequently, the new objective function is

$$h(\boldsymbol{p}) = f(\boldsymbol{p}) + \boldsymbol{\lambda}^T(\boldsymbol{K}(\boldsymbol{p})\boldsymbol{u} - \boldsymbol{f}(\boldsymbol{p})) \tag{3.11}$$

and the gradients of this new objective function can be computed by

$$\begin{aligned}
&\frac{\partial h(\boldsymbol{p})}{\partial \boldsymbol{p}_n} = \\
&\frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} + \frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{u}}\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{p}_n} + \boldsymbol{\lambda}^T\left(\frac{\partial \boldsymbol{K}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\boldsymbol{u} + \boldsymbol{K}(\boldsymbol{p})\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{p}_n} - \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\right) = \\
&\frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} + \left(\frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{u}} + \boldsymbol{\lambda}^T\boldsymbol{K}(\boldsymbol{p})\right)\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{p}_n} + \boldsymbol{\lambda}^T\left(\frac{\partial \boldsymbol{K}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\boldsymbol{u} - \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\right)
\end{aligned} \tag{3.12}$$

To get rid of the problematic term $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{p}_n}$, we simply find $\boldsymbol{\lambda}$ such that

$$\boldsymbol{K}(\boldsymbol{p})^T\boldsymbol{\lambda} = -\frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{u}} \tag{3.13}$$

which is called the *adjoint problem.* Note that the local stiffness matrices $\boldsymbol{K}_e(\boldsymbol{p})$, and thereby $\boldsymbol{K}(\boldsymbol{p})$, are symmetric so $\boldsymbol{K}(\boldsymbol{p})^T = \boldsymbol{K}(\boldsymbol{p})$.

Finally, Equation 3.13 is used to derive an expression for the gradients which is much less expensive to evaluate

$$\frac{\partial h(\boldsymbol{p})}{\partial \boldsymbol{p}_n} = \frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} + \boldsymbol{\lambda}^T\left(\frac{\partial \boldsymbol{K}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\boldsymbol{u} - \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\right) \tag{3.14}$$

To evaluate this, $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ are found by solving the equilibrium equations in Equation 2.6 and the adjoint problem in Equation 3.13 respectively. The partial derivatives $\frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n}$, $\frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{u}}$ and $\frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}$ can be evaluated when the objective function $f(\boldsymbol{p})$ and the load vector $\boldsymbol{f}(\boldsymbol{p})$ are known. Furthermore, the derivative of the global stiffness matrix $\frac{\partial \boldsymbol{K}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}$ can be derived from the expression in equations 2.10 and 2.11 and evaluated using Gaussian quadrature integration.

Next, we present three examples of deriving an expression for the continuous derivatives.

Example 3.6 (Compliance)
In the first example compliance is considered as the objective

$$f(\boldsymbol{p}) = \boldsymbol{u}^T \boldsymbol{K}(\boldsymbol{p})\boldsymbol{u} \tag{3.15}$$

Since the objective function contains the displacements $\boldsymbol{u}$*, the adjoint variable method is used. Therefore, the adjoint problem (Equation 3.13) is*

$$\boldsymbol{K}(\boldsymbol{p})\boldsymbol{\lambda} = -2 \cdot \boldsymbol{K}(\boldsymbol{p})\boldsymbol{u} \Leftrightarrow \boldsymbol{K}(\boldsymbol{p})\boldsymbol{\lambda} = -2 \cdot \boldsymbol{f}(\boldsymbol{p}) \Leftrightarrow \boldsymbol{\lambda} = -2 \cdot \boldsymbol{u} \tag{3.16}$$

In this special case, an analytical expression for $\boldsymbol{\lambda}$ *is found by using the equilibrium equations.*

An expression for the continuous gradients without the problematic term $\frac{\partial \boldsymbol{u}}{\partial \boldsymbol{p}_n}$ *can now be derived using equations 3.14 and 3.16*

$$\begin{aligned} \frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} =& \boldsymbol{u}^T \frac{\partial \boldsymbol{K}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\boldsymbol{u} - 2 \cdot \boldsymbol{u}^T \frac{\partial \boldsymbol{K}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\boldsymbol{u} + 2 \cdot \boldsymbol{u}^T \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}_n} \\ =& - \boldsymbol{u}^T \frac{\partial \boldsymbol{K}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\boldsymbol{u} + 2 \cdot \boldsymbol{u}^T \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}_n} \end{aligned} \tag{3.17}$$

Example 3.7 (Sum of von Mises stress)
Next, the sum of the element centroidal von Mises stresses is considered as the objective

$$f(\boldsymbol{p}) = \sum_e \sigma_e(\boldsymbol{p}) \tag{3.18}$$

Again, the objective function is a function of the displacements $\boldsymbol{u}$ *and, consequently, the adjoint variable method is used. The adjoint problem is*

$$\boldsymbol{K}(\boldsymbol{p})\boldsymbol{\lambda} = -\underset{e}{\mathbb{A}} \frac{1}{\sigma_e(\boldsymbol{p})} \cdot \boldsymbol{B}_e(\boldsymbol{p})^T \boldsymbol{E}_e \boldsymbol{A} \boldsymbol{E}_e \boldsymbol{B}_e(\boldsymbol{p}) \boldsymbol{u}_e \tag{3.19}$$

where $\underset{e}{\mathbb{A}}$ *is the FE assembly operator.*

When $\boldsymbol{\lambda}$ *has been computed, the gradients are computed by*

$$\begin{aligned} \frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{K}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\boldsymbol{u} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}_n} + \sum_e \frac{1}{2 \cdot \sigma_e(\boldsymbol{p})} \cdot \\ \left(\boldsymbol{u}_e^T \boldsymbol{B}_e(\boldsymbol{p})^T \boldsymbol{E}_e \boldsymbol{A} \boldsymbol{E}_e \frac{\partial \boldsymbol{B}_e(\boldsymbol{p})}{\partial \boldsymbol{p}_n} \boldsymbol{u}_e + \boldsymbol{u}_e^T \frac{\partial \boldsymbol{B}_e(\boldsymbol{p})}{\partial \boldsymbol{p}_n}^T \boldsymbol{E}_e \boldsymbol{A} \boldsymbol{E}_e \boldsymbol{B}_e(\boldsymbol{p}) \boldsymbol{u}_e \right) \end{aligned} \tag{3.20}$$

Example 3.8 (Maximal von Mises stress)
Finally, the p-norm of the element centroidal von Mises stresses is considered

$$f(\boldsymbol{p}) = \left(\sum_e \sigma_e(\boldsymbol{p})^p \right)^{\frac{1}{p}} \tag{3.21}$$

Here, the adjoint problem is

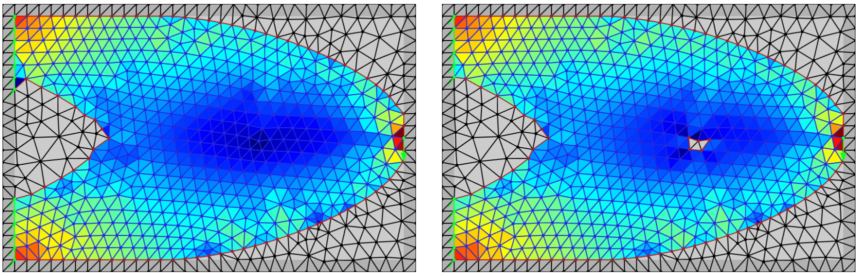
$$\boldsymbol{K}(\boldsymbol{p})\boldsymbol{\lambda} = -f(\boldsymbol{p})^{1-p} \cdot \underset{e}{\mathbb{A}}\ \sigma_e(\boldsymbol{p})^{p-2} \cdot \boldsymbol{B}_e(\boldsymbol{p})^T \boldsymbol{E}_e \boldsymbol{A} \boldsymbol{E}_e \boldsymbol{B}_e(\boldsymbol{p}) \boldsymbol{u}_e \tag{3.22}$$

and the gradients are

$$\begin{aligned}\frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{K}(\boldsymbol{p})}{\partial \boldsymbol{p}_n} \boldsymbol{u} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}_n} + f(\boldsymbol{p})^{1-p} \cdot \sum_e \frac{\sigma_e(\boldsymbol{p})^{p-2}}{2} \cdot \\ \left(\boldsymbol{u}_e^T \boldsymbol{B}_e(\boldsymbol{p})^T \boldsymbol{E}_e \boldsymbol{A} \boldsymbol{E}_e \frac{\partial \boldsymbol{B}_e(\boldsymbol{p})}{\partial \boldsymbol{p}_n} \boldsymbol{u}_e + \boldsymbol{u}_e^T \frac{\partial \boldsymbol{B}_e(\boldsymbol{p})}{\partial \boldsymbol{p}_n}^T \boldsymbol{E}_e \boldsymbol{A} \boldsymbol{E}_e \boldsymbol{B}_e(\boldsymbol{p}) \boldsymbol{u}_e \right)\end{aligned} \tag{3.23}$$

3.3.2 Discrete optimization step

In addition to the positions $\boldsymbol{p}$, the objective and constraint functions depend on the materials of the elements $\boldsymbol{m} = [\ldots, m_e, \ldots]^T$. Therefore, in this discrete optimization step, illustrated in Figure 3.3, the materials are changed such that the objective is minimized subject to the global constraints. Furthermore, the positions $\boldsymbol{p}$ are not changed during this step. The discrete optimization step has two purposes; to introduce holes inside the structure and to speed up the optimization.



(a) Before discrete step

(b) After discrete step

Figure 3.3: A discrete optimization step relabels element materials, e.g. from solid to void, based on discrete derivatives. The discrete derivatives are visualized by the 'jet' colormap and scaled such that minimum is blue and maximum is red. Note that this step can change both the shape and the topology of the structure.

The discrete optimization problem is

$$\begin{aligned} \boldsymbol{m}^* = \arg\min_{\boldsymbol{m}} &: f(\boldsymbol{m}) \\ \textit{subject to} &: g_i(\boldsymbol{m}) \leq 0, \; i = 1, \ldots, q \\ &: \boldsymbol{K}(\boldsymbol{m})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{m}) \\ &: m_e \in \{\ldots, m^m, \ldots\} \end{aligned} \tag{3.24}$$

where $\boldsymbol{m}^* = [\ldots, m_e^*, \ldots]^T$ are the optimized material labels. This optimization problem is computationally expensive to solve to optimality. However, since it is combined with the continuous optimization, Equation 3.24 does not have to be solved to optimality. Consequently, we will apply different heuristic strategies for finding an approximate solution to this problem as described in the included papers. However, all of them are in some way based on discrete derivatives which are described in the next section.

When an approximate solution is found, i.e. $\boldsymbol{m}^*$ has been estimated, the material label of each element e is changed from m_e to m_e^* using the DSC method as described in Chapter 4.

Discrete derivatives

The discrete derivatives for element e are the change in the objective and constraint functions when its material label is changed from the current material m_e to material m^m. This is written as

$$\begin{aligned} \Delta_e^m f(\boldsymbol{m}) &= f(\boldsymbol{m}_e^m) - f(\boldsymbol{m}) \\ \Delta_e^m g_i(\boldsymbol{m}) &= g_i(\boldsymbol{m}_e^m) - g_i(\boldsymbol{m}), \; i = 1, \ldots, q \end{aligned} \tag{3.25}$$

where $\boldsymbol{m}_e^m$ equals $\boldsymbol{m}$ with m_e replaced by m^m. The discrete derivatives of compliance are depicted in Figure 3.3.

The discrete derivatives of most objective and constraint functions can be computed efficiently using the expression in Equation 3.25 directly. However, computing the discrete derivatives of for example compliance involves solving the equilibrium equations once for each derivative computation. Consequently, these derivatives are computationally expensive and are therefore approximated.

The approximation will be based on the theory of topological derivatives (Eschenauer et al., 1994; Sokolowski and Zochowski, 1999; Garreau et al., 2000; Feijóo et al., 2003; de Gournay et al., 2008). The topological derivative is the change in objective function f when introducing an infitesimal hole in element e. The Hashin-Shtrikman bounds (Hashin and Shtrikman, 1963) are bounds on the stiffness of the least and the most optimal structure when combining two materials. If one of the materials is void and the part of the structure which is void is infinitely small, the derivative of the upper Hashin-Shtrikman bound is the same as the topological derivative. Since the density based approach approximate the upper Hashin-Shtrikman bound (Sigmund and Guest, 2014), and is exact for materials with Poisson's ratio $\frac{1}{3}$, we will use this to approximate the discrete derivatives. For the density approach, the objective is a function of the element densities $\boldsymbol{\rho} = [\ldots, \rho_e, \ldots]^T$ which are continuous variables. Consequently, the discrete derivative of changing the material of element e from solid (denoted by s and with density $\rho^s = 1$) to void (denoted by v and with density $\rho^v = 0$) can be approximated by

$$\Delta_e^v f(\boldsymbol{m}) = f(\boldsymbol{m}_e^v) - f(\boldsymbol{m}) \approx \frac{\partial f(\boldsymbol{\rho})}{\partial \rho_e} \cdot (\rho^v - \rho^s) = -\frac{\partial f(\boldsymbol{\rho})}{\partial \rho_e} \tag{3.26}$$

where $\frac{\partial f(\boldsymbol{\rho})}{\partial \rho_e}$ is evaluated for $\boldsymbol{\rho} = \mathbf{1}$. Now, the constitutive matrix of element e is a function of its density ρ_e

$$\boldsymbol{E}_e(\boldsymbol{\rho}) = \rho_e^\alpha \cdot \boldsymbol{E}_e \tag{3.27}$$

where α is a parameter that depends on the objective. Consequently,

$$\frac{\partial \boldsymbol{E}_e(\boldsymbol{\rho})}{\partial \rho_e} = \alpha \cdot \rho_e^{\alpha-1} \cdot \boldsymbol{E}_e \tag{3.28}$$

and

$$\frac{\partial \boldsymbol{K}(\boldsymbol{\rho})}{\partial \rho_e} = \alpha \cdot \rho_e^{\alpha-1} \cdot \boldsymbol{K}_e \tag{3.29}$$

Example 3.9 (Compliance)
First, compliance is considered

$$f(\boldsymbol{\rho}) = \boldsymbol{u}^T \boldsymbol{K}(\boldsymbol{\rho}) \boldsymbol{u} \tag{3.30}$$

As when computing the continuous derivatives, the adjoint variable method is used to get rid of the computationally expensive term $\frac{\partial \boldsymbol{u}}{\partial \rho_e}$. Consequently, the adjoint problem is

$$\boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{\lambda} = -2 \cdot \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{u} \Leftrightarrow \boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{\lambda} = -2 \cdot \boldsymbol{f}(\boldsymbol{\rho}) \Leftrightarrow \boldsymbol{\lambda} = -2 \cdot \boldsymbol{u} \tag{3.31}$$

and the derivative is

$$\begin{aligned} \frac{\partial f(\boldsymbol{\rho})}{\partial \rho_e} &= \boldsymbol{u}^T \frac{\partial \boldsymbol{K}(\boldsymbol{\rho})}{\partial \rho_e} \boldsymbol{u} - 2 \cdot \boldsymbol{u}^T \frac{\partial \boldsymbol{K}(\boldsymbol{\rho})}{\partial \rho_e} \boldsymbol{u} + 2 \cdot \boldsymbol{u}^T \frac{\partial \boldsymbol{f}(\boldsymbol{\rho})}{\partial \rho_e} \\ &= -\boldsymbol{u}^T \frac{\partial \boldsymbol{K}(\boldsymbol{\rho})}{\partial \rho_e} \boldsymbol{u} + 2 \cdot \boldsymbol{u}^T \frac{\partial \boldsymbol{f}(\boldsymbol{\rho})}{\partial \rho_e} \end{aligned} \tag{3.32}$$

Using Equation 3.29 and the expression for the force vector, we get

$$\frac{\partial f(\boldsymbol{\rho})}{\partial \rho_e} = -\alpha \cdot \rho_e^{\alpha-1} \cdot \boldsymbol{u}^T \boldsymbol{K}_e \boldsymbol{u} + 2 \cdot V_e \cdot \sum_{c \in e} a_e \cdot \boldsymbol{u}_c^T \boldsymbol{g} \tag{3.33}$$

In practice, we will introduce a (triangle or tetrahedron shaped) hole with the size of the element. However, we assume that it is approximately equal to introducing an infitesimal round hole in the center of the element. Consequently, the derivatives are evaluated for $\boldsymbol{\rho} = \mathbf{1}$

$$\Delta_e^v f(\boldsymbol{m}) \approx \alpha \cdot \boldsymbol{u}^T \boldsymbol{K}_e(\boldsymbol{m}) \boldsymbol{u} - 2 \cdot V_e \cdot \sum_{c \in e} a_e \cdot \boldsymbol{u}_c^T \boldsymbol{g} \tag{3.34}$$

We will use $\alpha = 3$ *since, for this value of* α*, the density approach approximate the Hashin-Shtrikman bounds for compliance (Sigmund and Guest, 2014).*

Example 3.10 (Sum of von Mises stress)

Next, the objective is the sum of element von Mises stresses evaluated in their centroid

$$f(\boldsymbol{\rho}) = \sum_i \sigma_i(\boldsymbol{\rho}) = \sum_i \sqrt{\boldsymbol{u}_i^T \boldsymbol{B}_i^T \boldsymbol{E}_i(\boldsymbol{\rho}) \boldsymbol{A} \boldsymbol{E}_i(\boldsymbol{\rho}) \boldsymbol{B}_i \boldsymbol{u}_i} \tag{3.35}$$

Again, we use the adjoint method and get the adjoint problem

$$\boldsymbol{K}(\boldsymbol{\rho}) \boldsymbol{\lambda} = -\mathop{\mathbb{A}}_i \frac{1}{\sigma_i(\boldsymbol{\rho})} \cdot \boldsymbol{B}_i^T \boldsymbol{E}_i(\boldsymbol{\rho}) \boldsymbol{A} \boldsymbol{E}_i(\boldsymbol{\rho}) \boldsymbol{B}_i \boldsymbol{u}_i \tag{3.36}$$

and the derivatives

$$\begin{aligned} \frac{\partial f(\boldsymbol{\rho})}{\partial \rho_e} &= \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{K}(\boldsymbol{\rho})}{\partial \rho_e} \boldsymbol{u} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}(\boldsymbol{\rho})}{\partial \rho_e} + \sum_i \frac{1}{2 \cdot \sigma_i(\boldsymbol{\rho})} \cdot \\ &\left(\boldsymbol{u}_i^T \boldsymbol{B}_i^T \frac{\partial \boldsymbol{E}_i(\boldsymbol{\rho})}{\partial \rho_e} \boldsymbol{A} \boldsymbol{E}_i(\boldsymbol{\rho}) \boldsymbol{B}_i \boldsymbol{u}_i + \boldsymbol{u}_i^T \boldsymbol{B}_i^T \boldsymbol{E}_i(\boldsymbol{\rho}) \boldsymbol{A} \frac{\partial \boldsymbol{E}_i(\boldsymbol{\rho})}{\partial \rho_e} \boldsymbol{B}_i \boldsymbol{u}_i \right) \end{aligned} \tag{3.37}$$

Using equations 3.28 and 3.29 and the expression for the force vector, we get

$$\begin{aligned} \frac{\partial f(\boldsymbol{\rho})}{\partial \rho_e} =& \alpha \cdot \rho_e^{\alpha-1} \cdot \boldsymbol{\lambda}_e^T \boldsymbol{K}_e \boldsymbol{u}_e - V_e \cdot \sum_{c \in e} a_e \cdot \boldsymbol{\lambda}_c^T \boldsymbol{g} + \\ & \alpha \cdot \rho_e^{\alpha-1} \cdot \sigma_e(\boldsymbol{\rho})^{-1} \cdot \boldsymbol{u}_e^T \boldsymbol{B}_e^T \boldsymbol{E}_e \boldsymbol{A} \boldsymbol{E}_e \boldsymbol{B}_e \boldsymbol{u}_e \end{aligned} \tag{3.38}$$

Again, the expression is evaluated for $\boldsymbol{\rho} = \mathbf{1}$ *to approximate the discrete derivative*

$$\Delta_e^v f(\boldsymbol{m}) \approx -\alpha \cdot \boldsymbol{\lambda}_e^T \boldsymbol{K}_e(\boldsymbol{m}) \boldsymbol{u}_e + V_e \cdot \sum_{c \in e} a_e \cdot \boldsymbol{\lambda}_c^T \boldsymbol{g} - \alpha \cdot \sigma_e(\boldsymbol{m}) \tag{3.39}$$

Le et al. (2010) uses $\alpha = \frac{1}{2}$ for solving stress-constrained topology optimization problems using the standard density method. We will use the same value, however, note that if gravity is not taken into account, α reduces to a scaling of $\Delta_e^v f(\boldsymbol{m})$.

Example 3.11 (Maximal von Mises stress)

Finally, the p-norm of the element centroidal von Mises stresses is considered

$$f(\boldsymbol{\rho}) = \left(\sum_i \sigma_i(\boldsymbol{\rho})^p \right)^{\frac{1}{p}} \tag{3.40}$$

In this case, the adjoint problem is

$$\boldsymbol{K}(\boldsymbol{\rho})\boldsymbol{\lambda} = -f(\boldsymbol{\rho})^{1-p} \cdot \mathop{\mathbb{A}}_i \sigma_i(\boldsymbol{\rho})^{p-2} \cdot \boldsymbol{B}_i^T \boldsymbol{E}_i(\boldsymbol{\rho}) \boldsymbol{A} \boldsymbol{E}_i(\boldsymbol{\rho}) \boldsymbol{B}_i \boldsymbol{u}_i \tag{3.41}$$

and the derivative is

$$\begin{gathered} \frac{\partial f(\boldsymbol{\rho})}{\partial \rho_e} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{K}(\boldsymbol{\rho})}{\partial \rho_e} \boldsymbol{u} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}(\boldsymbol{\rho})}{\partial \rho_e} + f(\boldsymbol{\rho})^{1-p} \cdot \sum_i \frac{\sigma_i(\boldsymbol{\rho})^{p-2}}{2} \cdot \\ \left(\boldsymbol{u}_i^T \boldsymbol{B}_i^T \frac{\partial \boldsymbol{E}_i(\boldsymbol{\rho})}{\partial \rho_e} \boldsymbol{A} \boldsymbol{E}_i(\boldsymbol{\rho}) \boldsymbol{B}_i \boldsymbol{u}_i + \boldsymbol{u}_i^T \boldsymbol{B}_i^T \boldsymbol{E}_i(\boldsymbol{\rho}) \boldsymbol{A} \frac{\partial \boldsymbol{E}_i(\boldsymbol{\rho})}{\partial \rho_e} \boldsymbol{B}_i \boldsymbol{u}_i \right) \end{gathered} \tag{3.42}$$

Using equations 3.28 and 3.29 and the expression for the force vector, we get

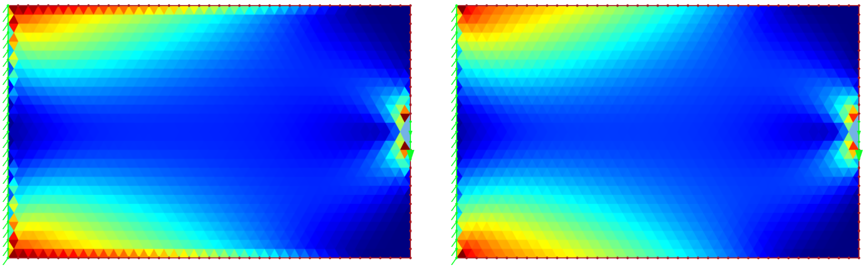
$$\begin{aligned} \frac{\partial f(\boldsymbol{\rho})}{\partial \rho_e} =& \alpha \cdot \rho_e^{\alpha-1} \cdot \boldsymbol{\lambda}_e^T \boldsymbol{K}_e \boldsymbol{u}_e - V_e \cdot \sum_{c \in e} a_e \cdot \boldsymbol{\lambda}_c^T \boldsymbol{g} \\ &+ \alpha \cdot \rho_e^{\alpha-1} \cdot f(\boldsymbol{\rho})^{1-p} \cdot \sigma_e(\boldsymbol{\rho})^{p-2} \cdot \boldsymbol{u}_e^T \boldsymbol{B}_e^T \boldsymbol{E}_e \boldsymbol{A} \boldsymbol{E}_e \boldsymbol{B}_e \boldsymbol{u}_e \end{aligned} \tag{3.43}$$

Finally, the derivative is evaluated for $\boldsymbol{\rho} = \boldsymbol{1}$ to get an approximation to the discrete derivative

$$\begin{aligned} \Delta_e^v f(\boldsymbol{m}) \approx & -\alpha \cdot \boldsymbol{\lambda}_e^T \boldsymbol{K}_e(\boldsymbol{m}) \boldsymbol{u}_e + V_e \cdot \sum_{c \in e} a_e \cdot \boldsymbol{\lambda}_c^T \boldsymbol{g} \\ & - \alpha \cdot f(\boldsymbol{m})^{1-p} \cdot \sigma_e(\boldsymbol{m})^p \end{aligned} \tag{3.44}$$

where we again use $\alpha = \frac{1}{2}$ as proposed by Le et al. (2010).

We will now investigate the validity of using these approximations to the discrete derivatives. However, note that it is not feasible to compute the true discrete derivatives and an approximation strategy is therefore needed.

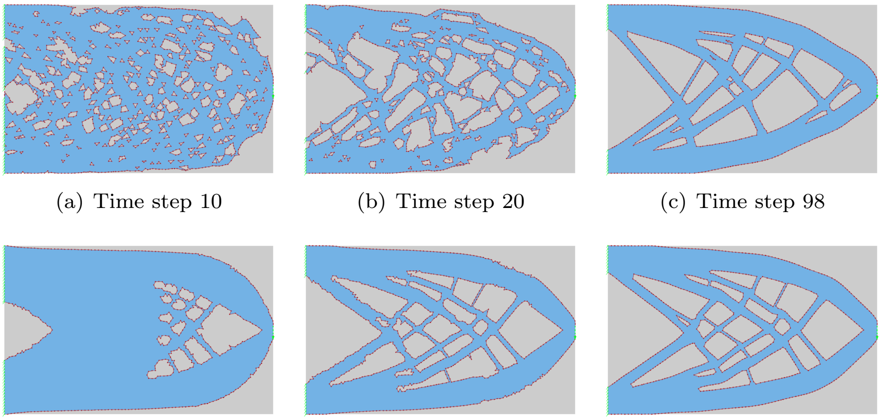


(a) True discrete derivatives (b) Approximate discrete derivatives

Figure 3.4: The true and the approximate discrete derivatives of compliance.

First, in Figure 3.4, we see that the difference between the true and the approximate discrete derivatives for compliance is subtle. Furthermore, the importance of high quality elements is evident due to the inaccuracies at the boundary. In that regard, note that the discretization in Figure 3.4 is the initial discretization before the DSC method improves the mesh quality.

Next, we investigate the effect of using random discrete derivatives versus approximate discrete derivatives for minimizing compliance. A few iterations and



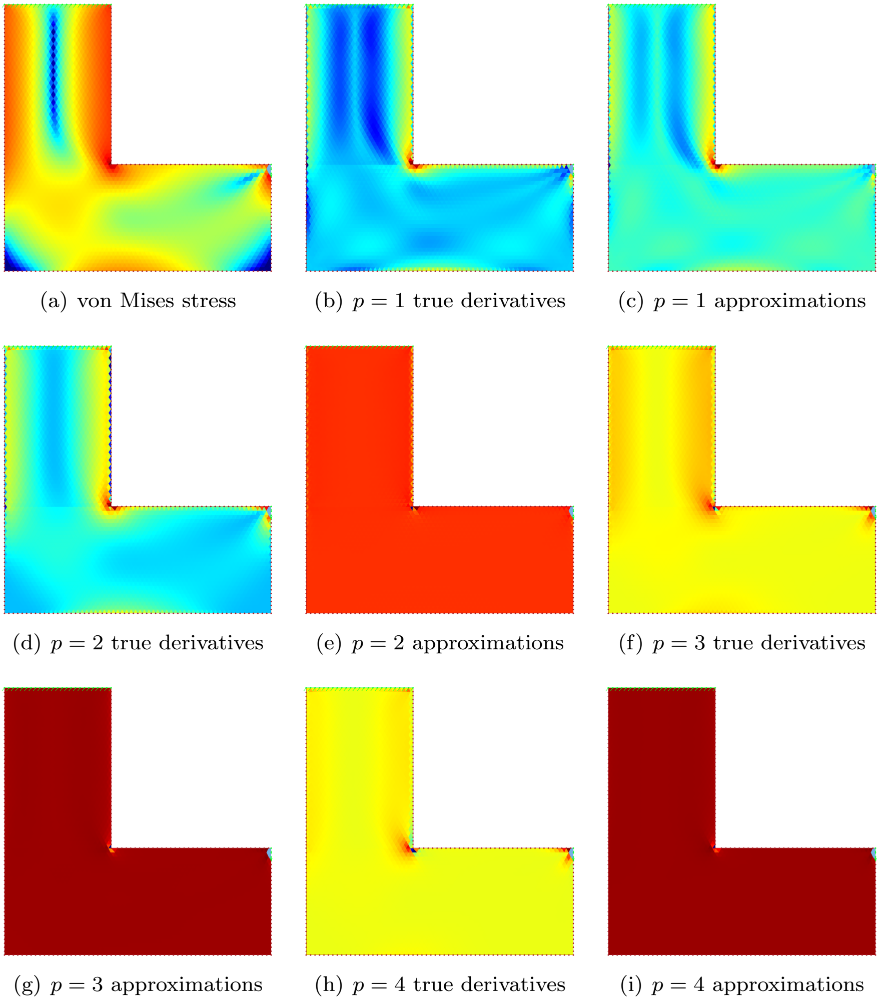
(a) Time step 10 (b) Time step 20 (c) Time step 98

(d) Time step 10 (e) Time step 20 (f) Time step 39

Figure 3.5: Optimization of compliance for the cantilever beam problem using random (top) and approximated (bottom) discrete derivatives.

the final designs for the two cases are seen in Figure 3.5. Using approximate discrete derivatives reduces the number of time steps from 98 to 39. Furthermore, compliance is 5535 for the random removal of elements versus 5550 when using approximate discrete derivatives. This implies that the continuous optimization step makes the optimization procedure converge to an optimum no matter which discrete derivatives are used. However, using accurate discrete derivatives increases the convergence rate significantly.

Finally, Figure 3.6 displays the approximate and true discrete derivatives of the p-norm von Mises stress objective together with the element von Mises stresses. It is evident that for $p = 1$ the approximations to the discrete derivatives are quite accurate. Both the true and approximated derivatives suggest to remove internal elements where the stress concentration is low. However, for $p = 3$ and $p = 4$, the approximations suggest to remove the element with the maximal stress and the true derivatives suggest to remove a few elements next to it that are also highly stressed. Drilling a small hole where the largest stress concentration resides reduces the maximal stress which explains this behavior. However, since we remove many finite sized elements at a time, using these derivatives will change the shape of the corner instead of introducing a small hole. Consequently, it will work against the continuous optimization, thereby having a negative effect on the convergence rate, and the final design will not contain any holes. Therefore, we concluded that we could not use these approximate derivatives in Paper C. Based on this and the experiment in Figure 3.5, we chose to use the element von Mises stresses as approximation to the discrete derivatives which turned out to work well. Another choice could be the approximate derivatives for $p = 1$ which is computationally more expensive since it requires us to solve an adjoint problem.


(a) von Mises stress (b) $p = 1$ true derivatives (c) $p = 1$ approximations

(d) $p = 2$ true derivatives (e) $p = 2$ approximations (f) $p = 3$ true derivatives

(g) $p = 3$ approximations (h) $p = 4$ true derivatives (i) $p = 4$ approximations

Figure 3.6: The true and the approximate discrete derivatives of the p-norm von Mises stress objective (using different values of p) together with the element von Mises stresses.

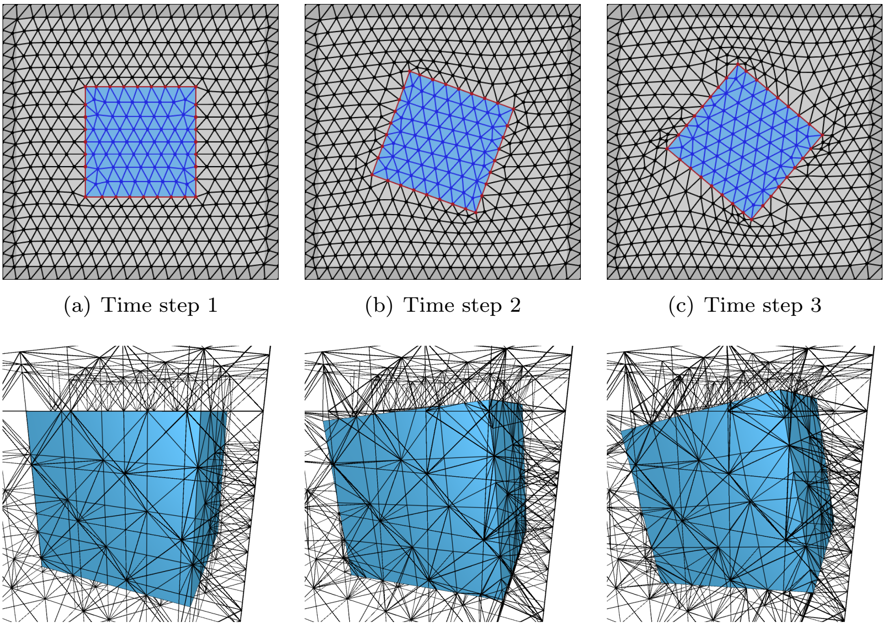
Chapter 4

Deformable Simplicial Complex method

We want to be able to deform the shape and topology of the explicitly represented structure as dictated by the optimization procedure in Chapter 3. Furthermore, since the accuracy of the FEA in Chapter 2 depends on the quality of the elements, it is important to sustain a high quality mesh during these deformations. Finally, it is advantageous to have local control of the mesh resolution since mesh adaptivity can be used to speed up the computationally expensive FEA. To be able to do all of this, we will use the Deformable Simplicial Complex (DSC) method.

In this chapter, the DSC method is described and its capabilities are illustrated by a few examples. Furthermore, the mesh operations used to improve the mesh quality, control the level of detail and accommodate topology changes are described. Finally, the strategies used by the DSC method to deform the shape and topology of an object are described and related to the optimization in Chapter 3.

The Deformable Simplicial Complex (DSC) method (Misztal et al., 2010a; Misztal and Bærentzen, 2012) has previously been used for fluid simulation (Misztal et al., 2010b, 2012) and is available as an open-source framework at `www.github.com/janba/2D-DSC` (2D) and `www.github.com/janba/DSC` (3D). The DSC method explicitly represents objects by embedding their surfaces (a set of line pieces in 2D and triangles in 3D) in a simplicial complex as described in Chapter 2.2. Furthermore, the simplicial complex discretizes both the inside and the outside of the objects into elements (triangles in 2D and tetrahedra in 3D). The DSC method is then able to deform the objects embedded in the simplicial complex without ruining the mesh by continuously applying a set of mesh operations. These mesh operations improve the quality of the elements, control the level of detail and accommodate topology changes while the objects are deformed. Note that the DSC method only applies the mesh operations where necessary (often near the surface) which means that it is much more efficient than a complete remeshing. A simple example of a deformation is rotation


(a) Time step 1 (b) Time step 2 (c) Time step 3

(d) Time step 1 (e) Time step 2 (f) Time step 3

Figure 4.1: Rotation of a square and cube using the 2D and 3D version of the DSC method respectively. The surface of the objects are depicted in red in 2D and turquoise in 3D. Furthermore, all edges of the simplicial complex are drawn in black or blue.

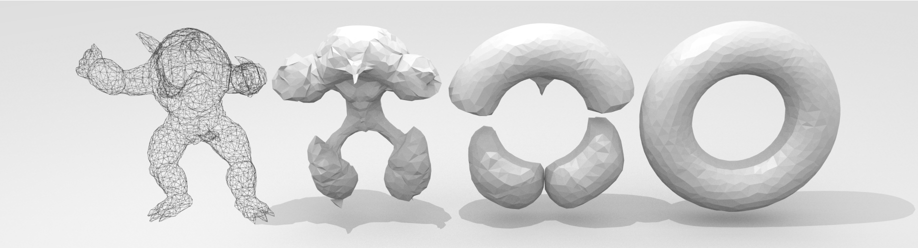


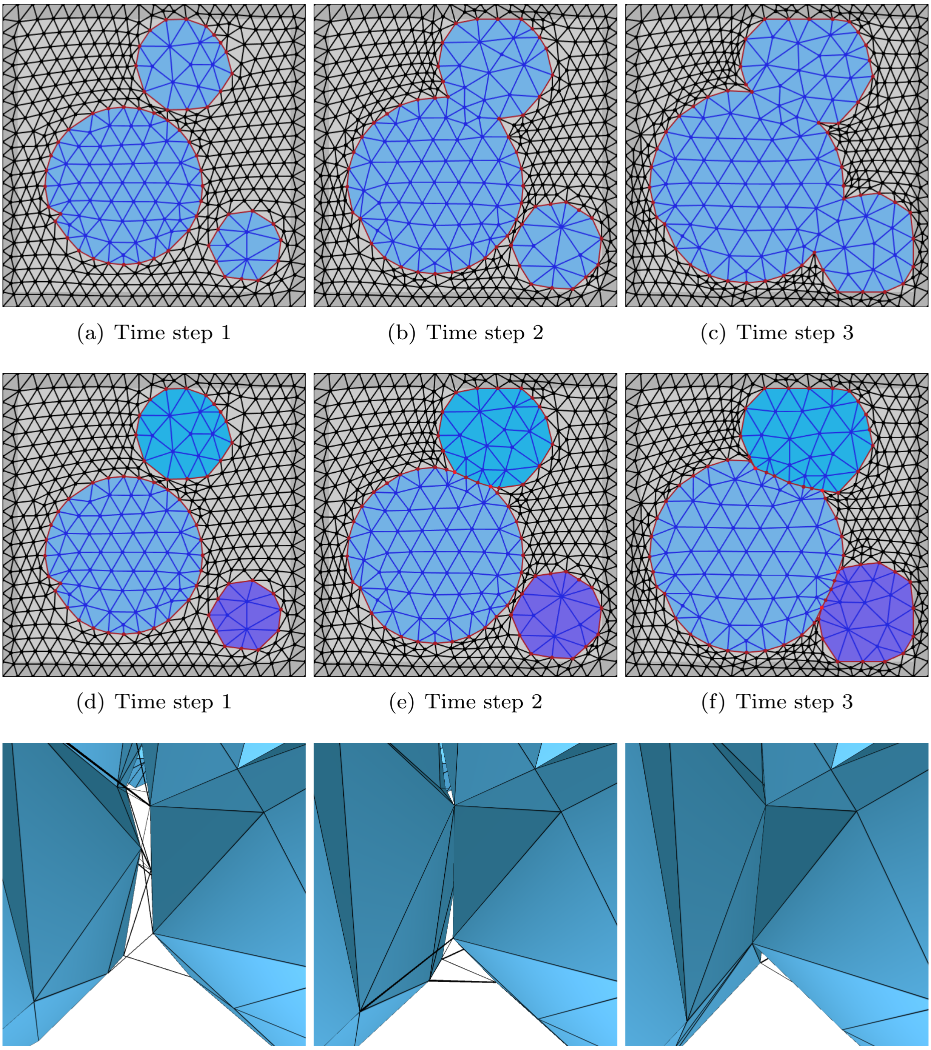
Figure 4.2: A morphing between an Armadillo model and a torus using the DSC method.

of a square in 2D and a cube in 3D as seen in Figure 4.1. A more complex deformation is the morphing from an Armadillo model to a torus in Figure 4.2.

Improving the quality of the elements is first and foremost necessary to ensure that the simplicial complex discretization is not ruined during the deformations. If the mesh contains inverted elements, the FEA would be invalid and the deformation strategies described in sections 4.3 and 4.4 would no longer function. Secondly, the accuracy of the FEA depends on the quality of the elements. Therefore, ensuring high quality elements also ensures an accurate FEA and thereby an accurate estimation of the structural properties.

The DSC method controls the mesh complexity – described by the discretization parameter δ corresponding to the average edge length – of both the surface and the mesh during the deformations. Consequently, we can increase the mesh complexity as the optimization converges or allow for smooth surface regions or regions away from the surface to be represented by a coarser discretization than surface regions with small scale features. This is used to decrease the number of elements, thereby decreasing the computation time of the FEA, while ensuring a certain level of detail.

Finally, the DSC method accommodate topology changes by removing low quality elements that are sandwiched between two surfaces, cf. Figure 4.3. As two objects approach each other the elements in between get squeezed and their quality decreases. The only option to improve such low quality elements is to alter the surfaces, e.g. by collapsing edges that connects the two objects. Now, the only thing separating the two objects is a set of line pieces/triangles. In the case where the objects have different labels, the line pieces/triangles are part of the surface and the objects stay separated. However, if the two objects have identical labels, these line pieces/triangles are no longer part of the surface and, as a consequence, the two objects are now merged into one.



(a) Time step 1 (b) Time step 2 (c) Time step 3

(d) Time step 1 (e) Time step 2 (f) Time step 3

(g) Time step 1 (h) Time step 2 (i) Time step 3

Figure 4.3: Collisions of objects with the same and different labels in 2D and 3D. In the 3D case, only edges where both end nodes are surface nodes are visualized.

The advantages of the DSC method include

Explicit representation
The surface of an object is represented explicitly and the object is discretized into triangular/tetrahedral elements by a simplicial complex. Consequently, a single mesh is used for both object representation and discretization.

Topology adaptivity
An object automatically changes topology whenever it collides with itself or another object with the same labels, or when elements residing inside the object is relabeled. Collision detection is inherent to the DSC method and it automatically performs the mesh changes which are necessary to change the topology.

Multiple phases support
Since the whole domain is explicitly represented, it is simple to represent multiple objects and multiple materials.

Multiple optimization strategies
It is possible to change the shape and topology of the structure both by moving nodes on the surface and by changing the label of elements.

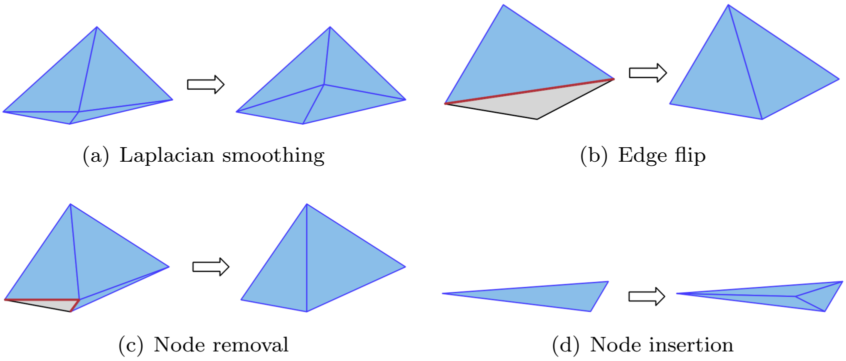
Scale adaptivity
Tiny details can be modeled efficiently since the scale of the surface representation can be adapted locally to fit these details. Furthermore, it is possible to change the resolution during deformations.

Little numerical diffusion
Sharp details are preserved for example while rotating an object.

4.1 2D mesh operations

The mesh operations that improve the mesh quality, control the level of detail and changes the topology in 2D are *smoothing*, *edge flip*, *node insertion* and *node removal*. Furthermore, the measure that determines the element quality is the minimal angle. If a degenerate triangle has one small angle, it is called a needle and if it has one large angle, it is called a cap.

Smoothing
Laplacian smoothing (Field, 1988) moves the nodes that are not part of the surface to the barycenter of its neighbors, cf. Figure 4.4(a). This is


(a) Laplacian smoothing

(b) Edge flip

(c) Node removal

(d) Node insertion

Figure 4.4: The mesh operations used by the 2D DSC method; Laplacian smoothing, edge flip, node insertion and node removal. The edge flip and node removal operations may change the surface (red) between elements with different labels (gray and blue).

a fast operation which improves the quality of a mesh significantly and is therefore widely used.

Edge flip

Edge flip (Bærentzen et al., 2012) flips the shared edge between two adjacent triangles, cf. Figure 4.4(b). Therefore, this is an operation which does not alter the positions of the nodes, only the connectivity. Edges that are not a part of the surface are flipped recursively to maximize the quality of the triangles. This procedure is computationally expensive since it can result in a vast amount of edge flips, however, it is essential to improve the mesh quality. Furthermore, edge flips are used to remove caps where the flipped edge is part of the surface. The two resulting triangles are both labeled according to the label of the largest triangle prior to the edge flip. Consequently, this changes the surface and may thereby change the topology.

Node removal

Node removal (Bærentzen et al., 2012) moves a node to the position of one of its neighbours and merges overlapping edges and triangles, cf. 4.4(c). If none of the other operations have been successful in improving the mesh quality, node removal is applied irrespective of whether or not the removed node is part of the surface. Consequently, this operation is not safe with respect to conserving the surface and therefore may change the topology. Furthermore, it is used for detail control of both the surface and the mesh by removing short surface edges and small triangles if the local mesh quality is not decreased too much.

Node insertion

Node insertion (Bærentzen et al., 2012) inserts a node at the barycenter of an edge or triangle, cf. Figure 4.4(d). It improves the mesh quality by subdividing needles which makes it possible to improve the quality of the three new triangles by other means e.g. edge flips. In addition, node insertion controls the level of detail of both the surface and the mesh by subdividing long surface edges and large triangles.

For detailed descriptions of these mesh operations, see for example Bærentzen et al. (2012) or Cheng et al. (2012). Furthermore, the 2D DSC method uses the mesh representation and the mesh operations implemented in the *GEL library* (Bærentzen et al., 2013).

4.2 3D mesh operations

The mesh operations that improves mesh quality, control the level of detail and changes the topology in 3D are *smoothing*, *edge removal*, *multi-face removal*, *node insertion* and *node removal*. Edge removal and multi-face removal apply the flips described in Figure 4.5 and therefore only change the connectivity. Furthermore, we will use the tetrahedral quality measure Q_e for element e defined by

$$Q_e = \frac{6 \cdot \sqrt{2} \cdot V_e}{(\frac{1}{6} \sum_{i \in e} l_i^2)^{3/2}} \tag{4.1}$$

(Parthasarathy et al., 1994) where V_e is the volume and l_i is the length of edge i adjacent to element e.

Smoothing

Smart Laplacian smoothing (Field, 1988) moves all nodes that are not a part of the surface to their barycenters. However, it is only applied if it improves the minimum quality of the tetrahedra locally or if the resulting minimum quality is larger than a user-defined threshold.

Edge removal

Edge removal (Shewchuk, 2002) removes an edge by performing a series of 2-3 flips followed by a single 3-2 flip. The series of flips that improves the local mesh quality the most are computed efficiently using the algorithm by Klincsek (1980). Edge removal can also be applied on the surface or on the boundary of the discretized domain where the final 3-2 flip is replaced

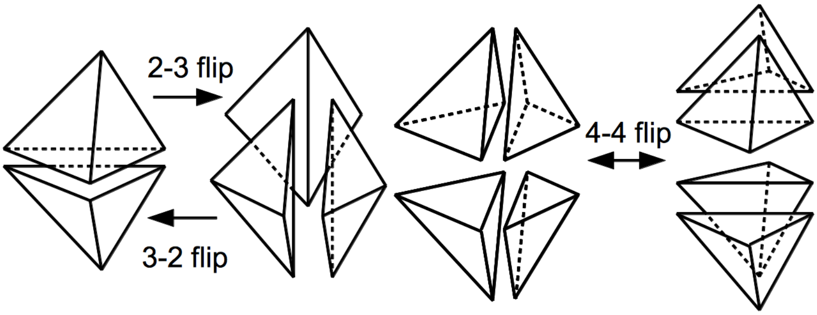


Figure 4.5: Illustrations of 2-3, 3-2 and 4-4 flips inspired by the illustration in Shewchuk (2002). The 2-3 flip replaces two adjacent tetrahedra with three and the 3-2 flip is the inverse operation. The 4-4 flip performs an edge flip of the shared edge between two triangles. This can only be performed on edges with four adjacent tetrahedra and when the flip does not result in any inverted tetrahedra. The 2-2 flip is similar to the 4-4 flip, but applies to edges on the boundary of the discretized domain.

by a 4-4 flip or 2-2 flip respectively. However, 4-4 flips are only applied on the surface when it is sufficiently flat.

Multi-face removal

Multi-face removal (Shewchuk, 2002) removes a set of triangles that are sandwiched between two nodes by performing a single 2-3 flip followed by a series of 3-2 flips. Again, the optimal series of flips with respect to local mesh quality are computed using the algorithm by Klincsek (1980). Multi-face removal is not applied to the surface or boundary since it would alter these.

Node removal

Node removal (Bærentzen et al., 2012) merges the two end nodes, which may reside on surfaces, of an edge if it does not result in any inverted tetrahedra. It is used to remove low quality tetrahedra whose quality cannot be improved by any other operation. Since node removal can deform surfaces, it may change the topology of objects. Furthermore, node removal controls the level of detail of the surface by collapsing small edges that reside on a flat region on the surface. It also controls the mesh complexity by collapsing an edge of large elements.

Node insertion

Node insertion (Bærentzen et al., 2012) introduces a node at the barycenter of an edge thereby splitting it in two. If a node removal operation cannot be performed because it inverts tetrahedra and one of the element edges are longer than average, this edge is split in two. This makes it possible for other operators to improve the quality of the new tetrahedra. Furthermore, node insertion is used to control the detail level of the surface by splitting long surface edges that reside in a region which is not flat. It is also used to control the mesh complexity by splitting edges of large tetrahedra.

See Misztal et al. (2010a) and Misztal and Bærentzen (2012) for more details. However, note that we have changed the DSC method in connection with this application. The multi-face retriangulation, optimization-based smoothing, null-space smoothing and tetrahedron relabeling operations have not been necessary. Removing these operations has simplified the DSC method as well as resulted in a significant speed-up. Furthermore, we have added the edge removal operation on the surface and boundary which often improves the mesh quality in situations where no other operation can.

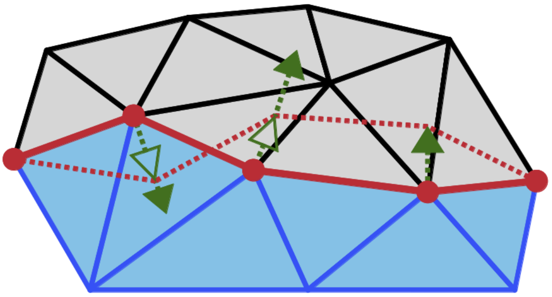


Figure 4.6: The node movement strategy in 2D which also applies to 3D. A filled arrow indicates the destination $\boldsymbol{p}_n^*$ of the surface node n (red). One node will create low quality elements and another will invert elements if moved to their destinations. These nodes are therefore only moved as depicted by the unfilled arrow while the last node is moved all the way to its destination. Then, mesh operations are applied to improve the mesh quality and the nodes that did not reach their destination are moved again. This is repeated until all nodes have reached their destinations.

4.3 Node movement strategy

The DSC method allows us to move surface nodes, thereby deforming the shape of objects. Furthermore, since the DSC method merges two objects (with identical labels) if they collide, moving surfaces nodes can change objects topology in addition to their shape. The node movement strategy is to compute a destination $\boldsymbol{p}_n^*$ for each surface node n currently at position $\boldsymbol{p}_n$. The destinations $\boldsymbol{p}_n^*$ are computed by a user-defined velocity function which, for the case of topology optimization, is described in Section 3.3.1. Afterwards, all surface nodes are moved from $\boldsymbol{p}_n$ to $\boldsymbol{p}_n^*$ using the strategy illustrated in Figure 4.6 and written in pseudocode in Figure 4.7.

```
while $\boldsymbol{p}_n \neq \boldsymbol{p}_n^*$ for any $n \in$ surface nodes do
    for all $n \in$ surface nodes do
        if $\boldsymbol{p}_n \neq \boldsymbol{p}_n^*$ then
            $\boldsymbol{p}_n^i$ = intersection($\boldsymbol{p}_n^* - \boldsymbol{p}_n$, $f \in$ faces)
            $l^{\max}$ = max(0.5$\cdot$ length($\boldsymbol{p}_n^i - \boldsymbol{p}_n$) $-d^{\min}$, 0)
            $l$ = min($l^{\max}$, length($\boldsymbol{p}_n^* - \boldsymbol{p}_n$))
            $\boldsymbol{p}_n = \boldsymbol{p}_n + l\cdot$ normalize($\boldsymbol{p}_n^* - \boldsymbol{p}_n$)
        end if
    end for
    improve_mesh_quality()
end while
perform_detail_control()
improve_mesh_quality()
```

Figure 4.7: The strategy for deforming the surface is to move each surface node n along a straight line from the current position $\boldsymbol{p}_n$ to the destination $\boldsymbol{p}_n^*$ without creating any low quality elements. Here, $d^{\min}$ determines the minimum distance from a node to any face which is not adjacent to the node. Subsequently, the quality of the mesh is improved through the mesh operations described in sections 4.1 and 4.2. These two steps are repeated until all surface nodes have reached their destinations. At that point, the appropriate detail level of the mesh is ensured and, finally, another round of quality improvement is performed.

4.4 Element relabeling strategy

In addition to moving surface nodes, the DSC method allows us to change the element labels which makes it possible to introduce new object and cavities inside existing objects. If the label of an element that is part of an object is changed, this element is no longer part of that object. If the element resides at the surface of the object, its shape changes in a discrete manner. However, if the element resides inside the object, a cavity is created and its topology is changed. The element relabeling strategy is to determine a new label m_e^* for each element e which in our case is computed by the discrete optimization step described in Section 3.3.2. The DSC method will then change the current label m_e to m_e^* while keeping track of the surface changes.

CHAPTER 5

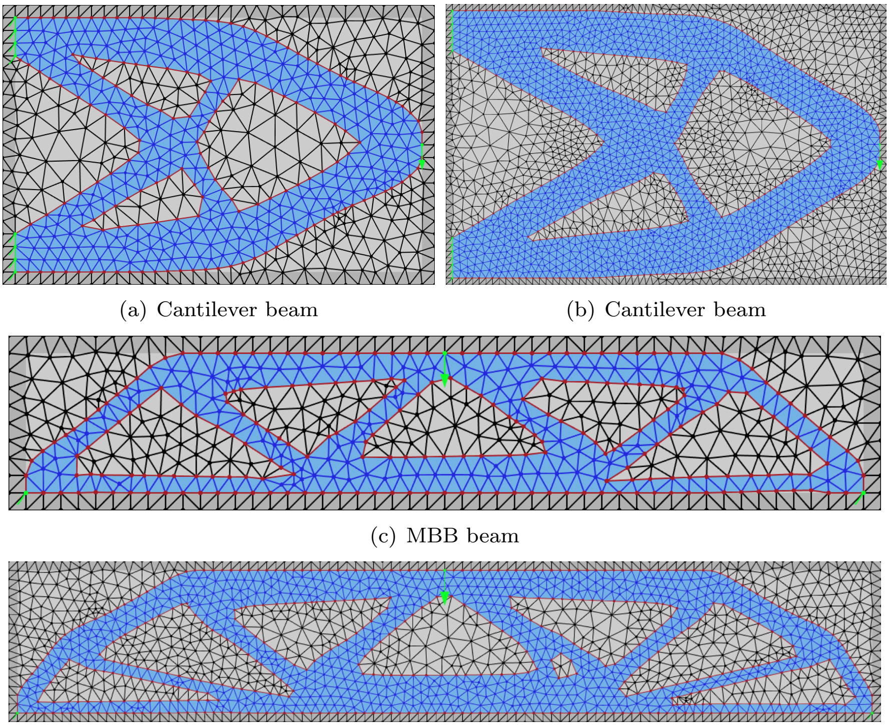
Results

In the following, the combined shape and topology optimization method is applied, however, this chapter will only contain a brief summary of the achieved results. For detailed results and how the method has been applied to specific problems, the reader is referred to the included papers in the appendix.

First, 2D mechanical problems, where the objective is to increase stiffness while limiting the material consumption, are solved. Next, we show that the method can solve real-world mechanical problems in 3D within reasonable time. Again, the objective is compliance subject to a volume constraint. Then, we minimize the maximal von Mises stress in 2D. This objective is more interesting than compliance since it penalizes regions with large stress concentrations that lead to structural fatigue. Finally, we show that the presented method can balance 3D models automatically.

The purpose of the first application (Paper A) is to show that the presented combined shape and topology optimization method works. Consequently, we apply the method to standard topology optimization problems in 2D; the cantilever beam (cf. Figure 2.2) and the MBB beam problems. Here, the objective is compliance subject to a volume constraint, i.e. the stiffness is maximized while the amount of material is limited. Considering the designs computed by other topology optimization methods to the same problems, Figure 5.1 shows that the method produces the designs that we expect. Furthermore, the resulting surfaces are smooth due to the explicit surface representation. Finally, when increasing the resolution, the detail level of the results increase and thin beams appear. We show in Paper A that this can be remedied by a perimeter constraint if thin beams are not desired for example because they are not manufacturable.

After we have shown that the method can solve shape and topology optimization



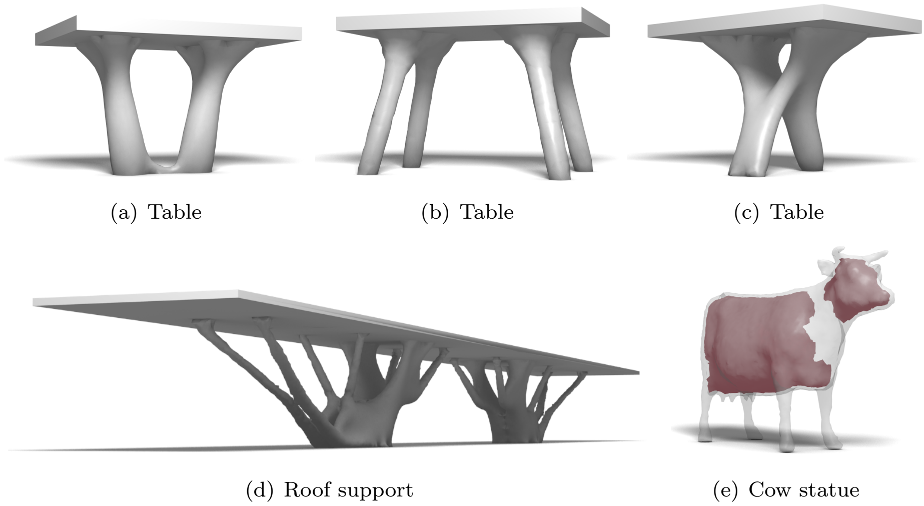
(a) Cantilever beam

(b) Cantilever beam

(c) MBB beam

(d) MBB beam

Figure 5.1: Shape and topology optimized designs in 2D from Paper A entitled 'Topology optimization using an explicit interface representation'.



(a) Table

(b) Table

(c) Table

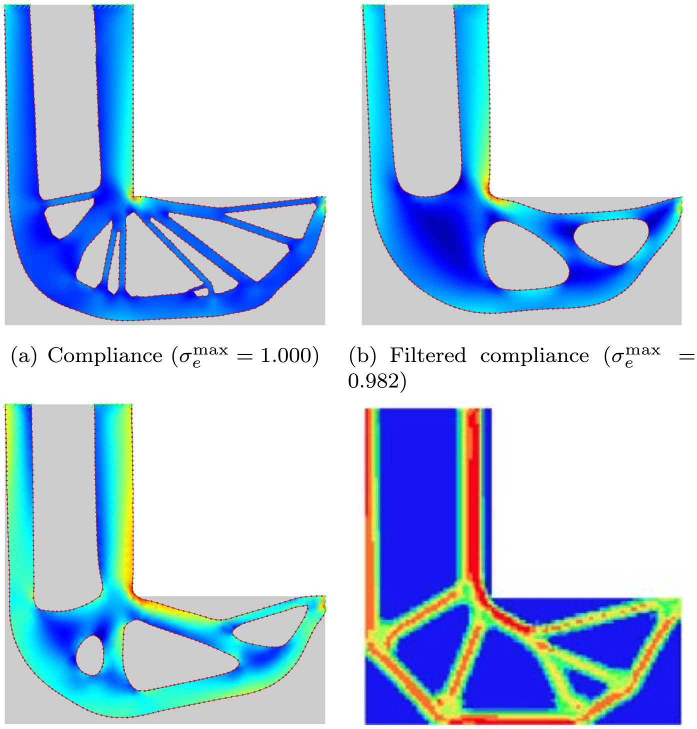
(d) Roof support

(e) Cow statue

Figure 5.2: Shape and topology optimized designs in 3D from Paper B entitled 'Combined shape and topology optimization of 3D structures'.

problems in 2D, it is natural to extend the approach to 3D (Paper B). Again, we want to maximize the stiffness of the structures and, consequently, the objective is compliance. Furthermore, several different constraints are applied, e.g. volume and perimeter constraints. Figure 5.2 shows a few real-world examples of 3D structures optimized by the presented approach. The table designs show that the approach can be used to create diverse functional designs by changing the boundary conditions slightly. The bridge (Figure 1.2) and roof support designs show that the approach can be used to find functional solutions to engineering problems. Finally, the cow statue shows that it is possible to initialize the optimization by a 3D model. In this case, we constrain the difference from the original model and the thickness of the shell of the structure. Consequently, the amount of material is limited and the stiffness is increased while it is ensured that the outer shape is changed only slightly. The biggest challenge when going from 2D to 3D is the time consumption. However, eliminating void elements from the FEA and taking advantage of the mesh adaptivity gives a huge performance boost. For example, the bridge in Figure 1.2 only took 68 minutes to compute on a laptop. Furthermore, the tables took between 13 and 20 minutes.

Until now, the objective has been compliance which optimizes the overall stiffness of the structure. Usually, we are not concerned with the overall stiffness but rather whether the structure will break or not. Consequently, the next application (Paper C) minimizes the maximal von Mises stress since large stress



(a) Compliance ($\sigma_e^{\max} = 1.000$)

(b) Filtered compliance ($\sigma_e^{\max} = 0.982$)

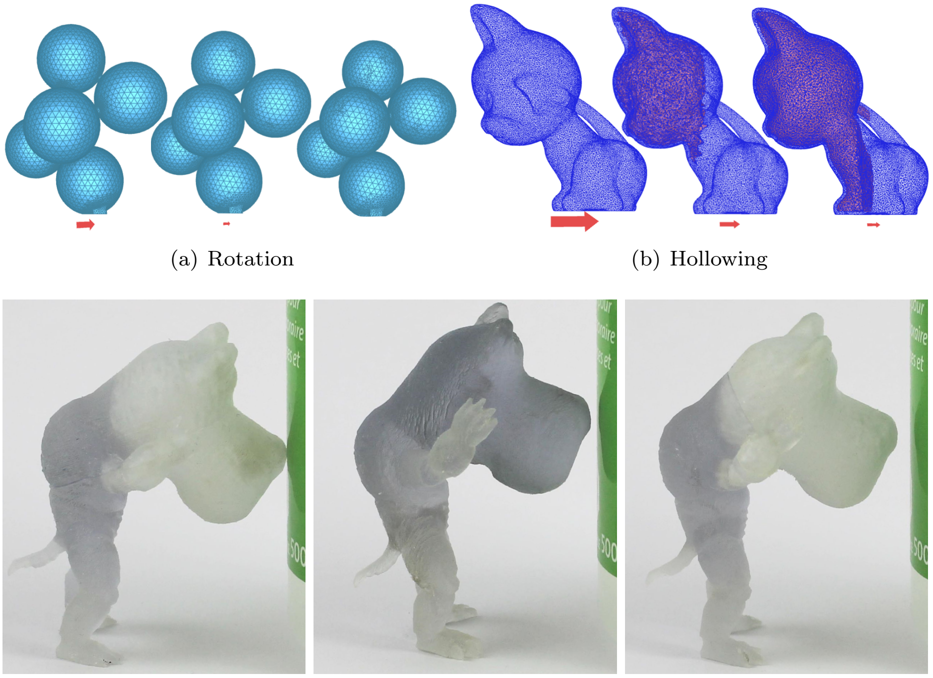
(c) Filtered p-norm stress ($\sigma_e^{\max} = 0.673$)

(d) Le et al. (2010)

Figure 5.3: Shape and topology optimized designs from Paper C entitled 'Combined shape and topology optimization for minimization of maximal von Mises stress'.

concentrations lead to structural fatigue. Of course the two objectives are related, however, they can lead to significantly different designs. Furthermore, the minimum stress problem is much harder to solve than the minimum compliance problem and it has caused trouble to other topology optimization methods. We also found that regularization, for example in the form of a Gaussian smoothing filter, is necessary to achieve smooth results. Figure 5.3 visualizes the von Mises stress across three optimized designs for the L-shaped cantilever beam problem together with a design by Le et al. (2010). It shows that using the maximal von Mises stress as objective instead of compliance leads to a more uniform distribution of stress and a significant reduction of the maximal von Mises stress $\sigma_e^{\max}$. It also shows that the presented designs are comparable to the design by Le et al. (2010) but with superior resolution. Furthermore, current methods, including the method by Le et al. (2010), minimizes volume or compliance subject to a p-norm stress constraint whereas we are able to minimize the p-norm stress subject to a volume constraint.

The final application (Paper D) is not concerned with the structural strength. Instead, the objective is to balance a 3D model subject to as few changes from the original model as possible. Here, two optimization procedures (illustrated in Figure 5.4) are applied. The first, hollowing, uses discrete and continuous optimization steps to create and deform internal cavities while the external surface is not allowed to deform. Consequently, this step does not change the outer shape of the model. The second procedure, rotation, uses the continuous optimization step and a change of variable to rotate the model around its base. During rotation, the parts of the model surface that are connected to the ground are not rotated and the base of the model is therefore deformed. The main advantage of this approach is illustrated by the balancing of an armadillo-dog



(a) Rotation

(b) Hollowing

(c) Hollowed model

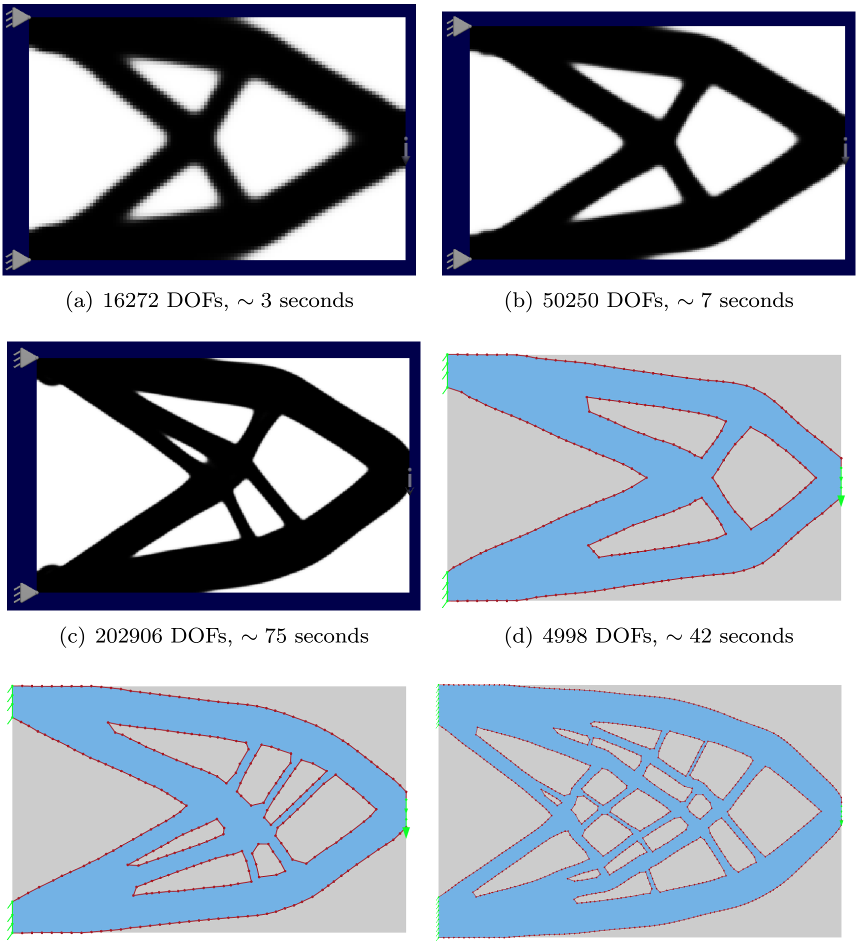
(d) Rotated model

(e) Hollowed and rotated model

Figure 5.4: Optimization strategies together with balanced and 3D printed designs from Paper D entitled 'Automatic balancing of 3D models'.

model in Figure 5.4. Hollowing does not deform the outer surface of the model and, consequently, it is to be preferred. However, hollowing is not always able to balance a 3D model. In these cases, a rotation, which deforms the external surface of the model slightly near its base, is applied. Furthermore, if rotation is preceded by hollowing, the needed rotation, and thereby the deformation of the external surface, is often small.

Finally, we present a small study which superficially compares the presented method to the popular density method in the form of the TopOpt App (Aage et al., 2013). This study is only meant as an indication of the speed and quality of designs for the presented method and an exhaustive study is therefore subject for future work. Comparing the quality of implicitly and explicitly represented designs is difficult since the implicit representation requires infinitely many elements to represent an inclined surface whereas the explicit can do with very few. However, it is evident in Figure 5.5 that infinitely many elements are not needed to get a very precise estimation of the optimized design. Nevertheless, the number of degrees of freedom (DOF) is significantly larger for the density method compared to the presented method. The DOFs equals the number of equations to solve in the FEA and the density method therefore solves much larger systems of equations than the proposed method. However, the density method is very fast per DOF by for example only assembling the global stiffness matrix once and by using a multiresolution approach (Aage et al., 2013). The density method is thereby generally faster in the presented cases, however, one should note that the presented method in 2D do not utilize multiple cores and the mesh adaptivity which are essential to achieve good performance in 3D. Consequently, one can expect a significant speed-up if focusing on improving performance of the 2D method. Furthermore, the scalability of the combined shape and topology optimization method seems to be better than for the density method. This suggests that the performance of the presented method may be better than for the density method in 3D, however, this is for a future study to show.



(a) 16272 DOFs, ∼ 3 seconds

(b) 50250 DOFs, ∼ 7 seconds

(c) 202906 DOFs, ∼ 75 seconds

(d) 4998 DOFs, ∼ 42 seconds

(e) 5150 DOFs, ∼ 107 seconds

(f) 19888 DOFs, ∼ 204 seconds

Figure 5.5: Designs produced by the density method and the combined shape and topology optimization method for the 2D cantilever beam problem.

Chapter 6

Conclusion and outlook

This dissertation introduces a novel method for combined shape and topology optimization. The presented method has several advantages where the most important is that it optimizes both the shape and topology of a structure that is explicitly represented.

In this chapter, the advantages and disadvantages of the method is summarized and an outlook is provided.

The novel method presented in this dissertation combines shape and topology optimization approaches into one unified framework. It does so by using the DSC method which represents the surface explicitly, yet, is able to accommodate large shape and topology changes.

Only one representation, a simplicial complex, is used to represent the structure and to discretize it into elements which are needed for the structural analysis. Using a single explicit representation implies that no conversions between representations are necessary. As a consequence, there is a simple relation between changes to the design variables and changes to the shape as opposed to when using implicit representations. Implicit representations have primarily been used because of their ability to handle topology changes, however, the introduction of the DSC method makes this argument superfluous.

Topology optimization is computationally very expensive and, consequently, speed is of essence. Here, the explicit surface representation is again an advantage because void elements can be eliminated from the structural analysis. Since the optimization is initialized by filling the design domain with material, this does not make a big difference in the first iterations of the optimization. However, as the optimization progresses and the volume of the structure is decreased, eliminating void elements from the FEA results in a significant speed-up.

Utilizing the mesh adaptivity of the DSC method by increasing the resolution as the optimization progresses will make the computational gain even more significant. In that case, a coarse discretization is used when the volume of the structure is large and a fine discretization when the volume has been reduced. Thereby, few elements are used for the FEA during the entire optimization compared to the resolution of the result. The mesh adaptivity can also be used to increase the resolution near the surface and decrease the resolution away from the surface. Again, this will increase the resolution of the optimized design with only a slight increase in computation time. However, the accuracy of the discrete optimization step depends on the size of the internal elements which limits the applicability of this strategy.

Finally, applying both a discrete and a continuous optimization step increases speed. The discrete step is fast but inaccurate whereas the continuous optimization step is accurate but slow. Consequently, the discrete step makes the structure converge fast to an approximate solution in the beginning of the optimization. At the end of the optimization, the continuous step ensures an accurate solution. Furthermore, it can close holes and collapse thin beams but it cannot introduce internal cavities. Therefore, the discrete step is essential to reach optimized designs containing holes.

We have shown in our superficial comparison in Chapter 5 that, due to these advantages, the speed of the presented method is comparable to the speed of the density method in 2D. This is even without increasing resolution as the optimization progresses which was shown to give a significant speed-up in 3D. Furthermore, the seemingly superior scalability of this method compared to the density approach suggests that it is better suited for topology optimization in 3D. The time consumption for computing the 3D designs in Paper B suggests the same. However, one should note that it is difficult to make a fair comparison of the quality between designs created by methods using an implicit and explicit representation. A meticulous study is therefore subject for future work.

Another subject for future work is to use geometric multi-grid, multi-scale or similar methods (Trottenberg et al., 2000; Vassilevski, 2008). These methods have been used as preconditioners for fixed-grid topology optimization methods with significant speed-up as a result (Amir et al., 2014; Aage et al., 2014; Nobel-Jørgensen et al., 2014). Applying multi-grid methods to a fixed grid is straight-forward, however, this is not the case for unstructured grids. Consequently, investigating how to apply them for this approach and how they perform is subject for future work. Note that using a multi-scale method does not necessarily result in a speed-up if producing meshes at different scales turns out to be too time consuming. In that case, the presented method has a disadvantage. On the other hand, using meshes which adapt to different scales over time is related to the idea of the multi-scale methods and might compensate for this.

Of course, the combined shape and topology optimization method has disadvantages. In addition to the complexity of applying multi-grid methods, the DSC method is difficult to implement, however, it is available as an open-source framework implemented in C++[1]. Furthermore, the presented method is new and, consequently, regularization schemes and the effect of parameters are still only investigated superficially. For example a minimum thickness constraint, which has been successfully applied to both the density and level set methods (Guest et al., 2004; Allaire et al., 2014), could be an interesting addition to the approach. Since we use an explicit shape representation, the local thickness can easily be computed which is not the case for the implicit methods. However, introducing a constraint on the minimum thickness may hinder topology changes by the continuous optimization and how to introduce this constraint is therefore not obvious.

The biggest drawback of the presented approach is the discrete optimization step. It is heuristic and therefore not guaranteed to converge though we base it on discrete derivatives. However, since it is combined with the continuous optimization, the optimization converges even when using random values for the discrete derivatives. Therefore, we are not that concerned with the heuristic na-

[1] Available at www.github.com/janba/2D-DSC (2D) and www.github.com/janba/DSC (3D)

ture of the discrete optimization. Comparing random and approximate discrete derivatives also showed that the convergence rate can be significantly improved by using accurate derivatives. However, using the true discrete derivatives is not an option since they are very expensive to compute. Furthermore, using the theory of topological derivatives to approximate the discrete derivatives has shown to be troublesome when minimizing the p-norm von Mises stress. Also, topological derivatives reflect the change in objective when creating a small round hole in the structure which is a crude approximation since the presented method removes an entire element of material. Consequently, we have approximated the discrete derivatives by simple expressions that are related to the topological derivatives. This has worked well in the presented applications, however, we must conclude that the discrete optimization step is subject for future work.

Even though the dissertation leaves a few possible improvements to future work, the combined shape and topology optimization method shows great promise. The approach has been shown to work well on mechanical problems in both 2D and 3D, and, already now, it can be used to solve mechanical problems in industry. However, these problems could also have been solved by other topology optimization methods though our superficial study shows superior scalability of the presented method. Furthermore, we have solved stress minimization problems which have been difficult using an implicit representation. However, to prove the advantage of using an explicit representation for topology optimization, the method should be applied to fluid, heat, electromagnetic or similar problems where it is of essence. If the method can be shown to produce new and interesting designs for these problems, the usefulness of the approach is evident. Another area where this approach has an advantage over previous methods is multi-material optimization which has only been treated briefly when balancing 3D structures. However, the only hurdle is to extend the discrete optimization step to compute the optimal element material between more than two materials. Consequently, this is an obvious area for future applications.

Finally, we consider the prospects of the DSC method. The DSC method has successfully been applied for fluid simulation (Misztal et al., 2010b, 2012) and now for topology optimization. In both cases, it has several advantages compared to implicit representations and simplified things compared to hybrid methods. Furthermore, an early investigation of its applicability to image segmentation has shown great promise (Dahl et al., 2014). Consequently, we find it safe to say that the DSC method can be applied with advantage to a wide variety of problems where virtual objects are represented and deformed. It is especially powerful for applications that need a discretization of the objects, for example to be able to use the finite element method.

APPENDIX A

Paper A

Christiansen, A. N., Nobel-Jørgensen, M., Aage, N., Sigmund, O., and Bærentzen, J. A. (2014b). Topology optimization using an explicit interface representation. *Structural and Multidisciplinary Optimization*, 49(3):387–399

Struct Multidisc Optim
DOI 10.1007/s00158-013-0983-9

RESEARCH PAPER

Topology optimization using an explicit interface representation

Asger Nyman Christiansen · Morten Nobel-Jørgensen · Niels Aage · Ole Sigmund · Jakob Andreas Bærentzen

Received: 26 March 2013 / Revised: 9 July 2013 / Accepted: 25 July 2013

Abstract We introduce the *Deformable Simplicial Complex* method to topology optimization as a way to represent the interface explicitly yet being able to handle topology changes. Topology changes are handled by a series of mesh operations, which also ensures a well-formed mesh. The same mesh is therefore used for both finite element calculations and shape representation. In addition, the approach unifies shape and topology optimization in a complementary optimization strategy. The shape is optimized on the basis of the gradient-based optimization algorithm MMA whereas holes are introduced using topological derivatives. The presented method is tested on two standard minimum compliance problems which demonstrates that it is both simple to apply, robust and efficient.

Keywords Topology optimization · Topological derivative · Non-parametric shape optimization · Explicit interface representation · Deformable simplicial complex method

A. N. Christiansen (✉) · M. Nobel-Jørgensen · J. A. Bærentzen
Department of Applied Mathematics and Computer Science,
Technical University of Denmark,
Matematiktorvet, B.303B, 2800 Kgs.
Lyngby, Denmark
e-mail: asny@dtu.dk

N. Aage · O. Sigmund
Department of Mechanical Engineering,
Technical University of Denmark,
Nils Koppels Allé, B.404, 2800 Kgs.
Lyngby, Denmark

1 Introduction

Current methods of topology optimization primarily represent the interface between solid and void implicitly on fixed grids. In contrast, shape optimization methods represent this interface explicitly, but do not allow for any topological changes to the structure. Using an explicit interface representation has a number of advantages. Consequently, we propose to adapt the *Deformable Simplicial Complex (DSC)* method (Misztal and Bærentzen 2012), which has been used to simulate fluids accurately (Misztal et al. 2010, 2012), to topology optimization.[1]

The DSC method represents the interface explicitly as one or more closed piecewise linear curves in 2D. In many cases, the explicit representation is necessary to be able to model a problem, for example flow or electromagnetic problems with localized boundary effects. Furthermore, local constraints, to control fillet radius at corners, interface smoothness, min/max length scale of the structure and so on, are possible to implement because of the explicit representation. Finally, and in all cases, the explicit interface is necessary when interpreting the final design.

As opposed to most other Lagrangian methods, such as pure shape optimization methods, the DSC method is able to handle topology changes. It does so by discretizing the entire design domain into a simplicial complex, i.e. an irregular triangle mesh. It hereby represents both the structure and the embedding space explicitly. Consequently, the interface is represented as piecewise linear curves between void and solid triangles. The adaptive mesh representation has been used previously in topology optimization (Eschenauer et al. 1994; Maute and Ramm 1995). However, the DSC

[1] Presented at the 10th World Congress on Structural and Multidisciplinary Optimization in 2013.

Published online: 21 August 2013

method adapts the mesh directly when changing the shape or topology instead of performing a remeshing at each iteration.

In recent years, using a Lagrangian mesh in combination with the Eulerian level set method (Osher and Fedkiw 2002) has gained popularity (Ha and Cho 2008; Allaire et al. 2011, 2013 Yamasaki et al. 2011; Xia et al. 2012). Here, the topology is evolved using the level set method and, again, the design domain including the interface is remeshed after each iteration. The purpose of the level set method is to be able to handle topology changes. Since the DSC method can handle topology changes, we avoid dealing with and switching between the implicit and explicit representations. Instead, we can do with just the explicit. However, the DSC method and the level set method have many similarities. Consequently, many of the techniques used in connection with the level set method (Allaire et al. 2004; Wang et al. 2003) can also be used in connection with the DSC method.

Another advantage of the DSC method is that the triangle mesh can be exploited for finite element (FE) analysis. The solid triangles define the structure and their deformation is described by second order shape functions. To increase performance, degrees of freedom associated with void triangles can be eliminated from the FE equations. Using the triangle mesh for analysis as well as shape representation is possible since the DSC method ensures a mesh with no degenerate elements. If the mesh contained degenerate or close to degenerate elements, the analysis would break down and the results would no longer be valid. The DSC method solves this issue by a series of mesh operations, which keeps the mesh well-formed. Put another way, the benefit of using a well-formed adaptive mesh is that the representation for the FE analysis and the shape of the structure can be one and the same.

In addition to unifying analysis and shape representation, the approach also combines shape and topology optimization. Consequently, the method consists of the three steps:

Step 1: *Topology optimization* Introduces holes using topological derivatives (Eschenauer et al. 1994; Sokolowski and Zochowski 1999; Feijóo et al. 2003), which are calculated directly on the triangle mesh.

Step 2: *Shape optimization* Performs a non-parametric shape optimization (Le et al. 2011; Arnout et al. 2012; Ding 1986; Mohammadi and Pironneau 2001) on the basis of the gradient-based optimization algorithm MMA (Svanberg 1987). This step calculates an improved shape, which is within a small perturbation of the current shape.

Step 3: *DSC deformation* Deforms the interface to the improved shape estimated in step 2 using DSC (Misztal and Bærentzen 2012). While the interface is deformed, the mesh is adapted such that it is well-formed at all times.

These three steps are iterated until convergence.

Previous work has sought to combine shape and topology optimization e.g. Eschenauer et al. (1994), Bletzinger and Maute (1997) and Kim et al. (2008). However, our approach is quite different from previous methods. It follows a similar paradigm in the sense that all of these methods iteratively introduce holes and optimize the shape. Yet, in the presented approach, the shape and topology optimization steps are highly interdependent. Our non-parametric shape optimization in combination with the DSC deformation can (and does) change the topology by closing holes. Moreover, the topology optimization step removes material on the boundary, thereby speeding up the shape optimization process. Furthermore, the presented approach distinguishes itself from its predecessors on several points. We do not perform a remeshing at each time step as opposed to the bubble method (Eschenauer et al. 1994). We use a single representation when performing shape and topology optimization as opposed to, for example, the method by Bletzinger and Maute (1997). Moreover, we perform a non-parametric shape optimization instead of the more complex parametric shape optimization used by both Eschenauer et al. (1994), Bletzinger and Maute (1997) and Kim et al. (2008).

The presented approach is efficient. This is because the FE analysis is performed only on solid triangles, the gradients are calculated only for the interface vertices and the beneficial interaction between the shape and topology optimization steps. The method is also easy to use, since no starting guess is required and only a few natural parameters have to be set. Finally, the method shows promising results. To demonstrate this, we solve two standard topology optimization problems where compliance is minimized subject to a volume constraint. In this context, we conduct a small parameter study to demonstrate the robustness of the method. These results are presented in Section 3 and discussed in Section 4. The method is described in the following section.

2 Method

2.1 Deformable simplicial complex method

The ability to virtually track deformable interfaces has many applications and is therefore of great interest. However, Eulerian methods, such as the level set method (Osher and Fedkiw 2002), tend to suffer from numerical diffusion and Lagrangian methods are mostly not able to handle topology changes. This was the reason for developing a new method, the DSC method (Misztal and Bærentzen 2012). This method has recently been applied for fluid simulations and the results are promising (Misztal et al. 2010, 2012).

The DSC method is a pure Lagrangian method, i.e. it represents the interface explicitly as one or more piecewise linear curves in 2D. Furthermore, it is able to handle topology changes naturally by discretizing the embedding space into a simplicial complex as illustrated in Fig. 1. This means that the domain is divided into triangles, such that every point in the embedding space is encapsulated in the interior of exactly one triangle, lies on the boundary between triangles or is a vertex of at least one triangle. The discretized domain is the design domain and a border surrounding the design domain (marked light and dark gray respectively in Fig. 1). The border is convenient, since it allows objects to extend to the boundary of the design domain without any mesh complications. One could resolve these mesh complications with the same operations used for solving the complications at the interface. However, it is more efficient to add a boundary. Furthermore, all triangles are marked as either non-void (solid in our case) or void. The interface is therefore the set of line segments which have a non-void triangle on one side and a void triangle on the other. The same concept can be (and has been, cf. e.g. Misztal and Bærentzen 2012) transferred to 3D, as seen in Fig. 1. However, we will only deal with the 2D case here and leave the 3D case for future work.

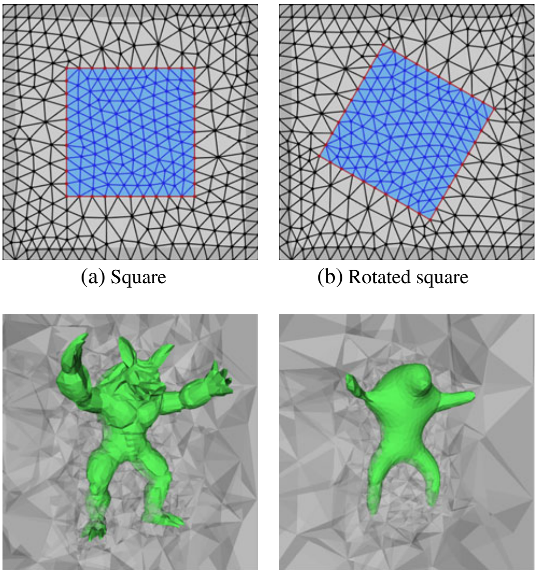
The goal is to be able to deform the interface while keeping the mesh well-formed at all times. The interface is deformed on the basis of a velocity function, for example rotation or smoothening as seen in Fig. 1. A velocity function determines a new position $\mathbf{p}_v^{t+1}$ for each of the interface vertices v at each time step t. The only limitation on the new positions is that they have to stay within the design domain.

When the positions $\mathbf{p}_v^{t+1}$ have been determined, each of the interface vertices are moved one at a time in a straight line from their old position $\mathbf{p}_v^t$ to $\mathbf{p}_v^{t+1}$ as illustrated by the filled arrows in Fig. 2. If moving a vertex v results in a degenerate triangle, we stop the movement before the triangle becomes degenerate as illustrated by the unfilled arrow in Fig. 2. After all interface vertices have been moved as far as possible, a series of mesh operations are performed to re-establish a well-formed mesh. Again, we move the interface vertices which have not reached their destination as far as possible. This procedure is repeated until all interface vertices have reached their new position $\mathbf{p}_v^{t+1}$.

To achieve a well-formed mesh, but also to be able to change topology, we need to perform a series of mesh operations. Each of the mesh operations has the purpose to complete one or more of the following tasks.

Quality control Ensures the quality of the triangles in the mesh. The quality of a triangle is determined by its angles, edge lengths and area.

Detail control Keeps the detail of the mesh at an appropriate level, i.e. such that details


(a) Square

(b) Rotated square

(c) Armadillo

(d) Smoothed armadillo

Fig. 1 This figure depicts a rotation of a square in 2D and a smoothening of an armadillo model in 3D using DSC. DSC ensures a well-formed mesh at all times during these deformations. The square in 2D is defined by the interface (*red*) between non-void (*blue*) and void (*gray*) triangles. In 3D, the interface is depicted in *green*

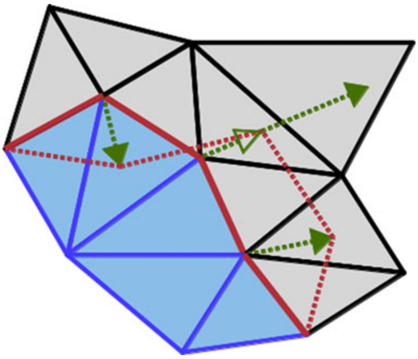


Fig. 2 The interface is deformed by, at each time step, moving each of the interface vertices to a new position (indicated by the *filled arrows*). If moving a vertex causes the mesh to degenerate, the vertex is moved as far as possible (indicated by the *unfilled arrow*) before the quality of the mesh is improved

are preserved but the amount of computations is minimized. The appropriate detail level of a triangle is determined by its edge lengths and area.

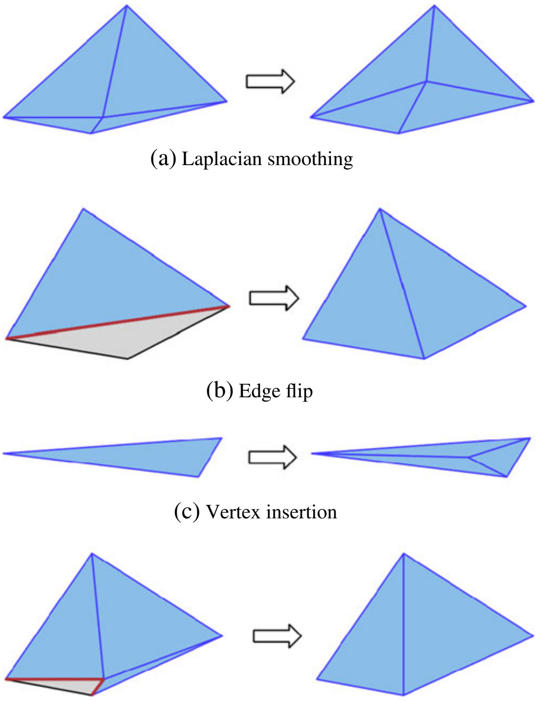
Topology changes Occurs when two interfaces meet and cause mesh configurations which can be impossible to disentangle with the operations used for quality and detail control.

The minimum angle is set to 30°. Furthermore, we want edges to be δ long on average. Therefore, the δ parameter determines the min/max distance between interface vertices to $0.1 \cdot \delta$ units and $1.9 \cdot \delta$ units respectively. However, it also determines the min/max area of the triangles to $0.1 \cdot \frac{\sqrt{3/4}}{2} \cdot \delta^2$ units^2 and $1.9 \cdot \frac{\sqrt{3/4}}{2} \cdot \delta^2$ units^2 respectively. Furthermore, a triangle is considered degenerate when an edge length, the area or an angle is less than half the minimum edge length, minimum area or minimum angle respectively.

The mesh operations used to handle the tasks are mentioned in the following and described in detail in e.g. Bærentzen et al. (2012) and Cheng et al. (2012).

Smoothing The non-interface part of the mesh is smoothed using *Laplacian smoothing* which moves each vertex to the barycenter of its neighbors (see Fig. 3a). This is a simple and fast operation which improves the quality of a mesh significantly and is therefore widely used.

Edge flip The *edge flip* operation is a topological operation which does not alter the position of the vertices but only the connectivity. A topological operation can be pictured as picking a set of adjacent triangles and replacing them with another set of triangles that fills out the same volume. The edge flip operation flips the shared edge between two neighboring triangles. In the DSC implementation, the edge flip operation is used for two things. Firstly, non-interface edges are flipped recursively to maximize the minimum angle of the triangles. This is a computationally heavy optimization procedure since it can result in a vast amount of edge flips. However, it is essential to be able to ensure a high quality mesh. Secondly, edge flips are used to remove caps, which are triangles with one large angle as seen in Fig. 3b. This is primarily necessary for handling topology changes and the flipped edge can therefore be an interface edge. The two resulting triangles are then both labelled either void or non-void, according to the label of the largest triangle prior to the edge flip.


(a) Laplacian smoothing

(b) Edge flip

(c) Vertex insertion

(d) Vertex removal

Fig. 3 Illustrations of the four 2D mesh operations used by the DSC method; Laplacian smoothing, edge flip, vertex insertion and vertex removal. *Gray triangles* represent void whereas blue triangles represent non-void. Therefore, the figures also illustrate that the edge flip and vertex removal can be used to handle topological changes

Vertex insertion This operation inserts a vertex at the barycenter of a triangle, thereby dividing it into a new set of triangles as seen in Fig. 3c. Vertex insertion is used to improve the mesh quality by subdividing needles, i.e. triangles with one very small angle. Subdividing a needle makes it possible to improve the quality of the three new triangles by other means e.g. the edge flip operator. In addition, vertex insertion is used to control the level of detail. This means, since we want to impose an upper limit on the size of the triangles, we subdivide too large triangles. Also, we want to control the detail level of the interface by limiting the length of the interface edges. Consequently, a vertex is inserted in the middle of the interface edges that are too long and the neighboring triangles are split in two.

Vertex removal Vertex removal is used for detail control in a similar manner as vertex insertion. If a triangle is too small or an interface edge is too short and the resulting triangles does not have too small an angle, a vertex is removed. The procedure to remove an unwanted vertex is to collapse an edge, i.e. moving the unwanted vertex to the position of its neighbour and merging overlapping edges and triangles. However, vertex removal is not just used for detail control but also quality control and to handle topology changes as illustrated in Fig. 3d. To elaborate, it is used if the previously described operations have not improved the quality of a triangle to a certain level, irrespective of whether or not the edge is part of the interface. Consequently, this operation is not safe with respect to conserving the interface and is used only as the last resort. However, since it can change the interface, vertex removal also handles topology changes where the other operations, except for the flip edge operation, cannot improve the mesh quality.

The DSC method uses the mesh representation and the mesh operations implemented in the *GEL library* (Bærentzen et al. 2013).

2.2 Discretization

For the linear elastic analysis, we use second order basis functions or Linear Strain Triangles (LST). This decision is based on a comparison between Constant Strain Triangles (CST) and LST which shows that LST have distinct advantages in spite of an increased computation time. When applying CST, we sometimes experience that a non-smooth interface is perceived as optimal. This is a well-known problem (Ding 1986; Mohammadi and Pironneau 2001) and several solutions exists for example applying a smoothness constraint on the move limits (Le et al. 2011) or a global perimeter constraint (Ambrosio and Buttazzo 1993). However, when using LST, we have not experienced such problems which implies that LST ensures a smooth result for the compliance objective considered.

Consequently, we associate six nodes with each triangular element, one at each vertex and one at the center of each edge. Furthermore, each node has two degrees of freedom, namely in the vertical and horizontal directions. In total, we have N_e solid elements indexed by e, the N_n associated nodes indexed by n and $N_d = 2 \cdot N_n$ degrees of freedom indexed by d. Note that only triangles which are a part of the structure, not the void triangles, are part of the N_e elements. DSC ensures that the triangular elements are well-formed at all times. In worst case, the minimum angle of an element is 15° and in general not less than 30°. Consequently, this discretization (depicted in Fig. 4) can be used to both represent the shape and for the analysis.

2.3 Optimization

The optimization procedure consists of two primary steps, namely shape and topology optimization. The shape optimization step calculates an improved position $\mathbf{p}_v^*$ for each non-fixed interface vertex v which are marked red in Fig. 4. Non-fixed means that the vertex is not supported and no load is applied to it. The positions $\mathbf{p}_v^*$ are determined by a gradient-based optimization algorithm as described in Section 2.3.1. The actual deformation of the interface is then handled by the DSC method. The topology optimization part changes the label of elements from solid to void according to the topological derivative, as described in Section 2.3.2. This is essential to be able to reach solutions containing holes. Furthermore, this allows the structure to let go of the supports in cases where this might be beneficial.

In addition to the fundamental abilities of the shape and topology optimization steps, the shape optimization

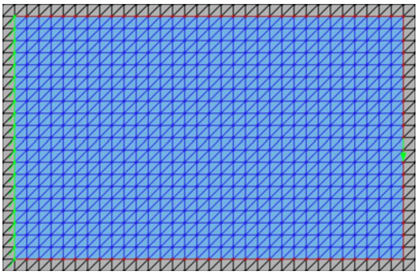


Fig. 4 This figure depicts the 500 × 800 units^2 Cantilever beam problem discretized by DSC. The design domain is completely filled with material in the form of blue triangular elements. Furthermore the internal vertices and edges are blue, whereas the non-fixed interface vertices and edges are red. The bright green points illustrate the supported vertices and a single load is illustrated by a bright green arrow. Both supported vertices and the vertices where a load is applied are fixed and cannot move

can change the topology and the topology optimization can change the shape. To elaborate, the topology optimization step moves the interface in discrete steps by switching triangles incident on the interface from solid to void. This improves convergence speed. On the other hand, the shape optimization step together with the DSC deformation step closes non-optimal holes created by the topology optimization step. It does so by making the hole so small that the triangles become degenerate and are removed by the DSC method. Therefore the two parts of the optimization procedure complement each other.

We will test the proposed method on examples in which compliance is minimized. Consequently, the discrete version of the objective function used for the optimization takes the form

$$\phi(\mathbf{u},\mathbf{x}) = \mathbf{u}^T\mathbf{K}(\mathbf{x})\mathbf{u} \tag{1}$$

Here $\mathbf{K}(\mathbf{x})$ is the global *stiffness matrix* and $\mathbf{x}$ is the *design variables*. Furthermore, $\mathbf{u}$ is the global *displacement vector* of length N_d which is found by solving the equilibrium equation

$$\mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f} \tag{2}$$

The global *load vector* f consist of zeros except for the degrees of freedom where a load is applied. The sparse solver *CHOLMOD* (Chen et al. 2008),[2] which is a part of the *SuiteSparse* library (Davis et al. 2013), is used to solve the equilibrium equation efficiently.

2.3.1 Shape optimization

The design variables $\mathbf{x}$ are the parameters we change to estimate an optimal solution $\mathbf{x}^*$ at time step t. Since we seek to optimize the shape, the design variables are associated with the positions of the N_v non-fixed interface vertices. Here N_v is always smaller (and in general much smaller) than N_d. Each vertex v has two degrees of freedom. However, we only search for the optimum in the normal direction $\mathbf{n}_v$ to an interface vertex v. Here, $\mathbf{n}_v$ is calculated by averaging the normals of the interface edges which is connected to v (Bærentzen et al. 2012). The idea is that moving v along the interface will not change the shape of the structure and is therefore not necessary (Sokołowski and Zolésio 1992). Consequently, the vector $\mathbf{x}$ is the collection of one design variable x_v for each non-fixed interface node $v \in [1,\ldots,N_v]$. The relation between the current position $\mathbf{p}_v$, the optimized position $\mathbf{p}_v^*$ and the optimized design variable x_v^* for a non-fixed interface vertex v is given by

$$\mathbf{p}_v^* = \mathbf{p}_v + x_v^*\mathbf{n}_v \tag{3}$$

Note that only non-fixed interface vertices are moved during the shape optimization step. When x^* is determined, the actual moving of v from $\mathbf{p}_v$ to $\mathbf{p}_v^*$ is handled by the DSC method. This is done in the same manner as moving the interface vertices according to any other velocity function. Vertices are therefore possibly moved, removed or added during this DSC deformation step. A shape optimization step and subsequent DSC deformation step can be seen in Fig. 5.

To estimate $\mathbf{x}^*$, we have to solve a smooth non-linear optimization problem of the type:

$$\begin{aligned}\mathbf{x}^* = \arg\min_x \; &: \phi(\mathbf{u},\mathbf{x}) = \mathbf{u}^T\mathbf{K}(\mathbf{x})\mathbf{u}\\ subject\ to \; &: g_m(\mathbf{x}) \le 0,\ m = 1,\ldots,q\\ &: \mathbf{K}(\mathbf{x})\mathbf{u} = \mathbf{f}\\ &: \mathbf{x}^{\min} \le \mathbf{x} \le \mathbf{x}^{\max}\end{aligned} \tag{4}$$

Here, the functions $g_m(\mathbf{x})$ are normalized global constraints on the properties of the structure. Since the objective is to minimize compliance, we will impose a constraint on the maximum volume of material V^* in the final solution. Specifically, the volume limit V^* will be a fraction of the volume of the entire design domain such that $g_1(\mathbf{x}) = \frac{V(\mathbf{x})}{V^*} - 1$ where $V(\mathbf{x})$ is the fraction of the design domain currently filled with material.

Enforcing a small volume constraint when the entire design domain is filled with material will force the algorithm to rapidly go towards a basic feasible solution. Thereby the algorithm is likely to end up in a non-optimal or invalid structure. We will therefore gradually decrease the volume limit V^t by 0.025 each time step until $V^t = V^*$. Consequently, the update rule is

$$V^{t+1} = \max(V^t - 0.025, V^*) \tag{5}$$

where $V^0 = 1$. Hereby, the volume constraint at time step t is

$$g_1(\mathbf{x}) = \frac{V(\mathbf{x})}{V^t} - 1 \tag{6}$$

The move limits $x_v^{\min}$ and $x_v^{\max}$ are determined by the design domain which ensures that the interface vertices stay inside the design domain. Furthermore, the move limits are affected by the discretization and the state of the mesh in the neighborhood of each interface vertex. The optimization algorithm needs to evaluate the objective function for different values of $\mathbf{x}$. Therefore, the limitation on the state of the mesh ensures that no elements are close to being degenerate when evaluating the objective function in the optimization process. This is controlled by setting the minimum distance μ between the interface node and any edge which is not

[2] CHOLMOD is the default solver for sparse symmetric positive definite linear systems in MATLAB.

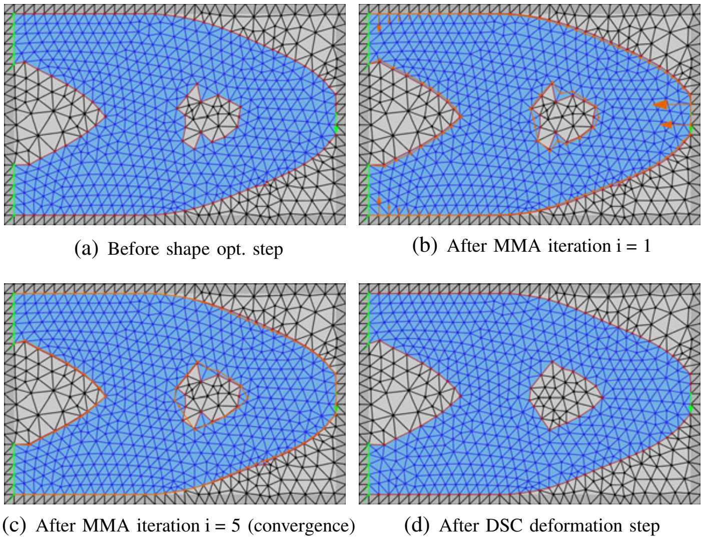

(a) Before shape opt. step
(b) After MMA iteration i = 1
(c) After MMA iteration i = 5 (convergence)
(d) After DSC deformation step

Fig. 5 This figure depicts a shape optimization step and a DSC deformation step when solving the Cantilever beam problem. The shape optimization step calculates an improved position for the non-fixed interface nodes (*red*) using MMA. The orange arrows indicate the gradients $\frac{d\phi(\mathbf{u},\mathbf{x})}{dx_v}$ for each non-fixed interface vertex v. In this case MMA converged in 5 iterations and the result (*orange*) is seen in Fig. 5c. The interface vertices are then moved to these more optimal positions using DSC as seen in Fig. 5d

connected to the node (and thereby the minimum quality of the mesh). The move limits are illustrated in Fig. 6.

Several optimization algorithms exist to solve this type of smooth, non-linear optimization problem. In this work, the gradient-based method *Method of Moving Asymptotes (MMA)* (Svanberg 1987) is applied. However, other optimization algorithms have successfully been applied and a thorough comparison is a suggestion for future work. The shape optimization step is a complete optimization procedure which consists of a number of iterations. At each iteration i, the objective function is evaluated once. Furthermore, the MMA optimization procedure has converged when either $i = 10$ or $||\mathbf{x}^i - \mathbf{x}^{i-1}||_\infty < \alpha$, where α is a user-defined parameter.

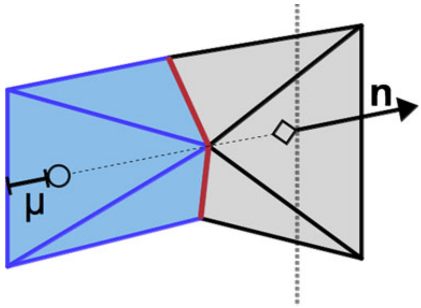


Fig. 6 Illustration of the move limits $x_v^{\min}$ (○) and $x_v^{\max}$ (◇) in the normal direction $\mathbf{n}$ imposed on design variable x_v. In this case, $x_v^{\min}$ is determined by the mesh since the design variable is not able to move closer to a non-neighbouring edge than the distance μ. On the other hand, $x_v^{\max}$ is determined by the design domain since design variables always should stay inside the design domain

Using a gradient-based method implies that we need to calculate the gradients of the objective function and the global constraints wrt. each of the design variables x_v. This is done using the standard adjoint variable method. Consequently, we will just state the result for the case where the objective is to minimize compliance

$$\frac{d\phi(\mathbf{u},\mathbf{x})}{dx_v} = -\mathbf{u}^T \frac{\partial \mathbf{K}(\mathbf{x})}{\partial x_v}\mathbf{u} \tag{7}$$

These gradients are, as the objective function, evaluated at each iteration of the MMA algorithm.

2.3.2 Topology optimization

In addition to changing the shape, we want the method to be able to change the topology of the structure. Therefore, a mechanism to introduce holes is needed. We will base such a mechanism on the well described *topological derivative* (Eschenauer et al. 1994; Sokolowski and Zochowski 1999; Céa et al. 2000; Feijóo et al. 2003). The topological derivative corresponds to the influence on the objective function of introducing an infinitesimal hole in element e. However, we will not introduce an infinitesimal hole, but remove all material from the element.

Comparing Amstutz (2010) and Bendsøe and Sigmund (1999) it is clear that the topological derivative and strain energy density (SED) are very closely related and equal for a number of compliance minimization cases. For 2D and 3D thermal problems and for 2D elasticity problems (for plane stress and Poisson's ratio $\frac{1}{3}$) and for 3D elasticity problems (for Poisson's ratio $\frac{1}{5}$), the topological gradients are equal to the SED multiplied by above factors. In this paper we consider 2D plane elasticity and hence the topological gradient for element e and Poisson's ratio $\frac{1}{3}$ is

$$U_e(\mathbf{u}, \mathbf{x}) = 3\mathbf{u}^T \mathbf{K}_e(\mathbf{x})\mathbf{u} \tag{8}$$

We use this expression independent of the physical Poisson's ratio which may introduce a small error. However, since we anyway introduce finite sized (and triangular) holes, this error is considered negligible.

The goal is to utilize the available material as efficiently as possible. Consequently, material is removed from elements where it affects compliance as little as possible. This means we remove material from the elements which have the smallest SED. The exact procedure (depicted in Fig. 7) is, at each time step t, to remove all material from the solid elements which fulfill

$$\frac{U_e(\mathbf{u}, \mathbf{x}) - \min_j U_j(\mathbf{u}, \mathbf{x})}{\max_j U_j(\mathbf{u}, \mathbf{x}) - \min_j U_j(\mathbf{u}, \mathbf{x})} < \tau \tag{9}$$

evaluated for $\mathbf{x} = \mathbf{0}$. Here, τ is a user-defined variable which determines the aggressiveness of the method. Furthermore, $\tau \in [0, 1]$ since the left-hand side of (9) evaluates to a value between 0 and 1. Also note that this step does not fulfill any constraints (e.g. the volume constraint).

We will only remove material since solid areas within void regions do not support any load. Furthermore, we do not remove any loaded elements. If we make a non-optimal removal of an element (e.g. by not fulfilling the constraints), the shape optimization will correct for this by moving the interface vertices such that the area is filled with material once again. Also, if the removed elements are close to the interface, the shape optimization process undergoes a speed up. Therefore the topology optimization step speeds up the optimization process in addition to introducing holes. Finally, this step detaches the structure from the supports where this is optimal. This is not possible in the shape optimization step since the supported nodes are fixed and therefore not able to move away from the support.

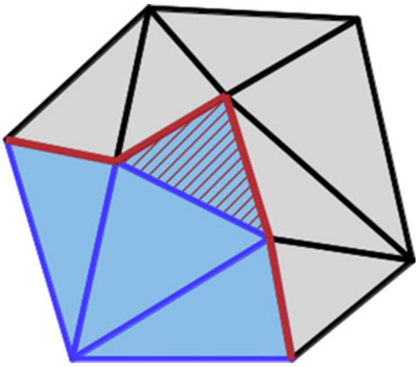
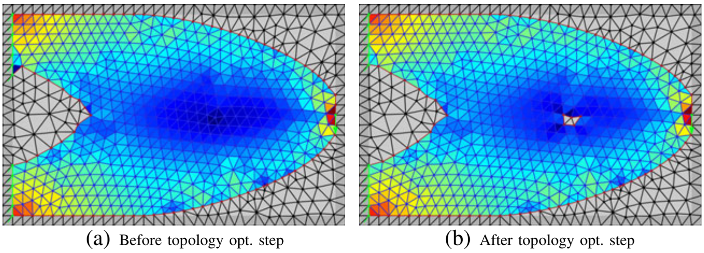


Fig. 8 The hatched triangle illustrates a redundant element, i.e. an element which is only connected to the rest of the structure via one edge

In addition to removing low SED elements, we will recursively remove elements which are only connected to the structure via one edge (e.g. the hatched element in Fig. 8). In the worst case, removing material from these elements changes the stiffness of the structure by a small amount. Therefore, these elements are most likely not optimal and therefore redundant. Again, if a redundant element was optimal anyway, the shape optimization would reintroduce material where it had been removed.

A large value of τ results in fast convergence to an optimized topology, but also higher probability of ending in a poor local minimum. In particular elements can get isolated in "islands" of material which are not connected to any loads. This means that the stiffness matrix is not positive definite anymore which causes the FE analysis to break down. To ensure this will not happen, we perform a connected component analysis which finds the connected structures. The largest connected structure is then maintained, while all others are removed.

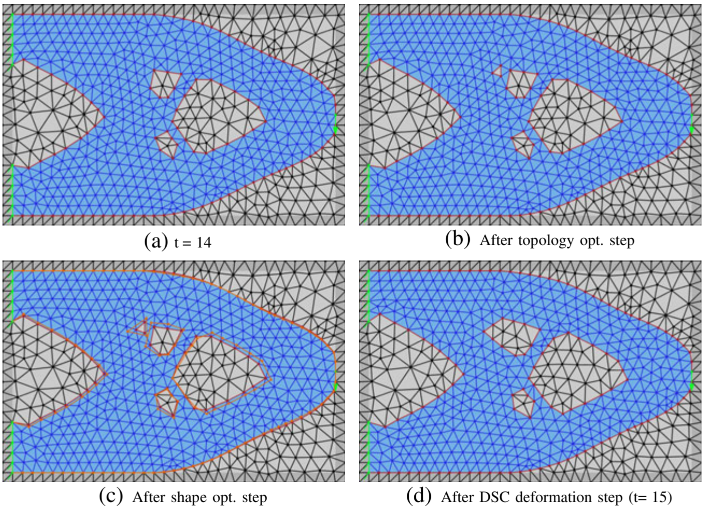
Removing material from redundant elements is computationally cheap and removing material from floating elements is important for the robustness. Therefore, we will remove material from these elements both before and after removing material from low SED elements.



(a) Before topology opt. step

(b) After topology opt. step

Fig. 7 The structure before and after a topology optimization step for the Cantilever beam problem. The 'jet' colormap has been used for visualizing the strain energy densities, which are scaled such that minimum is *blue* and maximum is *red*


(a) t = 14

(b) After topology opt. step

(c) After shape opt. step

(d) After DSC deformation step (t= 15)

Fig. 9 This figure displays the fundamental steps in one time step of the described method. Time step $t = 14$ of the Cantilever beam problem in (**a**) is used as a starting point. As seen in (**b**), step 1 creates a hole consisting of a single triangle. Furthermore, (**c**) displays the calculated optimal shape in orange, i.e. after step 2, the shape optimization step. Finally, (**d**) depicts the structure after step 3, after the interface vertices have been moved to their new position using the DSC method. (**d**) therefore also depicts the structure at time step $t = 15$

2.4 Method overview

To be able to explain how the described steps are combined, we will give an overview of the suggested method.

The method is initialized by filling the entire design domain with material as seen in Fig. 4. Not having to define a valid structure as a starting point increases the simplicity of use and our experiments have shown that this does not limit the design. However, it is also possible to initialize the algorithm with a valid structure as done by e.g. Allaire et al. (2004).

In addition, one has to define three parameters; discretization δ, threshold τ and accuracy α. The discretization δ specifies the average length of the edges in the simplicial complex. The threshold τ defines the aggressiveness in removing elements when performing the topology optimization step. Finally, the parameter α determines the accuracy of the solution $\mathbf{x}^*$ to the shape optimization problem (4) estimated by MMA. Furthermore, it determines when the proposed method has converged as described below.

When the approach has been initialized, the design domain is automatically discretized into a well-shaped triangular mesh using DSC. The bulk of the suggested method then consists of iterating a series of steps until convergence has been reached. These steps are listed below and an illustration can be seen in Fig. 9. In one time step, from time step t to $t + 1$, the algorithm:

Step 1: *Topology optimization* Recursively removes redundant elements, elements connected to the structure via one edge, and elements which are not connected to the largest structure. Then computes the strain energy density for each solid element and removes material from the solid elements which fulfill (9), i.e. the elements having the lowest SED. Note that we do not fulfill any constraints in this step. Finally, removes redundant elements and ”islands” of material again.

Step 2: *Shape optimization* Solves the optimization problem from (4) using the iterative gradient-based optimization algorithm MMA. MMA hereby estimates the optimal values $\mathbf{x}^*$ of the design variables to the level of accuracy α. The new position for each non-fixed interface vertex v is then calculated by $\mathbf{p}_v^* = \mathbf{p}_v + x_v^* \cdot \mathbf{n}_v$. Notice that this is a complete optimization procedure and so consists of a number of iterations (at maximum 10). Each of these iterations both evaluates the objective function and calculates gradients.

Step 3: *DSC deformation* Moves each non-fixed interface vertex v from position $\mathbf{p}_v$ to $\mathbf{p}_v^*$. This is done by iterating the following two steps until all interface vertices have reached their new position.

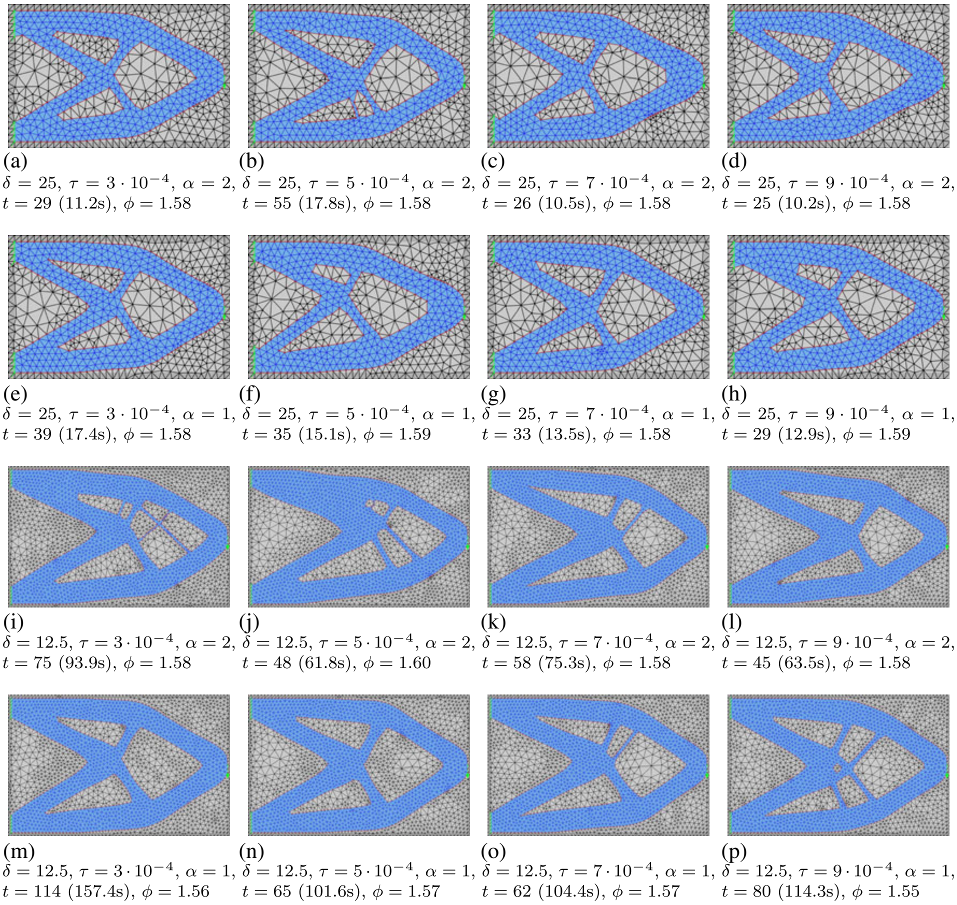
1. Move each non-fixed interface vertex v as far as possible, i.e. without creating any degenerate triangles, along a straight line from $\mathbf{p}_v$ to $\mathbf{p}_v^*$.
2. Improve the quality and detail level of the mesh through the mesh operations smoothing, edge flip, vertex insertion and vertex removal.

Note that moving the interface vertices to an optimized position will close any non-optimal holes created in step 1 since DSC removes degenerate triangles (triangles with very small area, short edges or small minimum angle).

As mentioned, this optimization procedure is repeated until convergence. The method has converged if the position $\mathbf{p}_v^{t+1}$ of each non-fixed interface node v is less than the distance α away from a line segment which is part of the interface at time step t. To ensure final convergence one may finalize the procedure with some time steps without the topology optimization step (i.e. a classical shape optimization) and with a finer convergence criterion. This may in some cases change the boundaries but our experience is that these additional steps have little influence on the objective function. Consequently, we will not apply this.

3 Results

In this section, we present the solutions to two standard minimum compliance problems using the presented approach. To generate the results, we have constrained the volume of the structure to be at maximum 50 % of the design domain ($V^* = 0.5$). The objective function is scaled such that $\phi = 1$ at time step $t = 0$, i.e. when the entire design domain is filled with material. The μ parameter follows the discretization by the relation $\mu = \frac{1}{40}\delta$. Finally, the timings are performed on a laptop with a 2.4 GHz Intel Core i7



(a) $\delta = 25$, $\tau = 3 \cdot 10^{-4}$, $\alpha = 2$, $t = 29$ (11.2s), $\phi = 1.58$

(b) $\delta = 25$, $\tau = 5 \cdot 10^{-4}$, $\alpha = 2$, $t = 55$ (17.8s), $\phi = 1.58$

(c) $\delta = 25$, $\tau = 7 \cdot 10^{-4}$, $\alpha = 2$, $t = 26$ (10.5s), $\phi = 1.58$

(d) $\delta = 25$, $\tau = 9 \cdot 10^{-4}$, $\alpha = 2$, $t = 25$ (10.2s), $\phi = 1.58$

(e) $\delta = 25$, $\tau = 3 \cdot 10^{-4}$, $\alpha = 1$, $t = 39$ (17.4s), $\phi = 1.58$

(f) $\delta = 25$, $\tau = 5 \cdot 10^{-4}$, $\alpha = 1$, $t = 35$ (15.1s), $\phi = 1.59$

(g) $\delta = 25$, $\tau = 7 \cdot 10^{-4}$, $\alpha = 1$, $t = 33$ (13.5s), $\phi = 1.58$

(h) $\delta = 25$, $\tau = 9 \cdot 10^{-4}$, $\alpha = 1$, $t = 29$ (12.9s), $\phi = 1.59$

(i) $\delta = 12.5$, $\tau = 3 \cdot 10^{-4}$, $\alpha = 2$, $t = 75$ (93.9s), $\phi = 1.58$

(j) $\delta = 12.5$, $\tau = 5 \cdot 10^{-4}$, $\alpha = 2$, $t = 48$ (61.8s), $\phi = 1.60$

(k) $\delta = 12.5$, $\tau = 7 \cdot 10^{-4}$, $\alpha = 2$, $t = 58$ (75.3s), $\phi = 1.58$

(l) $\delta = 12.5$, $\tau = 9 \cdot 10^{-4}$, $\alpha = 2$, $t = 45$ (63.5s), $\phi = 1.58$

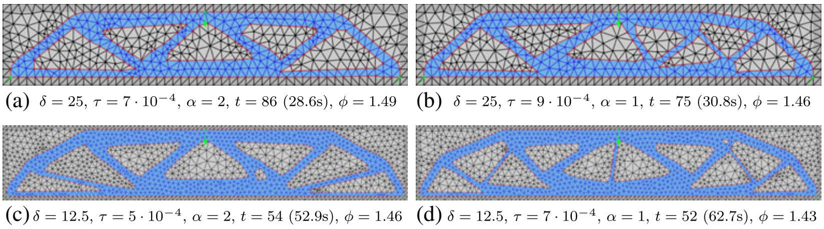
(m) $\delta = 12.5$, $\tau = 3 \cdot 10^{-4}$, $\alpha = 1$, $t = 114$ (157.4s), $\phi = 1.56$

(n) $\delta = 12.5$, $\tau = 5 \cdot 10^{-4}$, $\alpha = 1$, $t = 65$ (101.6s), $\phi = 1.57$

(o) $\delta = 12.5$, $\tau = 7 \cdot 10^{-4}$, $\alpha = 1$, $t = 62$ (104.4s), $\phi = 1.57$

(p) $\delta = 12.5$, $\tau = 9 \cdot 10^{-4}$, $\alpha = 1$, $t = 80$ (114.3s), $\phi = 1.55$

Fig. 10 This figure displays 16 results of solving a 500 × 800 units^2 Cantilever beam problem using varying discretization δ, threshold τ and accuracy α. The number of time steps t, the calculation time and the final value of the objective function ϕ are depicted in the captions along with the values of the parameters



(a) $\delta = 25,\ \tau = 7 \cdot 10^{-4},\ \alpha = 2,\ t = 86$ (28.6s), $\phi = 1.49$ (b) $\delta = 25,\ \tau = 9 \cdot 10^{-4},\ \alpha = 1,\ t = 75$ (30.8s), $\phi = 1.46$

(c) $\delta = 12.5,\ \tau = 5 \cdot 10^{-4},\ \alpha = 2,\ t = 54$ (52.9s), $\phi = 1.46$ (d) $\delta = 12.5,\ \tau = 7 \cdot 10^{-4},\ \alpha = 1,\ t = 52$ (62.7s), $\phi = 1.43$

Fig. 11 The 200 × 1200 units^2 MBB beam problem solved using the suggested method with parameters in units as depicted in the caption

processor and 8 GB of 1333 MHz DDR3 RAM utilizing a single thread only.

We have solved a 500 × 800 units^2 cantilever beam problem using different values of the δ, τ and α parameters. The solutions can be seen in Fig. 10. To give an idea of the problem sizes, the solution in Fig. 10g contains $N_e = 621$ solid elements, $N_d = 2830$ degrees of freedom and $N_v = 166$ design variables. Whereas the solution in Fig. 10o contains $N_e = 2378$ solid elements, $N_d = 10278$ degrees of freedom and $N_v = 371$ design variables. The convergence rates for all 16 tests are depicted in Fig. 13. Furthermore, the consequence of applying a global perimeter constraint to this problem is depicted in Fig. 12.

Besides the cantilever beam problem, we have solved the 200×1200 units^2 MBB beam problem, again using different parameters. The results are seen in Fig. 11.

Examining the results in Fig. 10, we observe that the results are quite consistent and relatively independent of the control parameters. As expected, the optimization process occasionally stops prematurely when the accuracy is low ($\alpha = 2$). This is evident when looking at the final objective function values, but it is also evident that the results are relatively close to optimum. For the coarse mesh ($\delta = 25$) the method converges to the same simple design in almost all cases. For the fine mesh ($\delta = 12.5$) the method sometimes converges to designs with one or two extra bars. This issue is also evident when solving the MBB beam problem (Fig. 11). This means there is no convergence with mesh refinement which was to be expected since we have not imposed any geometrical constraints on either minimum length scale or on global perimeter. Hence, minimum length scale is controlled by the chosen element size control parameter δ.

It is well known that a global perimeter constraint solves the issue of mesh dependency (Ambrosio and Buttazzo 1993). In Fig. 12, we show that this is also the case here. The global perimeter constraint

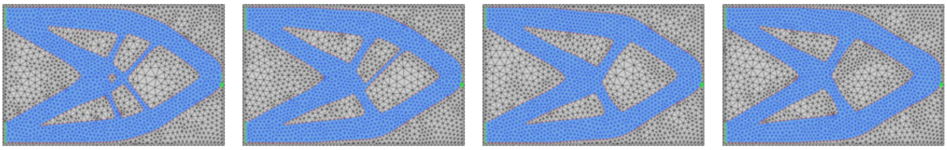
$$g_2(\mathbf{x}) = \beta \cdot \frac{L(\mathbf{x})}{L_0} - 1 \tag{10}$$

is applied to the shape optimization problem in (4). Here $L(\mathbf{x})$ is the length of the interface, L_0 is the initial length of the interface and β is a weight which determines how much regularization we apply. Specifically, we add the constraint to the objective function such that we minimize

$$\phi(\mathbf{u}, \mathbf{x}) = \mathbf{u}^T \mathbf{K}(\mathbf{x})\mathbf{u} + g_2(\mathbf{x}) \tag{11}$$

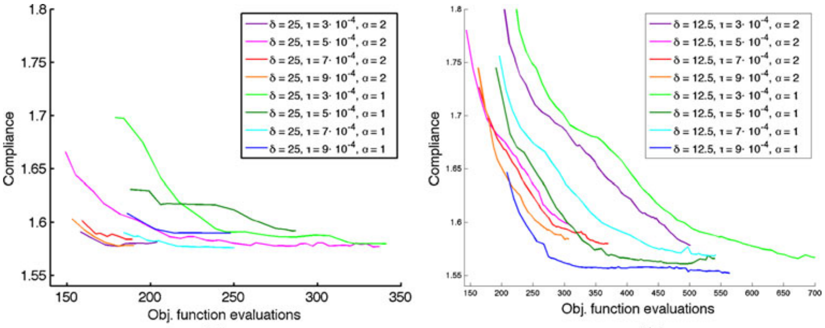
It is clearly seen from the results in Fig. 12 that a perimeter constraint solves the issue if β is chosen wisely. Furthermore, Fig. 12b depicts what happens if too little ($\beta = 0.2$) regularization is applied.

Figure 13 shows that all tests from Fig. 10 converge smoothly. However, note that since the mesh and possibly a number of design variables are changing, the objective function may be non-monotone between two time steps. The figure also confirms that a low accuracy ($\alpha = 2$) generally stops the optimization process prematurely, i.e. just before the curve levels off. However, the time consumption is decreased if the requirements to the accuracy are lowered.



(a) $\beta = 0.0,\ t = 80$ (114.3s), $\phi = 1.55$ (b) $\beta = 0.2,\ t = 57$ (99.5s), $\phi = 1.57$ (c) $\beta = 0.3,\ t = 35$ (70.0s), $\phi = 1.57$ (d) $\beta = 0.4,\ t = 213$ (249.5s), $\phi = 1.57$

Fig. 12 The Cantilever beam problem solved with a global perimeter constraint. The constraint is weighted by the parameter β which is depicted in the captions. Furthermore, the parameters $\delta = 12.5$, $\tau = 9 \cdot 10^{-4}$ and $\alpha = 1$ are used



(a) (b)

Fig. 13 Convergence plots for 16 tests that solve the Cantilever beam problem with the parameters shown in the legends. The plotting starts when the volume constraint is decreased to $V^* = 0.5$ which is at time step $t = 20$

Also the discretization δ influences the time consumption. This means, as expected, that both a finer discretization and higher accuracy results in more objective function evaluations and thereby slower convergence. On the other hand, a higher threshold τ does not seem to necessarily result in faster convergence.

4 Discussion

In this article, we present a novel approach to topology optimization based on the DSC method.

The approach is shown to be both efficient, simple to apply and robust. Efficiency is primarily achieved by unifying shape and topology optimization such that they complement each other. However, using a single adaptive explicit representation also contributes to the efficiency. Even so, we still believe we can improve the efficiency and therefore this is a topic for future work. To increase the simplicity of use, the method does not require any initial guess, only a few natural parameters. In addition, when the solution is found, the shape is available with no conversion due to the explicit representation. Finally, we have shown that the method converges smoothly to an optimized solution without any regularization for the cantilever beam and the MBB beam problems using a wide range of parameters. Therefore, besides mesh dependency, the method has shown to be robust. To show that it is possible to eliminate the mesh dependency, we have successfully imposed a global perimeter constraint. Still, applying a perimeter constraint, i.e. choosing the β parameter, is not intuitive. Therefore, a main goal of our on-going study is to include more intuitive local (minimum length scale) constraints.

Additional topics for future work are 3D and multiple phases. The first topic is about extending the method to 3D where it is possible to test the scalability of the method properly. Since the DSC method already exists in 3D, there should be nothing hindering this extension. The second topic is about supporting multiple types of material in the same optimization process. The DSC method already supports multiple phases (Misztal et al. 2012). It is therefore hopefully a straightforward extension.

Acknowledgments The authors appreciate the support from the Villum Foundation through the grant: “NextTop”

References

Allaire G, Jouve F, Toader AM (2004) Structural optimization using sensitivity analysis and a level-set method. J Comput Phys 194(1):363–393

Allaire G, Dapogny C, Frey P (2011) Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh. Compt Rendus Math 349(17–18):999–1003

Allaire G, Dapogny C, Frey P (2013) A mesh evolution algorithm based on the level set method for geometry and topology optimization. Struct Multidiscip Optim:1–5

Ambrosio L, Buttazzo G (1993) An optimal design problem with perimeter penalization. Calc Var 1(1):55–69

Amstutz S (2010) A penalty method for topology optimization subject to a pointwise state constraint. ESAIM: Control Optim Calc Var 16:523–544

Arnout S, Firl M, Bletzinger KU (2012) Parameter free shape and thickness optimisation considering stress response. Struct Multidiscip Optim 45(6):801–814

Bærentzen JA, Gravesen J, Anton F, Aanæs H (2012) Guide to computational geometry processing: foundations, algorithms, and methods. Springer, London

Bærentzen JA, Revall Frisvad J, Aanæs H (2013) Geometry and linear algebra (GEL) library. http://www2.imm.dtu.dk/projects/GEL/

Bendsøe MP, Sigmund O (1999) Material interpolation schemes in topology optimization. Arch Appl Mech 69(9–10):635–654

Bletzinger KU, Maute K (1997) Towards generalized shape and topology optimization. Eng Optim 29(1–4):201–216

Céa J, Garreau S, Guillaume P, Masmoudi M (2000) The shape and topological optimizations connection. Comput Methods Appl Mech Eng 188(4):713–726

Chen Y, Davis TA, Hager WW, Rajamanickam S (2008) Algorithm 887: Cholmod, supernodal sparse cholesky factorization and upyear/downyear. ACM Trans Math Softw 35(3):22:1–22:14

Cheng S, Dey T, Shewchuk J (2012) Delaunay mesh generation. Chapman & Hall/CRC Computer and Information Science. CRC Press INC, Boca Raton

Davis TA, Hager WW, Duff IS (2013) Suite Sparse. http://www.cise.ufl.edu/research/sparse/SuiteSparse/

Ding Y (1986) Shape optimization of structures: a literature survey. Comput Struct 24(6):985–1004

Eschenauer HA, Kobelev VV, Schumacher A (1994) Bubble method for topology and shape optimization of structures. Struct Multidiscip Optim 8:42–51

Feijóo RA, Novotny AA, Taroco E, Padra C (2003) The topological derivative for the poisson's problem. Math Model Methods Appl Sci 13(12):1825–1844

Ha SH, Cho S (2008) Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh. Comput Struct 86(13–14):1447–1455

Kim DH, Lee SB, Kwank BM, Kim HG, Lowther D (2008) Smooth boundary topology optimization for electrostatic problems through the combination of shape and topological design sensitivities. IEEE Trans Magn 44(6):1002–1005

Le C, Bruns T, Tortorelli D (2011) A gradient-based, parameter-free approach to shape optimization. Comput Methods Appl Mech Eng 200(9–12):985–996

Maute K, Ramm E (1995) Adaptive topology optimization. Struct Optim 10:100–112

Misztal MK, Bærentzen JA (2012) Topology adaptive interface tracking using the deformable simplicial complex. ACM Trans Graph 31(3):24

Misztal MK, Bridson R, Erleben K, Bærentzen JA, Anton F (2010) Optimization-based fluid simulation on unstructured meshes. In: 7th workshop on virtual reality interaction and physical simulation VRIPHYS, p 10

Misztal MK, Erleben K, Bargtei A, Fursund J, Christensen BB, Bærentzen JA, Bridson R (2012) Multiphase flow of immiscible fluids on unstructured moving meshes. In: Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, Eurographics Association, Aire-la-Ville, Switzerland, SCA '12, pp 97–106

Mohammadi B, Pironneau FO (2001) Applied shape optimization for fluids. Oxford University Press, Oxford

Osher SJ, Fedkiw RP (2002) Level set methods and dynamic implicit surfaces, 1st edn. Springer, New York

Sokolowski J, Zochowski A (1999) On the topological derivative in shape optimization. SIAM J Control Optim 37(4):1251–1272

Sokołowski J, Zolésio J (1992) Introduction to shape optimization: shape sensitivity analysis. Springer series in computational mathematics. Springer-Verlag, New York

Svanberg K (1987) The method of moving asymptotes—a new method for structural optimization. Int J Numer Methods Eng 24(2):359–373

Wang M, Wang X, Guo D (2003) A level set method for structural topology optimization. Comput Methods Appl Mech Eng 192(1):227–246

Xia Q, Shi T, Liu S, Wang MY (2012) A level set solution to the stress-based structural shape and topology optimization. Comput Struct 90–91:55–64

Yamasaki S, Nomura T, Kawamoto A, Sato K, Nishiwaki S (2011) A level set-based topology optimization method targeting metallic waveguide design problems. Int J Numer Methods Eng 87(9):844–868

Appendix B

Paper B

Christiansen, A. N., Bærentzen, J. A., Nobel-Jørgensen, M., Aage, N., and Sigmund, O. (2014a). Combined shape and topology optimization of 3D structures. *Computers & Graphics*, 46(2015):25 – 35. Shape Modeling International 2014

Computers & Graphics 46 (2015) 25–35

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

SMI 2014

Combined shape and topology optimization of 3D structures

Asger Nyman Christiansen*, J. Andreas Bærentzen, Morten Nobel-Jørgensen, Niels Aage, Ole Sigmund

Technical University of Denmark, Denmark

ARTICLE INFO

Article history:
Received 29 June 2014
Received in revised form
24 August 2014
Accepted 16 September 2014
Available online 2 October 2014

Keywords:
Topology optimization
Shape optimization
Deformable simplicial complex method
Structural design

ABSTRACT

We present a method for automatic generation of 3D models based on shape and topology optimization. The optimization procedure, or model generation process, is initialized by a set of boundary conditions, an objective function, constraints and an initial structure. Using this input, the method will automatically deform and change the topology of the initial structure such that the objective function is optimized subject to the specified constraints and boundary conditions. For example, this tool can be used to improve the stiffness of a structure before printing, reduce the amount of material needed to construct a bridge, or to design functional chairs, tables, etc. which at the same time are visually pleasing.

The structure is represented explicitly by a simplicial complex and deformed by moving surface vertices and relabeling tetrahedra. To ensure a well-formed tetrahedral mesh during these deformations, the Deformable Simplicial Complex method is used. The deformations are based on optimizing the objective, which in this paper will be maximizing stiffness. Furthermore, the optimization procedure will be subject to constraints such as a limit on the amount of material and the difference from the original shape.

1. Introduction

Topology optimization is the discipline of finding the optimal shape and topology of a structure [1,2]. It can be used to solve a wide variety of design problems arising when producing diverse products such as cars, houses, computer chips and antennas. The manufacturers are often concerned with finding the stiffest structure, the lightest structure which does not break, the structure with the highest cooling effect, or the structure with the best flow or highest efficiency.

With the advances in 3D printing technology, topology optimization is not just of interest to manufacturers, but to anyone who has access to a 3D printer. Most consumers lack formal training in structural mechanics, which can hinder the process with many iterations and costly failed attempts. Consumers can under-engineer a design unsuitable for the intended load, or over-engineer a design that wastes expensive construction material. Topology optimization offers consumers a tool for designing shapes that meet their structural needs while using minimal construction resources.

In this paper, we present a fully automated design tool for designing structurally sound structures which can be manufactured, constructed or printed. The modeler only has to specify boundary conditions, the optimization objective, constraints and an initial structure. In other words, the designer specifies a set of requirements (the functionality of the structure and not the structure itself) and the method automatically designs a structure which fits those requirements. Note that this design process is significantly different from today where a designer manually models a structure and requirements are taken into account during this design process.

The proposed method for topology optimization is based on the Deformable Simplicial Complex (DSC) method [3]. The DSC method represents a solid structure with a conforming tetrahedral mesh (a simplicial complex) whose tetrahedral elements either lie entirely inside or outside the structure. The interface between solid and void (the surface) is represented explicitly by the triangular faces shared by an interior and exterior tetrahedral element. Furthermore, the DSC method ensures well-formed tetrahedral elements by constantly performing mesh improvement routines while the surface is being deformed. Finally, it provides adaptive resolution, allowing fine details where and when needed.

The method uses two optimization strategies:

Discrete optimization Relabels elements from solid to void to improve the objective or constraints which are not satisfied. The relabeling is based on topological derivatives [4–8], i.e. the change in the objective or constraints by introducing an infinitesimal hole.

[*] Corresponding author. Tel.: +45 45255984.
E-mail address: asny@dtu.dk (A.N. Christiansen).

http://dx.doi.org/10.1016/j.cag.2014.09.021
0097-8493/© 2014 Elsevier Ltd. All rights reserved.

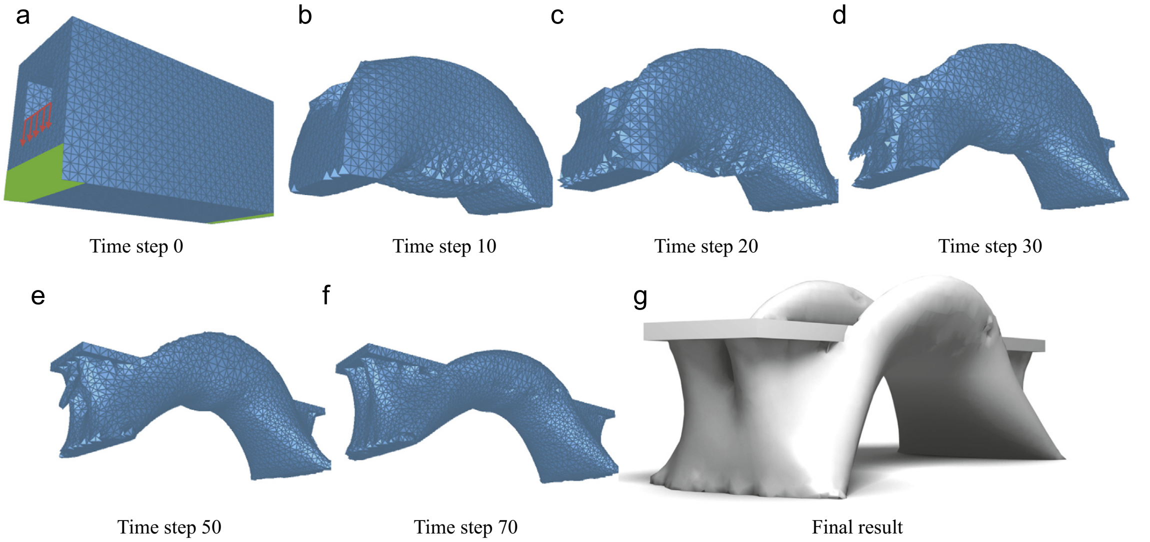


Fig. 1. Given a few input parameters, the proposed method automatically optimizes the shape and topology of a 3D structure. Here is an example of optimizing a bridge. The initial structure is seen to the upper left along with supports (green) and loads (red). This structure is optimized such that stiffness is maximized and the amount of material is minimized. A few iterations of the method are depicted along with the result. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Continuous optimization Performs a non-parametric shape optimization [9–13]. First, an improved shape, which is within a small perturbation of the current shape, is found by solving a constrained optimization problem using the Method of Moving Asymptotes (MMA) [14]. The surface is then deformed to this improved shape using the DSC method [3]. While the surface is deformed, the mesh is adapted such that its tetrahedral elements are well-formed at all times.

These optimization strategies are iterated until changes are small. An example is seen in Fig. 1.

We will show that this tool is of interest to both engineers and designers. For example, we show that it can be used to improve stiffness and balance of a 3D model, to save material and to generate functional as well as, in our opinion, visually pleasing designs.

1.1. Related work

Recent trends in the computer graphics society are to add mechanical properties to 3D models. Prévost et al. have been concerned with the balance of printed models [15], Skouras et al. about printing deformable characters using a stiff and soft material [16] and several research teams have focused on self-supporting masonry structures [17–19].

A major concern has been to improve the stiffness of 3D models. Umetani et al. perform a cross-sectional structural analysis and visualize the result [20]. A user can then manually edit the model to improve the stiffness while getting almost instant feedback. The instant feedback is only possible because the analysis is limited to cross-sections. Stava et al. present a more automated method for improving stiffness [21]. They perform a complete worst-case structural analysis on a tetrahedral mesh to determine the structurally weak regions. Based on this analysis, it is decided whether to improve the model by thickening, hollowing or adding a strut. Finally, Zhou et al. [22] also perform a worst-case structural analysis with more precise determination of the worst-case loads than in [21]. Furthermore, they conclude that solving a shape optimization problem to minimize stress is impractical due to the non-linearity and non-convexity of the problem. Therefore, they make do with visualizing the structurally weak regions.

Topology optimization problems are indeed non-convex. However, the topology optimization community has been solving these problems to at least local optimality for decades and the resulting designs usually perform better than designs optimized by humans [2]. Feasible solutions to these problems are often found by standard numerical gradient-based optimization algorithms. However, note that the smooth compliance functional is often chosen as the objective function to ease the optimization instead of the non-smooth, but often more interesting, maximal stress as Zhou et al. propose.

A key ingredient in a topology optimization method is the shape representation which is required to be able to handle topology changes. Hence, topological optimization has focused primarily on implicit representations over uniform voxel grids. Such representations can handle topology changes but lead to fixed-resolution results with cuberille artifacts. The most popular implicit topology optimization approaches are the density and level set approaches. The density approach [23,2] represents the structure by assigning a density value between 0 (void) and 1 (material) to each cell in a fixed grid or mesh. The structure is now deformed by changing these density values. The level set approach uses the level set method [24] evaluated on a fixed grid or mesh [25,26]. Here, the structure is represented by the zero level set and deformed by changes to the level set function. Both methods iteratively change the shape to approach the optimum.

We propose to represent the surface explicitly. An explicit representation, for example a triangle mesh, has previously been used for shape optimization [9,10]. However, shape optimization does not allow for topology changes and often only small shape deformations. Furthermore, it has been used in combination with the level set method [27–31] where it is necessary to constantly switch between the implicit and explicit representations. An explicit representation has also been used in combination with a computationally expensive remeshing of the entire design domain at each iteration [4,32]. Finally, it has previously been shown that using the DSC method for topology optimization works in 2D and therefore has potential [33]. However, here, we show that this concept is able to solve real-world topology optimization problems in 3D.

Note that this list of structural optimization methods is far from exhaustive.

1.2. Contributions

The main contributions of this paper are as follows.

- As opposed to previous methods introduced in computer graphics, our method automatically optimizes the shape and topology of structure given boundary conditions, an objective function, constraints and an initial shape. This completely eliminates the manual editing which has been characteristic for the current approaches.
- Compared to current methods from the topology optimization community, the method uses a single explicit representation to represent the structure and, at the same time, is able to handle topology changes. This gives rise to several advantages including a single mesh for shape representation and finite element calculations, possibility of both continuous and discrete optimization strategies and both the initial and optimized structure are in the form of surface triangle meshes. Finally, the adaptive mesh makes it possible to achieve a much more detailed result within reasonable time on an ordinary laptop than otherwise possible using the standard fixed grid methods.
- To be able to solve real-world topology optimization problems in 3D, it was necessary to make significant changes compared to the 2D proof-of-concept by Christiansen et al. [33]. Consequently, the discrete step relabels elements based on an optimization procedure which takes constraints into account instead of based on a simple threshold of the objective. Furthermore, the presented method handles self-weight, it is initialized by any surface triangle mesh, areas can be fixed to either solid or void and several global constraints have been implemented and utilized. Finally, the requirements for computational efficiency is much higher in 3D than 2D. Therefore, the mesh adaptivity of the DSC method is utilized and the computations are distributed on multiple cores.

2. Method

The proposed method uses a simplicial complex to represent the shape of a structure. A simplicial complex discretizes a domain into tetrahedral elements. In 3D it consists of the simplices; nodes (points), edges (line pieces), faces (triangles) and tetrahedra (triangular pyramids). Furthermore, the tetrahedra do not overlap and any point in the discretized domain is either inside a tetrahedron or on the boundary between tetrahedra. In addition, all tetrahedra are labeled as being either void (no material) or solid (filled with material). Therefore, the interface between solid and void (the surface) is represented by the faces that are sandwiched between a tetrahedron labeled void and a tetrahedron labeled solid. Fig. 2 depicts a cube represented by a simplicial complex. The tetrahedral mesh generator TetGen [34] is used to generate the initial mesh.

Apart from the shape representation, the tetrahedral elements of the simplicial complex can be used for physical computations using the finite element method. Since the finite element analysis will produce large errors if used with nearly degenerate tetrahedra, it is important to sustain a high quality mesh.

2.1. Deformable simplicial complex method

To ensure a high quality mesh, we use the Deformable Simplicial Complex (DSC) method [3].[1] The DSC method ensures high quality tetrahedral elements during deformation of a model embedded in a simplicial complex as illustrated in Fig. 2. Low quality tetrahedra (slivers, wedges, caps and needles) are removed by continuously performing a set of mesh operations while the surface is being deformed. The tetrahedron quality measure is $6\sqrt{2}V/(\frac{1}{6}\sum_i l_i^2)^{3/2}$ [35] where V is the volume of the tetrahedron and l_i is the length of edge i. Note that the DSC method only improves the mesh quality where necessary (often near the surface). Furthermore, the DSC method also handles topology changes by removing low quality tetrahedra which are sandwiched between two surfaces. This is illustrated by two objects colliding in Fig. 3.

In addition to ensuring high quality tetrahedral elements, the DSC method also controls the level of details of both the surface and the tetrahedral mesh. In practice, the DSC method attempts to collapse too small simplices and split too large simplices. Consequently, we always attain a mesh of the desired complexity, described by the discretization parameter δ (corresponding to the average edge length). More importantly, the detail control allows for mesh adaptivity. This means that smooth regions on the surface are represented by a more coarse discretization than regions with small features.

The mesh operations used are *smoothing* [36] (not performed on surface nodes), *edge split* [37], *edge collapse* [37], *edge removal* [38] and *multi-face removal* [38]. The latter two use the flips illustrated in Fig. 4. Consequently, these two mesh operations do not change the position of any nodes, only the connectivity. The quality of the mesh is improved by all five operations, whereas the detail level of the mesh is controlled through the operations edge split and edge collapse. Note that changes have been made compared to [3]. The multi-face retriangulation, optimization-based smoothing, null-space smoothing and tetrahedron relabeling operations have not been necessary for this application. Removing these operations has resulted in a significant speed-up. Also, the edge removal operation on the surface and boundary is an addition since [3].

The strategy for moving the surface nodes is to first compute a destination p_n^* for each surface node n currently at position p_n. The destination p_n^* is computed using a user-defined velocity function which, for the case of topology optimization, will be described later. Afterwards, all surface nodes are moved from p_n to p_n^* using the strategy illustrated in Fig. 5.

2.2. Structural analysis

In this paper, we will optimize the topology of physically valid structures in static equilibrium. In order to achieve physical validity, structural analyses using the finite element method are performed. This implies considering the *discretization*, *boundary conditions* and *equilibrium* which are the topics of this section.

As described previously, a domain is discretized into high quality tetrahedral elements which are analyzed using the finite element method. Using quadratic basis functions solves a well-known issue with a jagged surface when using the analysis as a basis for non-parametric shape optimization [11,12]. Consequently, quadratic basis functions are chosen instead of linear to interpolate the tetrahedral elements.

[1] An open-source framework is available at www.github.com/asny/DSC

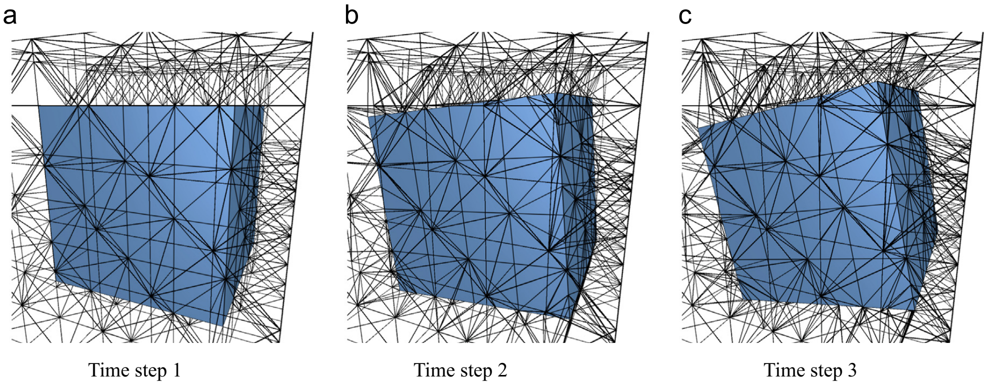


Fig. 2. Rotation of a cube using the Deformable Simplicial Complex method. The interface between solid and void (the surface of the cube) is depicted in turquoise. Furthermore, all edges of the simplicial complex are drawn in black. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

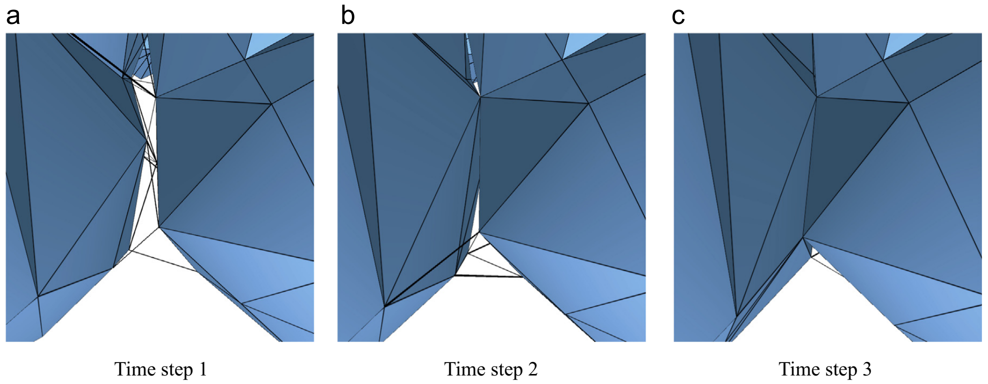


Fig. 3. Illustration of topology changes using the Deformable Simplicial Complex method. Here, only edges having both end nodes on the surface are drawn. As the objects approach each other the tetrahedra between the objects get squeezed. When a tetrahedron between the two surfaces is squeezed too much, this tetrahedron will be collapsed. Consequently, the only thing separating the two objects is a face. However, this face has tetrahedra which are labeled solid on both sides and it is therefore no longer part of the surface. Consequently, the two objects are now merged into one.

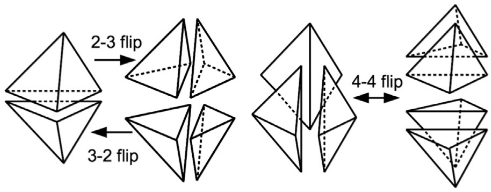


Fig. 4. Illustrations of 2-3, 3-2 and 4-4 flips inspired by the illustration in [38].

Therefore one control point c is associated with each node and edge of a tetrahedron. Furthermore, the positions of all control points are assembled in a vector termed $\boldsymbol{p} = [\dots, \boldsymbol{p}_c^T, \dots]^T$. In addition, each tetrahedron t has an associated material m_t with material parameters *density* ρ_t, *Young's modulus* E_t and *Poisson's ratio* ν_t. Finally, the materials of the tetrahedra are also assembled in a vector $\boldsymbol{m} = [\dots, m_t, \dots]^T$.

The local *stiffness matrix* $\boldsymbol{K}_t$ contains information on the stiffness of tetrahedron t. It depends on both the positions of the control points $\boldsymbol{p}$ and the materials of the tetrahedra $\boldsymbol{m}$ and can be calculated by

$$\boldsymbol{K}_t(\boldsymbol{m}, \boldsymbol{p}) = \int_{V_t} \boldsymbol{B}_t^T(\boldsymbol{p}) \boldsymbol{E}_t(\boldsymbol{m}) \boldsymbol{B}_t(\boldsymbol{p}) \partial(x, y, z) \tag{1}$$

We have chosen only to consider isotropic linear materials. Consequently, the constitutive matrix $\boldsymbol{E}_t(\boldsymbol{m})$ which relates stress and strain is

$$\boldsymbol{E} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ \nu & 1-\nu & \nu & 0 & 0 & 0 \\ \nu & \nu & 1-\nu & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1-2\nu}{2} \end{bmatrix}$$

where $\boldsymbol{E}_t(\boldsymbol{m})$ is shortened to $\boldsymbol{E}$, $E_t(\boldsymbol{m})$ to E and $\nu_t(\boldsymbol{m})$ to ν. Finally, the strain-displacement matrix $\boldsymbol{B}_t(\boldsymbol{p})$ is related to the shape of the tetrahedron and the basis functions. For more details, see a text book on the finite element method used for structural analysis, e.g. [39]. The global stiffness matrix $\boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p})$ can then be assembled from the local stiffness matrices $\boldsymbol{K}_t(\boldsymbol{m}, \boldsymbol{p})$. Note that for elements with void as the associated material, $\boldsymbol{K}_t$ is not defined.

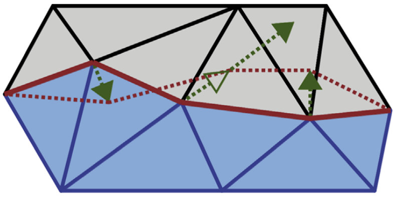


Fig. 5. Illustration of how the surface (red) is moved in 2D. The same principle applies to 3D. A filled arrow indicates the destination $\boldsymbol{p}_n^*$ of the surface node n. One of the nodes cannot move to its destination without creating low quality tetrahedra and it is therefore only moved as depicted by the unfilled arrow. The other two are moved to their destinations. Then, mesh operations are applied to improve the mesh quality and the node that did not reach its destination is moved again. This is repeated until all nodes have reached their destinations. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Consequently, the void elements are eliminated from the finite element analysis, which decreases computation time.

In this paper, we will limit ourselves to static problems subject to a single load case. These problems are modeled by supports and external forces $\boldsymbol{f}_c$ which are both applied to the surface of the structure. In addition to external forces, the weight of the structure will cause gravitational forces

$$\boldsymbol{w}_c(\boldsymbol{m},\boldsymbol{p}) = \boldsymbol{g}\sum_{i\in c} a_i \cdot \rho_i(\boldsymbol{m}) \cdot V_i(\boldsymbol{p}) \tag{2}$$

Here, $\boldsymbol{g} = [0, -9.8, 0]^T$ m/s^2 is a vector of the gravitational acceleration and a_i is a scale factor computed by a mass lumping scheme for each element i. Furthermore, ρ_i is the density and $V_i(\boldsymbol{p})$ is the volume of tetrahedral element i which is adjacent to control point c. Consequently, the global *force vector* is

$$\boldsymbol{f}(\boldsymbol{m},\boldsymbol{p}) = [\ldots,\boldsymbol{f}_c^T + \boldsymbol{w}_c^T(\boldsymbol{m},\boldsymbol{p}),\ldots]^T \tag{3}$$

Since we desire a structure in static equilibrium, the sum of the forces on all particles must be zero (Newton's first law). Consequently, we will utilize the equilibrium equations

$$\boldsymbol{K}(\boldsymbol{m},\boldsymbol{p})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{m},\boldsymbol{p}) \tag{4}$$

These equations are used to calculate the global *displacement vector* $\boldsymbol{u} = [\ldots,\boldsymbol{u}_c,\ldots]$. At each control point c, $\boldsymbol{u}_c$ represents the displacement caused by the forces $\boldsymbol{f}$ applied to the structure. Note that, since $\boldsymbol{K}$ and $\boldsymbol{f}$ are functions of $\boldsymbol{p}$ and $\boldsymbol{m}$, so is $\boldsymbol{u}$.

Solving the equilibrium equations is the most time consuming part of the optimization. Furthermore, the number of equations scales linearly with the number of degrees of freedom. Consequently, the sparse solver *CHOLMOD* [40], which is a part of the *SuiteSparse* library [41], is used to solve the equilibrium equation efficiently using multiple cores.

2.3. Optimization

We want to optimize an objective function f by changing the shape and topology of the structure. Therefore, the objective can be anything as long as it is a function of the shape and topology. Furthermore, there are two ways to change the shape and topology. The first is to change the position $\boldsymbol{p}_n$ of a design node n, the other is to change the material m_e of a design element e. A node is a design node n if it is

- on the surface of the structure,
- not supported,
- not subjected to any external forces and
- not part of a fixed domain (see Section 2.5).

Furthermore, a tetrahedral element is a design element e if it is

- solid,
- not adjacent to a control point subjected to external forces and
- not part of a fixed domain (see Section 2.5).

For the test cases presented here, we seek to find the structure which is as stiff as possible. Consequently, the objective function is compliance

$$f(\boldsymbol{m},\boldsymbol{p}) = \boldsymbol{u}^T\boldsymbol{K}(\boldsymbol{m},\boldsymbol{p})\boldsymbol{u} \tag{5}$$

Note that since this objective is a function of the displacements $\boldsymbol{u}$, we need to solve Eq. (4) to evaluate it. The reason for choosing to minimize compliance and not for example maximal Von Mises stress is that the compliance function is smooth. This is a significant advantage for the optimization algorithm. However, we plan to minimize the maximal Von Mises stress using the same method in the future.

It is often desirable to constrain the optimization. In some test examples, we choose to limit the amount of material used, i.e. the optimization is subject to a global volume constraint:

$$g_1(\boldsymbol{m},\boldsymbol{p}) = \frac{V(\boldsymbol{m},\boldsymbol{p})}{V^*} - 1 \tag{6}$$

where $V(\boldsymbol{m},\boldsymbol{p})$ is the total volume of the solid elements and V^* is the maximum volume of the structure.

Optimized results are often not manufacturable. For example, the optimized results often contain many details. A partial remedy is to constrain the total surface area, called a perimeter constraint [42].

$$g_2(\boldsymbol{m},\boldsymbol{p}) = \frac{A(\boldsymbol{m},\boldsymbol{p})}{A^*} - 1 \tag{7}$$

Here, $A(\boldsymbol{p})$ is the total area of triangles sandwiched between a void and a (not fixed) solid element and A^* is the maximum surface area allowed. This constraint enforces a smoothness of the surface and thereby to some degree prevents small details and thin plates. However, since it is a global constraint, these undesirable features are not guaranteed to be eliminated.

Finally, in some cases, we want to limit the possible change from the original shape. In these cases, the original design nodes are added to a set O. If, during the optimization, an edge connecting two original nodes is split, the new node will be added to the set. However, if a hole appears inside the structure, the nodes on that internal surface are not added. Furthermore, the original surface is stored such that the distance $d_n(\boldsymbol{m},\boldsymbol{p})$ from $n \in O$ to the original surface can be calculated. Finally, the function $t_n(\boldsymbol{m},\boldsymbol{p})$ computes the distance from $n \notin O$ to the surface represented by the nodes in the set O. In other words, this function calculates the thickness of the shell of the structure. We can now limit the change from the original surface as well as ensuring that holes will not appear in this surface by applying the constraint:

$$\begin{aligned} g_3(\boldsymbol{m},\boldsymbol{p}) = & \frac{1}{N_{\in O}} \sum_{n\in O} \max(d_n(\boldsymbol{m},\boldsymbol{p}) - D^*, 0)^2 \\ & + \frac{1}{N_{\notin O}} \sum_{n\notin O} \max(T^* - t_n(\boldsymbol{m},\boldsymbol{p}), 0)^2 \end{aligned} \tag{8}$$

Here, D^* is the maximal change from the original surface and T^* is the minimum thickness of the shell of the structure. Note that g_3 is C^1 continuous and thereby differentiable.

2.3.1. Continuous optimization

The first part of the optimization procedure is to locally perturb the surface of the structure such that it iteratively gets closer to

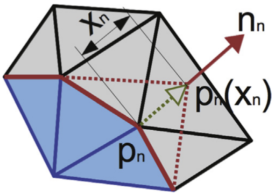


Fig. 6. Illustrates the destination $\boldsymbol{p}_n(x_n)$ of node n as a function of the design variable x_n. Furthermore, $\boldsymbol{p}_n$ is the current position and $\boldsymbol{n}_n$ is the normal.

optimum. This part of the optimization procedure consists of calculating an improved position $\boldsymbol{p}_n^*$ for each design node n. Afterwards, the structure is deformed by moving each design node from its current $\boldsymbol{p}_n$ to the more optimal position $\boldsymbol{p}_n^*$ as described in Section 2.1. Note that since the DSC method handles topology changes, these can occur. Thin structures can collapse and holes can disappear. However, holes will not appear inside the structure during this step. Also, note that the material parameter $\boldsymbol{m}$ is fixed during this step.

Moving the design nodes in the tangent directions will not change the surface much. Consequently, each design node n is associated with one design variable only. A design variable x_n represents the distance node n that is moved in the normal direction $\boldsymbol{n}_n$ from the current position $\boldsymbol{p}_n$ as illustrated in Fig. 6. The design variables are assembled in the vector $\boldsymbol{x} = [\dots, x_n, \dots]^T$. Consequently, the positions of the control points as a function of the design variables can be expressed as $\boldsymbol{p}(\boldsymbol{x})$.

The relation between the current position $\boldsymbol{p}_n$, the optimized position $\boldsymbol{p}_n^*$ and the optimized design variable x_n^* for a design node n is

$$\boldsymbol{p}_n^* = \boldsymbol{p}_n(x_n^*) = \boldsymbol{p}_n + x_n^* \cdot \boldsymbol{n}_n \tag{9}$$

To estimate $\boldsymbol{x}^* = [\dots, x_n^*, \dots]^T$, a smooth non-linear optimization problem is solved:

$$\begin{aligned} \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} :& \quad f(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x})) = \boldsymbol{u}^T \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))\boldsymbol{u} \\ \text{subject to} :& \quad g_i(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x})) \le 0, \quad i = 1,2,3 \\ :& \, \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x})) \\ :& \, \boldsymbol{x}^{\min} \le \boldsymbol{x} \le \boldsymbol{x}^{\max} \end{aligned} \tag{10}$$

Here, $\boldsymbol{x}^{\min} = [\dots, x_n^{\min}, \dots]^T$ and $\boldsymbol{x}^{\max} = [\dots, x_n^{\max}, \dots]^T$ are move limits on the design variables $\boldsymbol{x}$. Generally, $\boldsymbol{x}^{\min}$ and $\boldsymbol{x}^{\max}$ are chosen such that the design nodes will not create degenerate tetrahedra during the optimization. Consequently, the new shape can only be a small perturbation from the current shape and Eq. (10) will be solved many times. Furthermore, the move limits ensure that the design nodes stay inside a user-specified design domain. Therefore, the structure cannot extend beyond the boundaries of this design domain.

We use the gradient-based optimization algorithm Method of Moving Asymptotes (MMA) [14] to solve the optimization problem in Eq. (10). This is an iterative optimization procedure which is stopped when the infinity norm of the change in $\boldsymbol{x}$ is less than a threshold or at iteration 5. In addition to evaluating the objective function and constraints, the derivatives of these functions with respect to each of the design variables x_n have to be evaluated at each iteration. Computing $(\partial/\partial x_n)\boldsymbol{u}$ is not efficient. However, using the adjoint variable method (utilizing the equilibrium equations) [43,44], we get an analytical expression for $(\partial/\partial x_n) f(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))$ without the problematic term $(\partial/\partial x_n)\boldsymbol{u}$:

$$\frac{\partial f(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))}{\partial x_n} = -\boldsymbol{u}^T \frac{\partial \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))}{\partial x_n} \boldsymbol{u} + 2\boldsymbol{u}^T \frac{\partial \boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p}(\boldsymbol{x}))}{\partial x_n} \tag{11}$$

Still, since the equilibrium equations have to be evaluated at each iteration, this continuous optimization step is the most expensive part of the optimization procedure.

2.3.2. Discrete optimization

In addition to changing the shape by moving the design nodes, a discrete optimization step is performed where the materials $\boldsymbol{m}$ are changed and the positions $\boldsymbol{p}$ are not. The step has two purposes; introducing holes inside the structure and increasing the convergence rate of the continuous optimization. The optimization problem can be written as

$$\begin{aligned} \boldsymbol{m}^* = \arg\min_{\boldsymbol{m}} &: f(\boldsymbol{m}, \boldsymbol{p}) = \boldsymbol{u}^T \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p})\boldsymbol{u} \\ \text{subject to} &: g_i(\boldsymbol{m}, \boldsymbol{p}) \le 0, \quad i = 1,2,3 \\ &: \boldsymbol{K}(\boldsymbol{m}, \boldsymbol{p})\boldsymbol{u} = \boldsymbol{f}(\boldsymbol{m}, \boldsymbol{p}) \\ &: m_e \in \{void, solid\} \end{aligned} \tag{12}$$

Note that the set of possible materials is limited to void and solid. However, it is possible to extend this approach to handle multiple materials. Furthermore, we choose that only solid elements are design elements. Consequently, this step only removes material from the structure. If it removes material near the surface, this will speed up shape changes. On the other hand, if it removes material inside the structure, a hole is created.

The discrete optimization problem in Eq. (12) is NP-hard. However, since this optimization problem is combined with a continuous optimization, it is not necessary to solve it to optimality. Consequently, this step will seek to improve the objective while trying to satisfy the constraints by relabeling tetrahedra. The relabeling will be based on discrete derivatives, i.e. the change in objective or constraints when changing the material in element e from solid to void:

$$\Delta_e f(\boldsymbol{m}, \boldsymbol{p}) = f(\boldsymbol{m}_e^v, \boldsymbol{p}) - f(\boldsymbol{m}, \boldsymbol{p}) \tag{13}$$

$$\Delta_e g_i(\boldsymbol{m}, \boldsymbol{p}) = g_i(\boldsymbol{m}_e^v, \boldsymbol{p}) - g_i(\boldsymbol{m}, \boldsymbol{p}), i = 1,2,3 \tag{14}$$

Here, $\boldsymbol{m}_e^v$ equals $\boldsymbol{m}$ where m_e is void instead of solid. However, computing these discrete derivatives for compliance is inefficient since the equilibrium equations then have to be evaluated once for each solid tetrahedron. Instead, we will use an approximation based on the theory of topological derivatives [4,5,45,6]. The topological derivative corresponds to the influence on the objective function of introducing an infinitesimal hole in element e. For compliance, the discrete derivative can therefore be approximated by

$$\Delta_e f(\boldsymbol{m}, \boldsymbol{p}) \approx 3\boldsymbol{u}^T \boldsymbol{K}_e(\boldsymbol{m}, \boldsymbol{p})\boldsymbol{u} - \frac{2V_e(\boldsymbol{p})}{N_{\in e}} \sum_{c \in e} \boldsymbol{u}_c^T \boldsymbol{g} \tag{15}$$

The first part of the optimization strategy is to improve the objective function while decreasing or satisfying all constraints. A constraint i is decreased if

$$\Delta_e g_i(\boldsymbol{m}, \boldsymbol{p}) \le 0 \tag{16}$$

and satisfied if

$$g_i(\boldsymbol{m}, \boldsymbol{p}) + \Delta_e g_i(\boldsymbol{m}, \boldsymbol{p}) \le 0 \tag{17}$$

Hence, a design element e is relabeled from solid to void if either of Eqs. (16) and (17) are satisfied for all constraints and

$$\Delta_e f(\boldsymbol{m}, \boldsymbol{p}) < 0 \tag{18}$$

The second part of the optimization is to try to improve constraints which are not satisfied. Therefore, if constraint i is not satisfied, i.e. $g_i(\boldsymbol{m}, \boldsymbol{p}) > 0$, we will try to find an optimal design element e^* to relabel from solid to void. Noting that $\Delta_e f(\boldsymbol{m}, \boldsymbol{p}) \ge 0$, the optimal design element e^* is found by solving

$$e^* = \arg\min_e - \frac{\Delta_e f(\boldsymbol{m}, \boldsymbol{p})}{\Delta_e g_i(\boldsymbol{m}, \boldsymbol{p})} \tag{19}$$

where all arguments e satisfy

$$\Delta_e g_i(\boldsymbol{m}, \boldsymbol{p}) < 0 \tag{20}$$

and either Eq. (16) or (17) for all constraints. Design element e^* is then relabeled from solid to void. This process is repeated as long as constraint i is not satisfied and an optimal element e^* exists.

2.4. Disconnected material

The continuous and discrete optimization steps can very well result in material which is disconnected from the main structure. These parts do not contribute to the objective. Furthermore, since void elements are eliminated from the finite element analysis, disconnected material will result in the equilibrium equations not having a unique solution. Consequently, disconnected material is removed by performing a connected component analysis and making every component, except for the largest, void.

2.5. Initialization

To initialize the optimization, the user has to specify boundary conditions, an objective function, constraints and an initial structure.

The boundary conditions are the supports and external forces applied to the surface of the structure as described in Section 2.2. Furthermore, the boundaries of the design domain (the domain where material can reside) have to be specified. Finally, it is possible to specify fixed domains (areas that are either always solid or always void). The fixed void areas are implemented as not being a part of the design domain. However, the fixed solid domains are enforced by assigning a different label to the tetrahedra inside these domains. Consequently, an invisible surface exists between the fixed and non-fixed solid domains. The shape of this surface should not be changed in any way. However, we still want the DSC method to improve the mesh quality and control the level of detail at this surface. Consequently, the DSC method is modified such that only mesh operations which do not change the surface are performed at the surface between fixed and non-fixed domains.

In all of the example problems presented here, the objective is to minimize compliance since it is often desirable to produce as stiff a structure as possible. However, choosing another objective is as simple as changing the objective function and calculating the shape and topological derivatives of the new function. For example, the same approach has been used for balancing of 3D models [46]. Furthermore, different problems require different constraints. In this paper, we present several different global constraints to illustrate their effect on the design. The effect can be quite drastic and consequently the constraints are as important as the objective.

Finally, the initial model is a triangle mesh. Consequently, any surface mesh can be used as a starting point for the optimization without any conversions. In this paper, we choose to initialize the optimization by triangle meshes of existing models and by generated meshes that fill the entire design domain.

2.6. Method summary

The method consists of two steps:

Step 1: *Discrete optimization*
Improves the objective as well as unsatisfied constraints by relabeling elements from solid to void based on their topological derivatives as described in Section 2.3.2. Then, removes disconnected material.

Step 2: *Continuous optimization*
Solves the optimization problem in Eq. (10) using the gradient-based optimization algorithm MMA (Section 2.3.1). MMA hereby estimates the optimal values of the design variables $\boldsymbol{x}^* = [\dots, x_n^*, \dots]^T$. Then, each design node n is moved from position $\boldsymbol{p}_n$ to $\boldsymbol{p}_n^* = \boldsymbol{p}_n + x_n^* \cdot \boldsymbol{n}_n$ using the DSC method as described in Section 2.1. Finally, disconnected material is removed.

These two steps make up one time step and are iterated until the changes on the surface from consecutive time steps are small.

Problems can arise if a volume or perimeter constraint is applied. The optimization will seek to obey the constraint before taking the objective into account. This can lead to undesired removal of material from places where it is necessary. Our solution to this problem is to gradually lower the constraint such that $V^*(t) = \max(\alpha^t, V^*)$ and $A^*(t) = \max(\beta^t, A^*)$ where t is the time step and $0 < \alpha < 1$ and $0 < \beta < 1$ are constants.

2.7. Efficiency

Efficiency is essential when performing topology optimization in 3D. A major piece of the puzzle to make this approach more efficient than standard fixed grid methods is to take advantage of the mesh adaptivity inherent to the DSC method. Consequently, the surface is represented by a fine discretization whereas large tetrahedra discretize parts far away from the surface. Furthermore, the main computational power should be used to achieve a fine resolution near the optimum. When the optimization is initialized by a 3D model, the optimum is assumed to be close. However, that is probably not the case when the optimization is initialized by filling the design domain with material. Consequently, in these cases, we slowly lower the discretization parameter δ by multiplying it by 0.99 at each time step. The detail control, described in Section 2.1, will then increase the mesh complexity. Note that this strategy is especially effective since the method only calculates on solid elements. However, solving the equilibrium equations is still the most time-consuming part. Consequently, we utilize multiple threads on the CPU to speed up these computations. Also, computing the gradients of the compliance function and assembling the global stiffness matrix K and force vector F are parallelized.

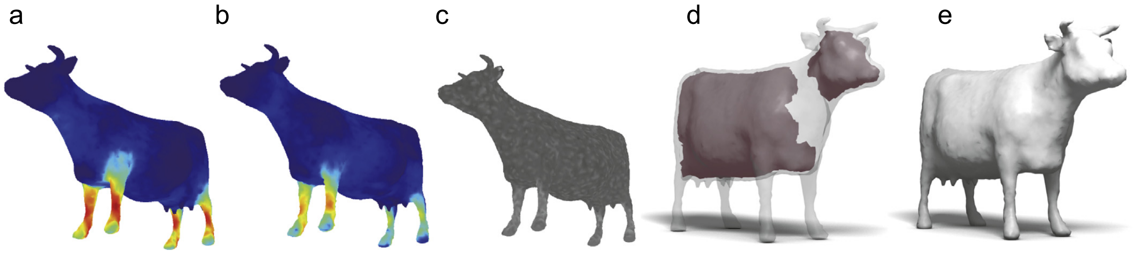
3. Results

The proposed method can be used in the fabrication design process in areas such as construction, manufacturing and design. In this section, we will illustrate this statement by solving problems within each of these fields. The results are generated on a laptop with a 2.4 GHz quad-core Intel Core i7 processor and 8 GB of 1333 MHz DDR3 RAM. Parameters and performance measures are depicted in Table 1. Furthermore, the objective of all examples is to minimize compliance subject to constraints as depicted in Table 1.

The raw surface triangle meshes of the optimized structures, i.e. the output as it looks from the optimization method, are visualized using Blender. No post-processing like subdivision and smoothing has been utilized to improve the appearance. Furthermore, when material has been removed from inside a structure, the internal cavities are visualized by making the structure transparent. In addition to the optimized result, we will in some cases visualize the strain energy density (SED) at the surface of the final model. The SED depicts how much strain an element at the surface is subjected to. Here, the jet colormap is used, where blue and red depict low and high SED respectively. Furthermore, the SEDs are scaled between the minimum and maximum SED of the initial structure. Consequently, this visualizes how the stiffness has changed as a consequence of the optimization. In the same cases, we will also visualize the difference from the original model by a grayscale colormap. Here, gray means no change, darker means it has moved in the negative normal direction and lighter

Table 1
Method parameters and performance measures for all example problems. The displayed values are the values as they appear after the optimization. The V^* and A^* values are stated in percent of the initial volume V^0 and surface area A^0 respectively whereas D^* and T^* are stated in percent of the discretization parameter δ. Furthermore, f^0 and f^* are initial and final compliance respectively. Finally, the # in the right-most column is the number of time steps.

Problem	δ mm	V^* (α)	A^* (β)	D^*	T^*	f^*/f^0	Surface	Complex	Running time
		% V^0 (–)	% A^0 (–)	% δ	% δ	–	# Faces	# Elements	Minutes (#)
Bridge	423	20 (0.96)	30 (0.98)	–	–	304%	9883	29,836	68 (70)
Statue	50	50 (0.95)	–	15	100	27%	35,868	66,314	275 (20)
Dinosaur	1.4	–	–	15	100	46%	6876	15,071	11 (5)
Armadillo	2.8	–	–	15	100	13%	9872	15,819	60 (50)
Table 1	42	15 (0.96)	30 (0.98)	–	–	2671%	5492	11,761	16 (100)
Table 2	62	15 (0.96)	35 (0.98)	–	–	964%	3543	5521	13 (60)
Table 3	42	15 (0.96)	30 (0.98)	–	–	5929%	5374	11,759	20 (100)
Chair 1	21	12.5 (0.96)	25 (0.98)	–	–	1199%	4413	7929	15 (100)
Chair 2	21	12.5 (0.96)	30 (0.98)	–	–	625%	5527	9026	18 (100)
Chair 3	27	12.5 (0.96)	30 (0.98)	–	–	927%	3382	4927	8 (75)
Support	655	20 (0.96)	20 (0.98)	–	–	17%	15,064	27,120	109 (100)



Initial SED Final SED Change Transparent statue Statue

Fig. 7. Topology optimized cow statue which show that the method can optimize stiffness while saving material.

that it has moved in the normal direction. The distance is scaled by the largest change.

3.1. Construction

Topology optimization has traditionally been used for construction where the objective is to save material while ensuring stiffness. The presented method has the same capabilities as previous methods. Furthermore, it extends those methods by being able to initialize an optimization by a surface triangle mesh with no conversion necessary.

First, a bridge problem is initialized by a steel cube ($30 \times 15 \times 12$ m^3) with a space for vehicles and supports as depicted in Fig. 1. The surface of the bridge is fixed and subjected to a distributed load pushing downwards (100 MPa). The result and optimization process are also depicted in Fig. 1. The result shows that compliance has increased to 304% of the initial value during the optimization process. However, the optimized structure only uses 20% of the material used by the initial structure.

Next, a 4 m-long concrete statue is initialized by a 3D model of a cow (source: Aim@Shape). The statue is solid concrete, only subjected to gravitational forces and supported underneath all of its hoofs. The change in SED, shape changes and the optimized cow statue are depicted in Fig. 7. This example shows that our method extends previous methods by being able to initialize an optimization by a 3D model (represented by a triangle mesh) without any conversion and, furthermore, remain close to this shape. Also, since the statue is subjected to gravitational forces only, compliance is improved at the same time as the amount of material is reduced.

3.2. Manufacturing

An important application of our method is as a tool to improve the stiffness of a given shape. Assume, we are given a 3D shape that is to be fabricated. The problem is to change the exterior shape as little as possible while using a minimum amount of material and ensuring that the fabricated object will be able to support itself and moreover withstand specified external loads. Furthermore, a side effect of optimizing a structure to bear its own weight is that the balance is improved.

A 10 cm-long plastic model of a dinosaur (source: Aim@Shape) is subjected to external forces (5 MPa) on the tail and the head where one would expect the model to be weakest. Furthermore, each of the four feet is supported. The SEDs, shape changes and optimized dinosaur are depicted in Fig. 8. Since the external forces are large compared to the gravitational forces, the optimization does not create any cavities. Instead, it redistributes material to places where it improves stiffness. Consequently, compliance is minimized to 46% of the initial value.

Next, a 10 cm-high plastic Armadillo model with a large head (source: Stanford University Computer Graphics Laboratory and edited in MeshMixer) is supported underneath both feet and only subject to gravity. The SEDs, shape changes and optimized model can be seen in Fig. 8. It is evident that since the model has a large head it will lean forward and thereby subject the shins to large strain. When optimizing compliance, the strain is minimized and the balance of the model is improved as a side effect. However, since imbalance is not directly penalized by the objective function, balance is not guaranteed. A modification of the objective function or constraints would, however, guarantee balance by requiring the center of gravity to stay within the convex hull of the supports.

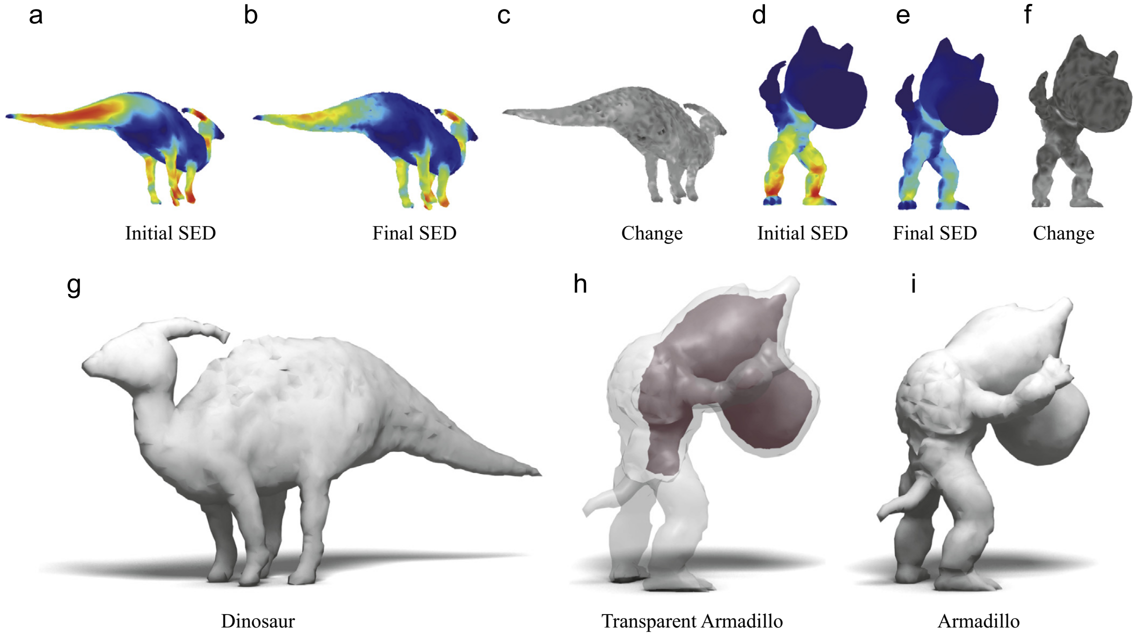


Fig. 8. Toy models optimized to improve both stiffness and balance while remaining close to the initial shape.

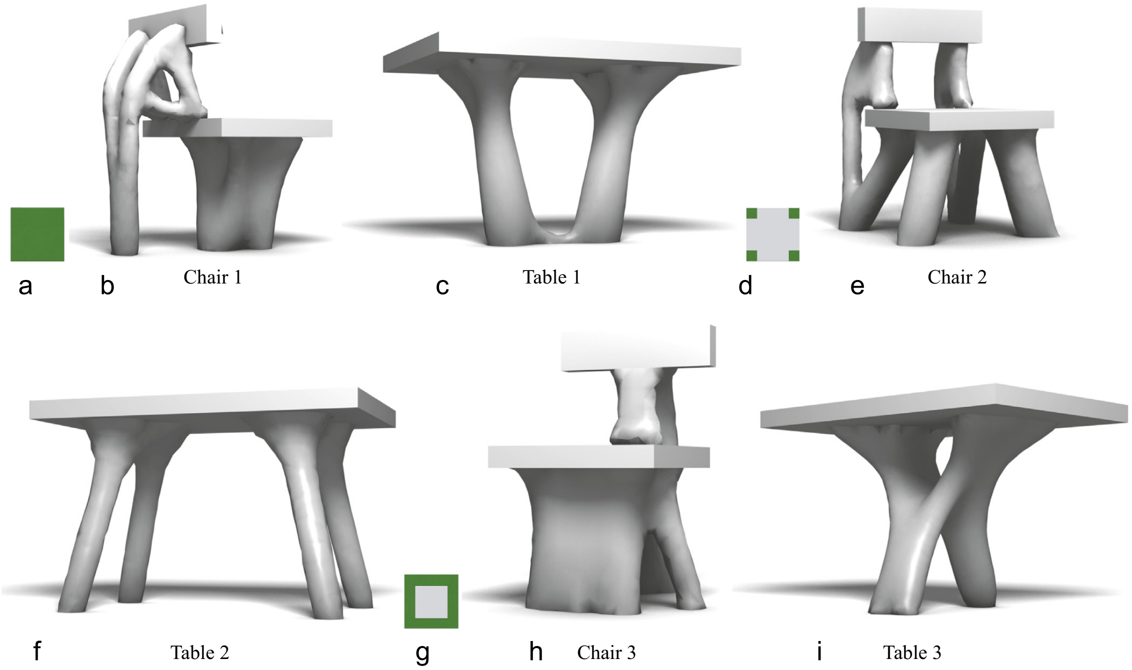


Fig. 9. Topology optimized tables and chairs which show the design capabilities of the suggested method. The difference between the problems is the supports (illustrated at the left of each row) and possibly the values of parameters. Note that the same illustration is used for both a table and a chair problem, therefore the dimensions of these illustrations are not correct.

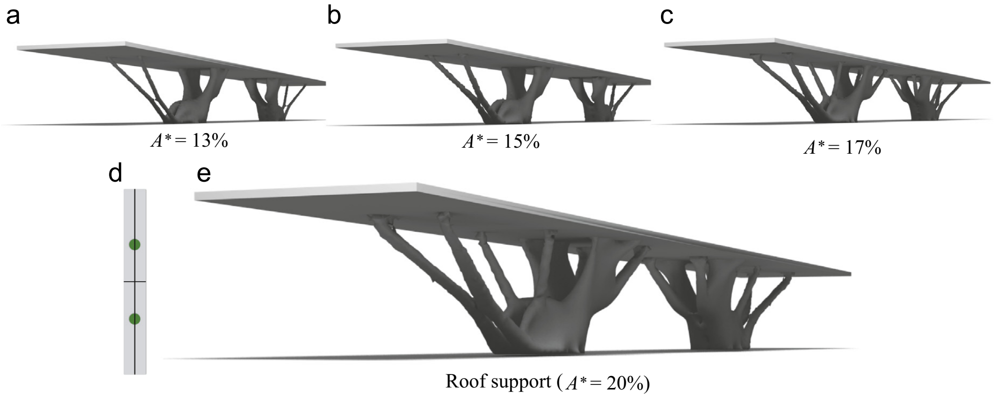


Fig. 10. Topology optimized roof support, optimized using different values for the perimeter constraint. This problem is inspired by the real world problem of supporting the roof of the Qatar National Convention Center. The supports are placed as depicted in Fig. 10(d) where also symmetry axes are visualized as black lines. (a) $A^* = 13\%$, (b) $A^* = 15\%$, (c) $A^* = 17\%$, (d) and (e) Roof support ($A^* = 20\%$).

3.3. Design

When humans design a given 3D object, the main concerns are often to satisfy aesthetic and functional requirements. Topology optimization is not concerned with aesthetics but it satisfies functional requirements. However, topology-optimized shapes exhibit an organic and sparse feeling that is often visually pleasing. Therefore, such a tool is useful as part of a design workflow [47]. Furthermore, the method can be used to generate significantly different designs by slight changes to the input. This is significantly simpler for a designer than remodeling a surface.

Three plastic tables are modeled by a fixed layer of material at the top of a design domain ($1.8 \times 1.2 \times 1.2\ \text{m}^3$) and a distributed load (2 MPa) pressing down on this layer. Furthermore, three chairs are initialized by filling a $0.6 \times 0.8 \times 0.6\ \text{m}^3$ design domain. The seat is modeled by a fixed void domain of size $0.4 \times 0.4 \times 0.4\ \text{m}^3$ and a fixed solid domain underneath which is subjected to a load (1 MPa). Finally, a backrest is modeled by a small fixed solid domain and subjected to a horizontal force (0.5 MPa). The difference between the problems is the position and extent of the supports. All supports are placed at the bottom of the design domain and have the shape depicted in Fig. 9(a)–(c) as seen from above.
The optimized designs are depicted in Fig. 9.

Finally, we will use the Qatar National Convention Center as an example of a real-world architectural design problem. The Convention Center has an impressive façade which is a roof supported by a concrete topology-optimized structure [47]. To model this, we take advantage of the symmetry and thereby only optimize a quarter of the structure (the symmetry axes are depicted in Fig. 10 (d)). Consequently, the problem is initialized by a $125 \times 20 \times 15\ \text{m}^3$ cube where the top layer (1 m) is fixed and solid. The structure is supported at the bottom in a half circular area (Fig. 10(d)) and only subjected to gravity. The result can be seen in Fig. 10(e) and, in addition, we illustrate in Fig. 10 the effect of changing the parameter for the perimeter constraint. Note that the result is not expected to look like the Convention Center since [47] use different boundary conditions and do not specify material, objective and constraints.

4. Conclusion

The presented method is the first to optimize both the 3D shape and topology of a surface triangle mesh without the use of an implicit representation. This is achieved by embedding the triangle mesh in a simplicial complex and using the Deformable Simplicial Complex method. Consequently, the method accepts a surface triangle mesh as input and outputs another surface triangle mesh which is only different from the input mesh where it has been optimized. Furthermore, as opposed to standard fixed grid methods, our method makes it possible to generate detailed designs within reasonable time on an ordinary laptop.

We have shown that the method automatically generates designs which satisfy some user-defined structural requirements. However, note that the search space is limited by global constraints and that there is no guarantee that the global optimum is reached. The bridge and the cow statue show that material can be saved where it is expensive or inconvenient while maintaining or improving stiffness. The dinosaur and Armadillo models show that 3D models automatically can be made stiffer and more balanced, while retaining the shape. Finally, the tables, chairs and roof support show that functional and, in our opinion, visually pleasing designs can be achieved with little effort from a designer. This is far from an exhaustive list of problems that can be solved using the presented method. As mentioned, topology optimization has been used to solve a wide variety of problems. To solve these or other problems, one only needs to model the boundary conditions and choose the objective, constraints and an initial structure. However, more advanced problems might require additional work. For example implementing additional objective functions and constraints, handling multiple load cases, using an anisotropic material model, handling dynamic problems and taking non-linearity into account.

We have shown that furniture and support structures for buildings can be modeled by specifying a few input parameters. Furthermore, both the input and output models are in the form of a surface triangle mesh. Consequently, this tool has potential to be used for modeling for films, videogames and other offline productions in addition to designing physical structures, especially if performance and user friendliness are improved. To increase performance, one idea is to take full advantage of the parallel nature of the finite element computations by, for example, feeding the computations to the GPU. Furthermore, parallelization of the DSC method would be beneficial. Another idea is to take even further advantage of the mesh adaptivity by lowering the discretization parameter more wisely. To increase the user friendliness, automatic determination of worst-case loads could be useful to limit the amount of user input. Also, finding an alternative to the perimeter constraint would be desirable since it can limit the optimization and its parameter is unintuitive and difficult to choose. Finally, most designers want to influence the design regularly during the design process. Therefore, a workflow which includes user feedback and post-processing is needed.

Acknowledgments

The authors appreciate the support from the Villum Foundation through the grant: “NextTop”.

References

[1] Bendsøe MP, Kikuchi N. Generating optimal topologies in structural design using a homogenization method. Comput Methods Appl Mech Eng 1988;71 (2):197–224.
[2] Bendsøe MP, Sigmund O. Topology optimization—theory, methods, and applications. second edition. Berlin: Springer Verlag; 2003.
[3] Misztal MK, Bærentzen JA. Topology adaptive interface tracking using the deformable simplicial complex. ACM Trans Graph 2012;31(3):24.
[4] Eschenauer HA, Kobelev VV, Schumacher A. Bubble method for topology and shape optimization of structures. Struct Multidiscip Optim 1994;8:42–51.
[5] Sokolowski J, Zochowski A. On the topological derivative in shape optimization. SIAM J Control Optim 1999;37(4):1251–72.
[6] Feijóo RA, Novotny AA, Taroco E, Padra C. The topological derivative for the poisson's problem. Math Models Methods Appl Sci 2003;13(12):1825–44.
[7] Garreau S, Guillaume P, Masmoudi M. The topological asymptotic for pde systems: the elasticity case. SIAM J Control Optim 2000;39(6):1756–78.
[8] de Gournay F, Allaire G, Jouve F. Shape and topology optimization of the robust compliance via the level set method. ESAIM: Control Optim Calc Var 2008;14:43–70.
[9] Le C, Bruns T, Tortorelli D. A gradient-based, parameter-free approach to shape optimization. Comput Methods Appl Mech Eng 2011;200(9–12):985–96.
[10] Arnout S, Firl M, Bletzinger K-U. Parameter free shape and thickness optimisation considering stress response. Struct Multidiscip Optim 2012;45(6):801–14.
[11] Ding Y. Shape optimization of structures: a literature survey. Comput Struct 1986;24(6):985–1004.
[12] Mohammadi B, Pironneau O. Applied shape optimization for fluids. Oxford: Oxford University Press; 2001.
[13] Bucur D, Buttazzo G. Variational methods in shape optimization problems, progress in nonlinear differential equations and their applications, Boston: Birkhäuser; 2005.
[14] Svanberg K. The method of moving asymptotes—a new method for structural optimization. Int J Numer Methods Eng 1987;24(2):359–73.
[15] Prévost R, Whiting E, Lefebvre S, Sorkine-Hornung O. Make it stand: balancing shapes for 3D fabrication. ACM Trans Graph (Proceedings of ACM SIGGRAPH) 2013;32(4):81:1–10.
[16] Skouras M, Thomaszewski B, Coros S, Bickel B, Gross M. Computational design of actuated deformable characters. ACM Trans Graph 2013;32(4):82:1–10.
[17] De Goes F, Alliez P, Owhadi H, Desbrun M. On the equilibrium of simplicial masonry structures. ACM Trans Graph 32 (4).
[18] Liu Y, Pan H, Snyder J, Wang W, Guo B. Computing self-supporting surfaces by regular triangulation. ACM Trans Graph 2013;32(4):92:1–10.
[19] Panozzo D, Block P, Sorkine-Hornung O. Designing unreinforced masonry models. ACM Trans Graph (Proceedings of ACM SIGGRAPH) 2013;32(4): 91:1–91:12.
[20] Umetani N, Schmidt R. Cross-sectional structural analysis for 3d printing optimization. In: SIGGRAPH Asia 2013 Technical Briefs, SA '13, ACM, New York, NY, USA, 2013. p. 5:1–5:04.
[21] Stava O, Vanek J, Benes B, Carr N, Měch R. Stress relief: improving structural strength of 3d printable objects. ACM Trans Graph 2012;31(4):48:1–48:11.
[22] Zhou Q, Panetta J, Zorin D. Worst-case structural analysis. ACM Trans Graph 2013;32(4):137:1–12.
[23] Bendsøe MP. Optimal shape design as a material distribution problem. Struct Optim 1989;1(4):193–202.
[24] Osher SJ, Fedkiw RP. Level set methods and dynamic implicit surfaces. 1st edition. New York: Springer; 2003.
[25] Wang M, Wang X, Guo D. A level set method for structural topology optimization. Comput Methods Appl Mech Eng 2003;192(1):227–46.
[26] Allaire G, Jouve F, Toader A-M. Structural optimization using sensitivity analysis and a level-set method. J Comput Phys 2004;194(1):363–93.
[27] Ha S-H, Cho S. Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh. Comput Struct 2008;86(13–14): 1447–1455.
[28] Allaire G, Dapogny C, Frey P. Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh. Comptes Rendus Math 2011;349(17–18):999–1003.
[29] Yamasaki S, Nomura T, Kawamoto A, Sato K, Nishiwaki S. A level set-based topology optimization method targeting metallic waveguide design problems. Int J Numer Methods Eng 2011;87(9):844–68.
[30] Xia Q, Shi T, Liu S, Wang MY. A level set solution to the stress-based structural shape and topology optimization. Comput Struct 2012;90–91(0):55–64.
[31] Allaire G, Dapogny C, Frey P. A mesh evolution algorithm based on the level set method for geometry and topology optimization. Struct Multidiscip Optim 2013;48(4):711–5.
[32] Maute K, Ramm E. Adaptive topology optimization. Struct Optim 1995;10: 100–112.
[33] Christiansen AN, Nobel-Jørgensen M, Aage N, Sigmund O, Bærentzen JA. Topology optimization using an explicit interface representation. Struct Multidiscip Optim 2014;49(3):387–99.
[34] Si H. TetGen: a quality tetrahedral mesh generator and a 3d delaunay triangulator, 2013. URL ⟨http://wias-berlin.de/software/tetgen/⟩.
[35] Parthasarathy VN, Graichen CM, Hathaway AF. A comparison of tetrahedron quality measures. Finite Elem Anal Des 1994;15(3):255–61.
[36] Field DA. Laplacian smoothing and delaunay triangulations. Commun Appl Numer Methods 1988;4(6):709–12.
[37] Bærentzen JA, Gravesen J, Anton F, Aanæs H. Guide to computational geometry processing: foundations, algorithms, and methods. London: Springer; 2012.
[38] Shewchuk JR. Two discrete optimization algorithms for the topological improvement of tetrahedral meshes; 2002 (unpublished manuscript).
[39] Cook RD, Malkus DS, Plesha ME, Witt RJ. Concepts and applications of finite element analysis. New York: John Wiley & Sons; 2007.
[40] Chen Y, Davis TA, Hager WW, Rajamanickam S. Algorithm 887: cholmod, supernodal sparse cholesky factorization and update/downdate. ACM Trans Math Softw 2008;35(3):22:1–14.
[41] Davis TA, Hager WW, Duff IS. SuiteSparse. URL ⟨http://www.cise.ufl.edu/research/sparse/SuiteSparse/⟩; 2013.
[42] Haber R, Jog C, Bendsøe M. A new approach to variable-topology shape design using a constraint on perimeter. Struct Optim 1996;11(1–2):1–12.
[43] Pironneau O. Optimal shape design for elliptic systems. In: Drenick R, Kozin F, editors. System modeling and optimization, Lecture notes in control and information sciences, vol. 38. Berlin, Heidelberg: Springer; 1982. p. 42–66.
[44] Christensen PW, Klarbring A. An introduction to structural optimization, solid mechanics and its applications, vol. 153. Ontario: Springer; 2008.
[45] Céa J, Garreau S, Guillaume P, Masmoudi M. The shape and topological optimizations connection. Comput Methods Appl Mech Eng 2000;188(4): 713–726.
[46] Asger Nyman Christiansen, Ryan Schmidt, J. Andreas Bærentzen. Automatic balancing of 3D models. Computer-Aided Design 2015;58(0):236–41 solid and Physical Modeling 2014.
[47] Sasaki M, Itō T, Isozaki A. Morphogenesis of flux structure. London: AA Publications; 2007.

Appendix C

Paper C

Christiansen, A. N., Tortorelli, D. A., Aage, N., and Sigmund, O. (2014d). Combined shape and topology optimization for minimization of maximal von Mises stress. (Submitted)

Structural and Multidisciplinary Optimization manuscript No.
(will be inserted by the editor)

Combined Shape and Topology Optimization for Minimization of Maximal von Mises Stress

Asger N. Christiansen · Daniel A. Tortorelli · Niels Aage · Ole Sigmund

October 20, 2014

Abstract This work shows that a combined shape and topology optimization method can produce 2D designs where large stress concentrations are avoided. The method represents the surface explicitly and discretizes the domain into a simplicial complex which adopts both structural shape and topology. It has previously been used to produce minimal compliance designs. Here, we minimize the, more challenging, maximal von Mises stress which requires regularization via a filter scheme.

Keywords Stress minimization · Combined shape and topology optimization · Explicit surface representation · Deformable Simplicial Complex method

Asger N. Christiansen
Department of Applied Mathematics and Computer Science, Technical University of Denmark, Asmussens Allé, B.303B, 2800 Kgs. Lyngby, Denmark
E-mail: asny@dtu.dk

Daniel A. Tortorelli
Department of Mechanical Engineering, University of Illinois at Urbana-Champaign, 1206 West Green Street, Urbana, IL 61801, USA

Niels Aage · Ole Sigmund
Department of Mechanical Engineering, Technical University of Denmark, Nils Koppels Allé, B.404, 2800 Kgs. Lyngby, Denmark

1 Introduction

The Deformable Simplicial Complex (DSC) method (Misztal and Bærentzen 2012) has recently been used in structural topology optimization where the objective was to minimize compliance subject to a volume constraint in 2D and 3D (Christiansen et al 2014a,b). However, in many applications, the interest is not to minimize compliance but rather to avoid large stress concentrations which lead to structural fatigue and fracture. Consequently, we will apply the same 2D scheme as Christiansen et al (2014b) but for minimizing the maximal von Mises stress subject to a volume constraint[1].

One should note that minimizing the maximal von Mises stress is more difficult than compliance due to the 'singularity' phenomenon, its local nature and its highly non-linear behaviour (Le et al 2010). Nevertheless, minimizing the maximal von Mises stress has been treated in shape optimization (e.g. Francavilla et al (1975); Changwen and Minghua (1990); Le et al (2011)). Furthermore, a maximum von Mises stress constraint has been enforced in topology optimization problems (e.g. Duysinx and Sigmund (1998); Svanberg and Werme (2007); Bruggi and Venini (2008); Le et al (2010)). In this work, we minimize the maximal von Mises stress of a structure by changing both its shape and topology.

The design domain is discretized into non-overlapping triangular elements (a simplicial complex) that are either labelled solid (filled with material) or void (filled with air). Consequently, the structural surface is represented explicitly by the collection of edges which are sandwiched between a void and a solid element. Even though the structure is represented explicitly (as opposed to e.g. the density method (Bendsøe 1989; Bendsøe

[1] Presented at the 4th International Conference on Engineering Optimization (EngOpt 2014).

and Sigmund 2003) and level set method (Wang et al 2003; Allaire et al 2004)), the DSC method is able to accommodate large shape and topology changes while maintaining well-shaped triangular elements. It does so by continuously performing local mesh operations. Furthermore, this approach uses a single representation as opposed to combined implicit/explicit approaches, e.g. Allaire et al (2013).

The shape and topology of the structure are changed by a discrete and continuous optimization step.The discrete step relabels elements from solid to void based on an approximate optimization procedure. If the relabelled elements reside at the surface of the structure, the shape is changed and if they reside inside the structure, the topology is changed. The continuous step uses shape derivatives to find an improved position for each of the surface nodes which is close to the current position. The surface nodes are then moved to these improved positions by the DSC method. If two surfaces collide during the shape deformations, the surfaces will be merged together by the DSC method. Therefore, the continuous step may change the topology in addition to the shape. These two optimization steps are repeated until convergence.

To generate a differentiable cost function, the p-norm is used as an approximation to the maximum function (Duysinx and Sigmund 1998). This, in combination with the finite element discretization, gives rise to a non-optimal jagged surface during the continuous optimization step. To remedy this, Gaussian filtering is applied to the surface nodes. Finally, a global constraint is enforced to ensure that the structure stays inside the design domain during the continuous optimization step.

The method is described in detail in Section 2. In section 3, we apply the suggested approach to two 2D problems, the L-shaped cantilever beam and the portal. The results show that the maximal stress is significantly reduced compared to the corresponding compliance optimized designs. Finally, we summarize our findings in Section 4.

2 Method

2.1 Discretization

The proposed method uses a simplicial complex to represent the structure, i.e. it discretizes the design domain into triangular elements as seen in Figure 1. All triangular elements e have an associated material m_e which is either void (no material) or solid (filled with material). Therefore, the interface between solid and void (the surface) is represented by the collection of element edges that are sandwiched between a triangle labeled void and a triangle labeled solid. In addition to serving as the geometric representation, the solid elements of this discretization are used for the finite element analysis (FEA). To ensure an accurate analysis, it is important to sustain a high quality mesh with no degenerate triangles.

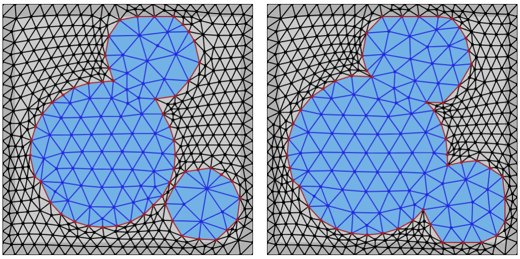

(a) Time step 1 (b) Time step 2

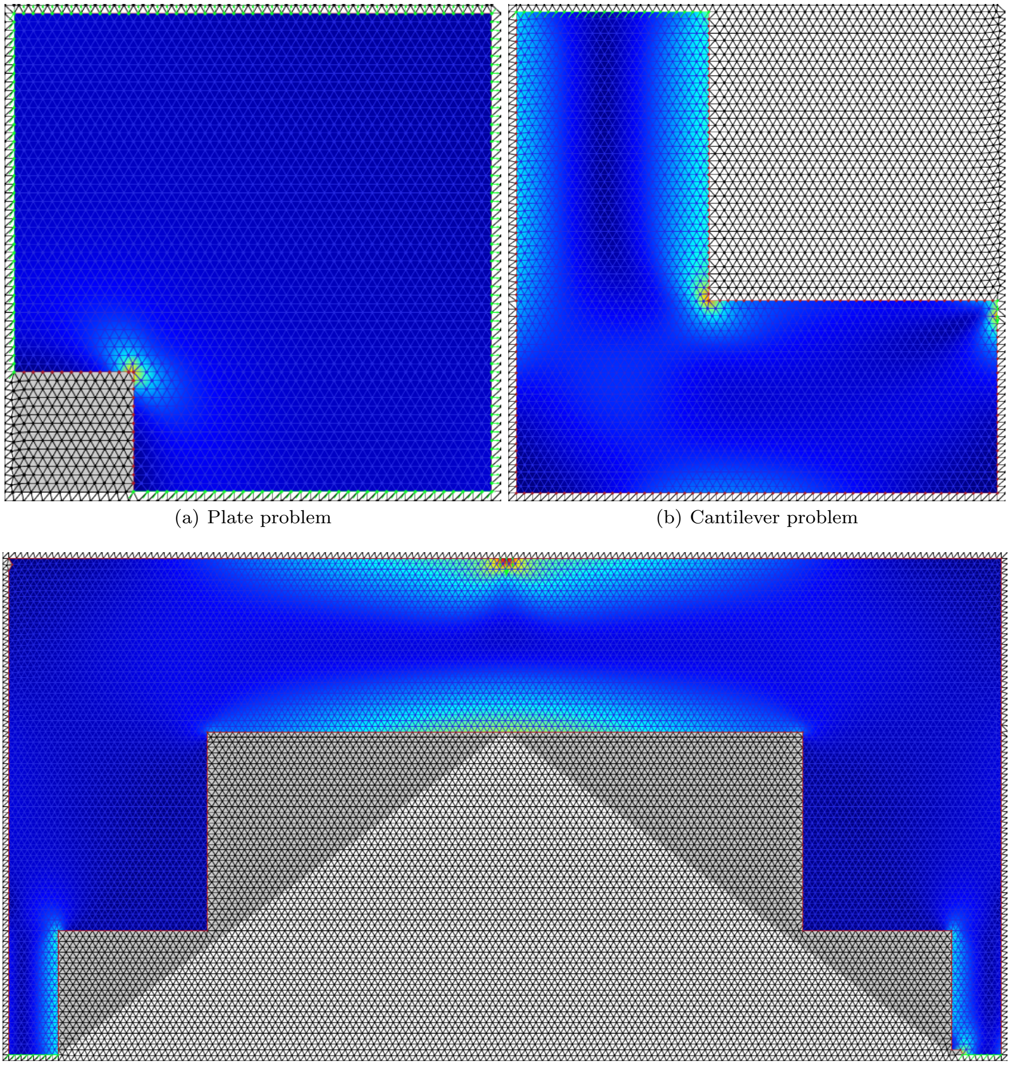
Fig. 2 The DSC method changes the topology.

2.2 Deformable Simplicial Complex method

To ensure a high quality mesh, the Deformable Simplicial Complex (DSC) method (Misztal and Bærentzen 2012)[2] is used. The DSC method maintains high quality triangular elements as the mesh deforms. It removes low quality triangles (caps and needles) by performing mesh operations such as Laplacian smoothing, edge flip, vertex insertion and vertex removal (Christiansen et al 2014b). Note that the DSC method only improves the mesh quality where necessary (often near the surface) and, as a consequence, is much faster than a complete remeshing. Furthermore, the DSC method performs topology changes by collapsing low quality triangles which are sandwiched between two surfaces as illustrated in Figure 2.

In addition to ensuring high quality elements, the DSC method also controls the level of detail of both the surface and the mesh. In practice, the DSC method collapses small triangles and splits large triangles. Consequently, we always attain a mesh of the desired complexity, described by the discretization parameter δ (corresponding to the average edge length). More importantly, the detail control potentially allows for mesh adaptivity so that smooth regions can be represented by a coarser discretization than regions with small scale features or large stress concentrations.

[2] An open-source framework is available at `www.github.com/asny/2D-DSC`



(a) Plate problem

(b) Cantilever problem

(c) Portal problem

Fig. 1 Initial discretization of the hole in a plate, L-shaped cantilever and portal problems. The 'jet' color map is used to visualize the von Mises stress across the structure.

2.3 Optimization

The objective of the optimization is to minimize the maximal von Mises stress of the solid elements

$$f(\boldsymbol{m},\boldsymbol{p}) = \max_{e} \sigma_e(\boldsymbol{m},\boldsymbol{p}) \tag{1}$$

subject to a volume constraint

$$g_1(\boldsymbol{m},\boldsymbol{p}) = \frac{V(\boldsymbol{m},\boldsymbol{p})}{V^*} - 1 \leq 0 \tag{2}$$

where $V(\boldsymbol{m},\boldsymbol{p})$ is the volume of the structure and V^* is the maximum volume allowed. Both objective and constraint are functions of the materials $\boldsymbol{m} = [\ldots, m_e, \ldots]^T$ associated with each triangular element e and the positions $\boldsymbol{p} = [\ldots, \boldsymbol{p}_n^T, \ldots]^T$ associated with each design node n, i.e. each surface node that is not subjected to external forces.

To evaluate the element von Mises stress, we first solve the equilibrium equations

$$\boldsymbol{K}(\boldsymbol{m},\boldsymbol{p})\boldsymbol{u} = \boldsymbol{f} \tag{3}$$

where $\boldsymbol{K}$ is the global stiffness matrix and $\boldsymbol{f}$ is the global load vector. Note that we do not include internal forces (e.g. gravity) which means that $\boldsymbol{f}$ is independent of the design variables. The von Mises stress at the centroid of the solid element e can then be evaluated by

$$\sigma_e(\boldsymbol{m},\boldsymbol{p}) = \sqrt{\boldsymbol{u}_e^T \boldsymbol{B}_e(\boldsymbol{p})^T \boldsymbol{E}_e(\boldsymbol{m}) \boldsymbol{A} \boldsymbol{E}_e(\boldsymbol{m}) \boldsymbol{B}_e(\boldsymbol{p}) \boldsymbol{u}_e} \tag{4}$$

Here, $\boldsymbol{E}_e$ is the constitutive matrix, $\boldsymbol{B}_e$ is the strain-displacement matrix which is evaluated at the element centroid, $\boldsymbol{u}_e$ is the element nodal displacement vector and $\boldsymbol{A}$ is given by

$$\boldsymbol{A} = \begin{bmatrix} 1 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix} \tag{5}$$

Since the max function is not smooth, it is replaced by the differentiable p-norm

$$\max_e \sigma_e(\boldsymbol{m},\boldsymbol{p}) \approx \left(\sum_e \sigma_e(\boldsymbol{m},\boldsymbol{p})^p \right)^{\frac{1}{p}} \tag{6}$$

with a relatively large value of p as also proposed by Duysinx and Sigmund (1998).

The optimization problem is solved by alternating between optimizing the materials $\boldsymbol{m}^*$ and the positions $\boldsymbol{p}^*$ while the others are fixed. These two optimization steps – the discrete and the continuous – are iterated until convergence.

The structure is initialized by filling the design domain as seen in Figure 1. Consequently, the optimization makes large changes in the first iterations to satisfy the volume constraint. This leads to structures that are not connected to the loads and supports. Therefore, the volume constraint V^* is lowered gradually during the optimization as proposed by Christiansen et al (2014b).

2.4 Discrete optimization

The discrete optimization step has two purposes; introducing holes inside the structure and increasing the speed of the optimization. This step determines an optimal material m_e^* for each element e. In this study, we limit the set of possible materials to void and solid. Furthermore, we only remove material in this step. If the current material m_e is solid and the optimal material m_e^* is void, the label of that triangle is changed from solid to void. If the effected element is adjacent to the surface, this will alter the shape. On the other hand, if the effected element is inside the structure, then a hole appears changing the topology

The discrete optimization problem is formulated as

$$\begin{aligned} \boldsymbol{m}^* = \arg\min_{\boldsymbol{m}} &: f(\boldsymbol{m}) = \left(\sum_e \sigma_e(\boldsymbol{m})^p \right)^{\frac{1}{p}} \\ \textit{subject to} &: g_1(\boldsymbol{m}) = \frac{V(\boldsymbol{m})}{V^*} - 1 \leq 0 \\ &: \boldsymbol{K}(\boldsymbol{m})\boldsymbol{u} = \boldsymbol{f} \\ &: m_e \in \{void, solid\} \end{aligned} \tag{7}$$

This problem is computationally expensive to solve to optimality, consequently, we approximate its solution.

To determine the appropriate changes in materials, we rely on 'discrete derivatives', i.e. the change in the objective and constraint functions when the material in element e is changed from solid m_e^s to void m_e^v. This is expressed by

$$\begin{aligned} \varDelta_e f(\boldsymbol{m}) &= f(\boldsymbol{m}_e^v) - f(\boldsymbol{m}) \\ \varDelta_e g_1(\boldsymbol{m}) &= g_1(\boldsymbol{m}_e^v) - g_1(\boldsymbol{m}) = -\frac{V_e(\boldsymbol{m})}{V^*} \end{aligned} \tag{8}$$

where $\boldsymbol{m}_e^v$ equals $\boldsymbol{m}$ with m_e^s replaced by m_e^v. The discrete derivatives of the volume constraint can be computed whereas computing the discrete derivatives of the objective function is computationally expensive since it requires an evaluation of the equilibrium equations per solid element.

Although we tested more elaborate approaches for evaluating the stress derivatives $\varDelta_e f(\boldsymbol{m})$, we ended up obtaining the most stable procedure by implementing a simple 'fully stressed design' approach. Although known to give wrong results if used on its own, its combination with the continuous optimization step produces designs that appear to be optimal with respect to the stress objective. Hence, we approximate the discrete derivative by the element von Mises stress, i.e.

$$\varDelta_e f(\boldsymbol{m}) \approx \sigma_e(\boldsymbol{m}) \tag{9}$$

For the compliance minimization results in Section 3, we approximate the discrete derivative via the element strain energy density

$$\varDelta_e f(\boldsymbol{m}) \approx \boldsymbol{u}_e^T \boldsymbol{K}_e(\boldsymbol{m}) \boldsymbol{u}_e \tag{10}$$

as in Christiansen et al (2014b).

The discrete optimization step addresses the volume constraint if it is not satisfied. First, the discrete derivatives $\varDelta_e f(\boldsymbol{m})$ and $\varDelta_e g_1(\boldsymbol{m})$ for each element e are evaluated. Then, while $g_1(\boldsymbol{m}) > 0$, we identify an optimal design element e^* to relabel from solid to void that has a minimal effect on the objective function f and maximal negative effect on the volume constraint g_1. Therefore, e^* is the element that solves

$$e^* = \arg\min_e \frac{\Delta_e f(\boldsymbol{m})}{-\Delta_e g_1(\boldsymbol{m})} = \arg\min_e \frac{\sigma_e(\boldsymbol{m})}{V_e(\boldsymbol{m})} \tag{11}$$

Design element e^* is then relabeled from solid to void and this process is repeated until the volume constraint is satisfied. Note that only one FEA is performed during the discrete optimization step and it is therefore relatively fast.

This is an improved strategy compared to Christiansen et al (2014b) which did not take the constraints into account in the discrete optimization step. Still, the proposed method uses a heuristic optimization algorithm and, consequently, it cannot guarantee convergence to optimality. However, when the parameter of the volume constraint V^* is lowered to its final value and the volume constraint is satisfied, the discrete optimization step has no effect. Hence, we conjecture that when this process is combined with the continuous optimization step which is guaranteed to converge, the entire optimization converges.

2.5 Continuous optimization

The continuous optimization step deforms the structure to a more optimal shape which is within a small perturbation of its current shape. The design variables are the positions $\boldsymbol{p}_n$ of each design node n, i.e. a surface node that is not subjected to any external loads. We realize that moving a design node in the tangent direction will only change the surface slightly. Consequently, to reduce the number of design variables, we only perturb the design nodes in their normal direction such that their new position is given by

$$\boldsymbol{p}_n(x_n) = \boldsymbol{p}_n^0 + x_n \cdot \boldsymbol{n}_n \tag{12}$$

Here, $\boldsymbol{p}_n^0$ is the initial position of node n, x_n is the design variable and $\boldsymbol{n}_n$ is the normal to node n.

Note that, if we allow support nodes to move away from the support, the optimization algorithm will do so. Therefore, support nodes are perturbed along the support instead of normal to it as illustrated in Figure 3. This direction is still denoted by $\boldsymbol{n}_n$. Furthermore, moving a supported node which is not at the boundary of the support does not change the shape or support. Therefore, the interior support nodes are not design nodes and their positions are not optimized.

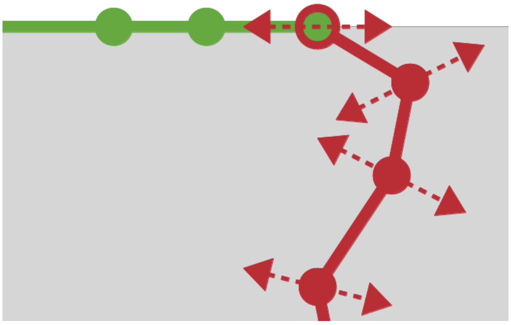


Fig. 3 Supported nodes (green) are not perturbed during the continuous optimization step except the nodes at the boundary of the support (green and red). However, these nodes are moved along the support rather than normal to it, opposed to the unsupported nodes (red).

The design variables are assembled in a vector $\boldsymbol{x} = [\ldots, x_n, \ldots]^T$ and the optimization problem is formulated as

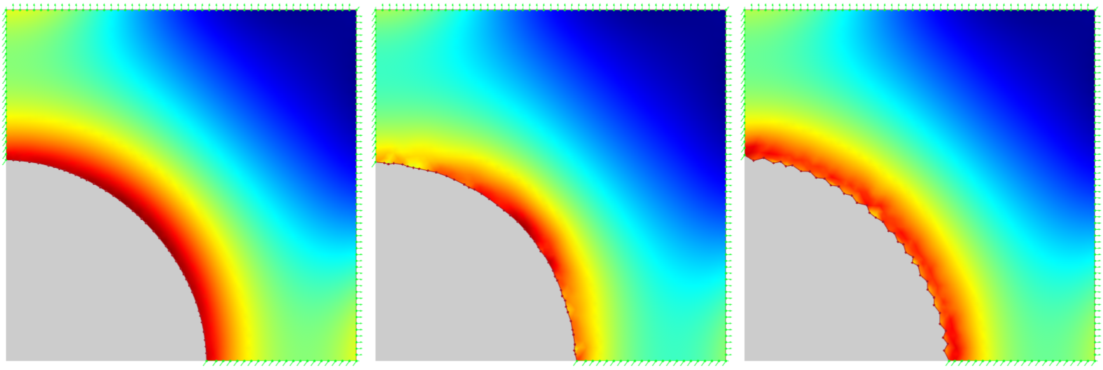
$$\begin{aligned} \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} &: f(\boldsymbol{x}) = \left(\sum_e \sigma_e(\boldsymbol{x})^p\right)^{\frac{1}{p}} \\ \textit{subject to} &: g_1(\boldsymbol{x}) = \frac{V(\boldsymbol{x})}{V^*} - 1 \leq 0 \\ &: \boldsymbol{K}(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{f} \\ &: \boldsymbol{x}^{\min} \leq \boldsymbol{x} \leq \boldsymbol{x}^{\max} \end{aligned} \tag{13}$$

Here, $\boldsymbol{x}^{\min}$ and $\boldsymbol{x}^{\max}$ are lower and upper limits on the design variables $\boldsymbol{x}$ which are prescribed such that triangles do not degenerate during the optimization. Consequently, the shape is only perturbed slightly during each continuous optimization step.

This smooth non-linear optimization problem is solved for $\boldsymbol{x}^* = [\ldots, x_n^*, \ldots]^T$ using the gradient-based optimization algorithm Method of Moving Asymptotes (MMA) (Svanberg 1987). We compute the necessary derivatives of the objective and constraint functions with respect to the position $\boldsymbol{p}_n$ of each design node n using the adjoint method. For the p-norm, the adjoint problem becomes

$$\begin{aligned} &\boldsymbol{K}(\mathbf{p})\boldsymbol{\lambda} = \\ &- f(\mathbf{p})^{1-p} \cdot \sum_e \sigma_e(\mathbf{p})^{p-2} \cdot \boldsymbol{B}_e(\mathbf{p})^T \boldsymbol{EAEB}_e(\mathbf{p})\boldsymbol{u}_e \end{aligned} \tag{14}$$

where $\sum_e$ here refers to the FE assembly operator. After solving Equation 14 for $\boldsymbol{\lambda}$, the gradient is computed



(a) Compliance ($\max_e \sigma_e = 1.000$)

(b) p-norm stress ($p = 1$, $\max_e \sigma_e = 1.112$)

(c) p-norm stress ($p = 8$, $\max_e \sigma_e = 1.044$)

Fig. 4 Hole in a plate designs by continuous optimization for different objectives. It is evident that minimizing the maximal von Mises stress through a p-norm approximation results in a jagged surface.

from

$$\frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} = \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{K}(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\boldsymbol{u} - \boldsymbol{\lambda}^T \frac{\partial \boldsymbol{f}(\boldsymbol{p})}{\partial \boldsymbol{p}_n} + f(\boldsymbol{p})^{1-p} \cdot \sum_e \frac{\sigma_e(\boldsymbol{p})^{p-2}}{2} \cdot \left(\boldsymbol{u}_e^T \boldsymbol{B}_e(\boldsymbol{p})^T \boldsymbol{EAE} \frac{\partial \boldsymbol{B}_e(\boldsymbol{p})}{\partial \boldsymbol{p}_n}\boldsymbol{u}_e + \boldsymbol{u}_e^T \frac{\partial \boldsymbol{B}_e(\boldsymbol{p})}{\partial \boldsymbol{p}_n}^T \boldsymbol{EAEB}_e(\boldsymbol{p})\boldsymbol{u}_e \right) \tag{15}$$

We then use the chain rule to compute the gradient with respect to each design variable, e.g.

$$\frac{\partial f(\boldsymbol{x})}{\partial x_n} = \frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} \cdot \frac{\partial \boldsymbol{p}_n}{\partial x_n} = \frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} \cdot \boldsymbol{n}_n \tag{16}$$

After the optimized design variables $\boldsymbol{x}^*$ are computed, the optimized positions are obtained from Equation 12 and the shape is deformed using the DSC method. It is noted that since the DSC method removes thin regions and closes small holes, this continuous step can also change topology.

2.6 Regularization

The governing equations are solved by the finite element method using quadratic element shape functions. This element choice has previously prevented the appearance of jagged surfaces during the continuous optimization step when minimizing compliance (Christiansen et al 2014b). However, when minimizing the p-norm of von Mises stress, the jagged surfaces reappear. Designs with jagged surfaces are unwanted since the stresses predicted by the FEA are not accurate.

Figure 4 illustrates the issue by shape optimizing a hole in a plate subject to biaxial loads, i.e. this is only a continuous optimization problem. The problem is initialized as seen in Figure 1(a) and a maximum volume constraint V^* of 80% of the initial volume is applied. When minimizing compliance, the result is a smooth round hole, which we know is optimal from analytical theory. However, when minimizing the p-norm of the von Mises stress with $p = 1$, the surface is less smooth and when using $p = 8$ it is jagged.

To alleviate the jaggedness, we filter the design variables $\boldsymbol{x}$ to get the filtered design variables $\tilde{\boldsymbol{x}} = [\ldots, \tilde{x}_n, \ldots]^T$ (inspired by Le et al (2011)). First, the design variables are converted to positions by Equation 12. Then, the positions $\boldsymbol{p}_n$ of each design node n are filtered to produce the filtered positions $\tilde{\boldsymbol{p}}_n$ via

$$\tilde{\boldsymbol{p}}_n = \frac{\sum_i h_r(|\boldsymbol{p}_n - \boldsymbol{p}_i|_2) \cdot \boldsymbol{p}_i}{\sum_i h_r(|\boldsymbol{p}_n - \boldsymbol{p}_i|_2)} \tag{17}$$

where $|\boldsymbol{p}_n - \boldsymbol{p}_i|_2$ is the Euclidean distance and the chosen kernel function h_r is a Gaussian, i.e.

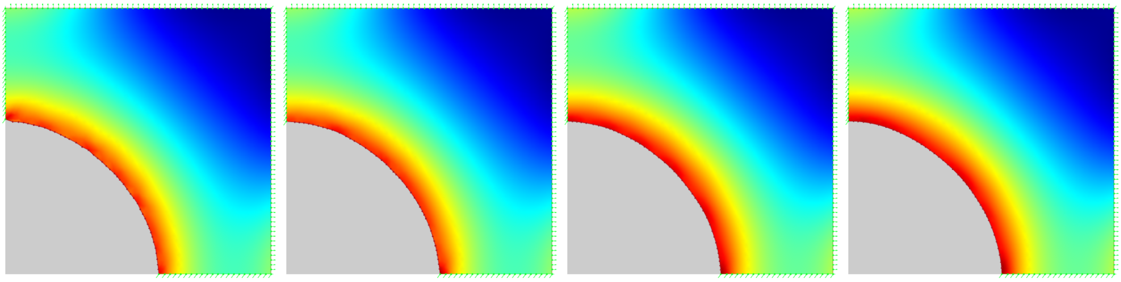
$$h_r(d) = \begin{cases} e^{-3 \cdot d^2/r^2} & \text{if } d \leq r \\ 0 & \text{if } d > r \end{cases} \tag{18}$$

where r is the filter radius. This filter is used to generate the filter matrix $\boldsymbol{H}$ such that

$$\tilde{\boldsymbol{p}} = \boldsymbol{H}\boldsymbol{p} \tag{19}$$

where $\tilde{\boldsymbol{p}} = [\ldots, \tilde{\boldsymbol{p}}_n^T, \ldots]^T$. Note that the above filter matrix is only computed when initializing the continuous optimization step. Finally, the positions are converted back to the filtered design variables

$$\tilde{x}_n = (\tilde{\boldsymbol{p}}_n - \boldsymbol{p}_n) \cdot \boldsymbol{n}_n \tag{20}$$



(a) $r = 1.5 \cdot \delta$, $\max_e \sigma_e = 1.098$ (b) $r = 2.5 \cdot \delta$, $\max_e \sigma_e = 1.089$ (c) $r = 3.5 \cdot \delta$, $\max_e \sigma_e = 1.046$ (d) $r = 4.5 \cdot \delta$, $\max_e \sigma_e = 1.042$

Fig. 5 Effect of the filter radius when minimizing the p-norm von Mises stress (using $p = 8$) for the hole in a plate problem.

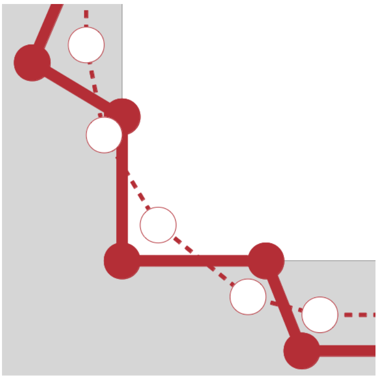


Fig. 6 Surface (solid red) movements outside the design domain (gray) after it has been filtered (white).

by the chain rule. Now, the structural surface is defined by the filtered design variables $\tilde{\boldsymbol{x}}$ rather than $\boldsymbol{x}$ so that $f(\boldsymbol{x})$ becomes $f(\tilde{\boldsymbol{x}})$ and $g_1(\boldsymbol{x})$ becomes $g_1(\tilde{\boldsymbol{x}})$.

This substitution requires a modification to the sensitivity analysis. We have that

$$\frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}} = \left(\frac{\partial \tilde{\boldsymbol{p}}}{\partial \boldsymbol{p}}\right)^T \frac{\partial f(\tilde{\boldsymbol{p}})}{\partial \tilde{\boldsymbol{p}}} = \boldsymbol{H}^T \frac{\partial f(\tilde{\boldsymbol{p}})}{\partial \tilde{\boldsymbol{p}}} \tag{21}$$

so that

$$\frac{\partial f(\boldsymbol{x})}{\partial x_n} = \frac{\partial f(\boldsymbol{p})}{\partial \boldsymbol{p}_n} \frac{\partial \boldsymbol{p}_n}{\partial x_n} = \left(\boldsymbol{H}^T \frac{\partial f(\tilde{\boldsymbol{p}})}{\partial \tilde{\boldsymbol{p}}}\right)_n \cdot \boldsymbol{n}_n \tag{22}$$

where

$$\begin{aligned} \frac{\partial f(\tilde{\boldsymbol{p}})}{\partial \tilde{\boldsymbol{p}}} &= [\ldots, \frac{\partial f(\tilde{\boldsymbol{p}})}{\partial \tilde{\boldsymbol{p}}_n}^T, \ldots]^T \\ &= [\ldots, \frac{\partial f(\tilde{\boldsymbol{x}})}{\partial \tilde{x}_n} \cdot \boldsymbol{n}_n^T, \ldots]^T \end{aligned} \tag{23}$$

Objective	p	r	$\boldsymbol{u}^T\boldsymbol{K}\boldsymbol{u}$	$\max_e \sigma_e$	$(\sum_e \sigma_e^8)^{\frac{1}{8}}$
Compliance	-	0	1.000	1.000	2.073
p-norm stress	1	0	1.001	1.112	2.081
p-norm stress	8	0	1.003	1.044	2.065
p-norm stress	8	$1.5 \cdot \delta$	1.000	1.098	2.077
p-norm stress	8	$2.5 \cdot \delta$	1.000	1.089	2.065
p-norm stress	8	$3.5 \cdot \delta$	1.000	1.046	2.079
p-norm stress	8	$4.5 \cdot \delta$	1.000	1.042	2.075

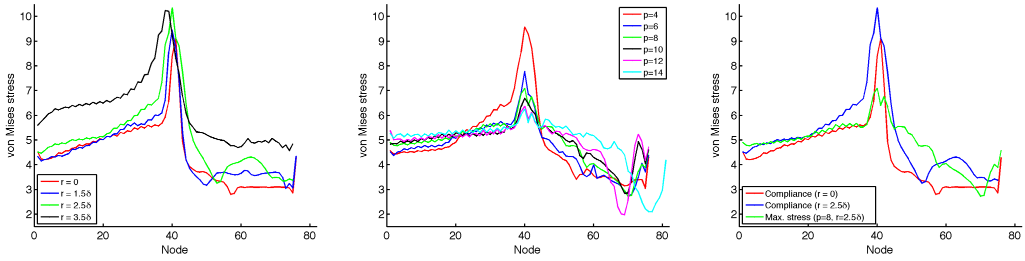
Table 1 Scaled compliance $\boldsymbol{u}^T\boldsymbol{K}\boldsymbol{u}$, maximal von Mises stress $\max_e \sigma_e$ and p-norm von Mises stress $(\sum_e \sigma_e^8)^{\frac{1}{8}}$ values for the hole in a plate problem from figures 4 and 5.

Unfortunately, the filter may cause the surface to move outside the design domain during the continuous optimization step when the design domain is non-convex, cf. Figure 6. To solve this problem, we apply a global constraint which penalizes regions of the structure that move outside of the design domain. The constraint is

$$g_2(\boldsymbol{p}) = \frac{1}{\delta^2} \cdot \left(\sum_n d(\boldsymbol{p}_n, \mathit{\Omega})^2\right) - \epsilon \leq 0 \tag{24}$$

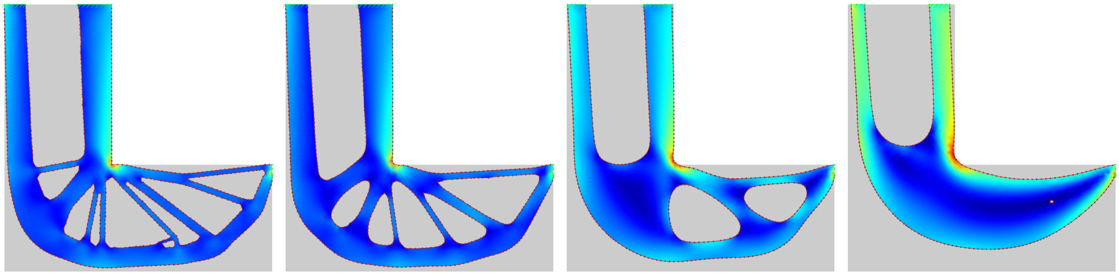
where $d(\boldsymbol{p}_n, \mathit{\Omega})$ is the Euclidean distance from position $\boldsymbol{p}_n$ to the design domain $\mathit{\Omega}$. This constraint limits the sum of squared distances from the design nodes to the design domain to be less than $\epsilon \cdot \delta^2$ where ϵ is an acceptable tolerance.

To test the proposed filtering scheme, we again solve the hole in a plate problem for different values of the filter radius r, cf. Figure 5. It is seen that applying the filter results in a relatively smooth surface for $r \geq 2.5 \cdot \delta$. However, when looking closely at the stress designs, it is seen that the results are not as smooth as the compliance design. Table 1 shows that all values of r result in compliance values which are close to optimum and maximal von Mises stress values which are slightly larger than for the compliance optimized structure. Furthermore, the maximal von Mises stress is reduced as



(a) (b) (c)

Fig. 7 Distribution of von Mises stress in the nodes from the support to the load along the corner for the designs seen in figures 8 and 9.



(a) $r = 0$, $\max\limits_e \sigma_e = 1.000$ (b) $r = 1.5 \cdot \delta$, $\max\limits_e \sigma_e = 1.028$ (c) $r = 2.5 \cdot \delta$, $\max\limits_e \sigma_e = 0.982$ (d) $r = 3.5 \cdot \delta$, $\max\limits_e \sigma_e = 0.949$

Fig. 8 Effect of applying a Gaussian filtering when minimizing compliance for the L-shaped cantilever problem.

r is increased, i.e. as the surface is further smoothed. It is clear that despite smoothing, the minimum stress problem is a much harder and more sensitive problem than the minimum compliance problem. Hence we do not expect to get the same smooth and consistent results as Christiansen et al (2014b). We want the filter to ensure a smooth surface while effecting the optimization process minimally, so we choose $r = 2.5 \cdot \delta$ in the following.

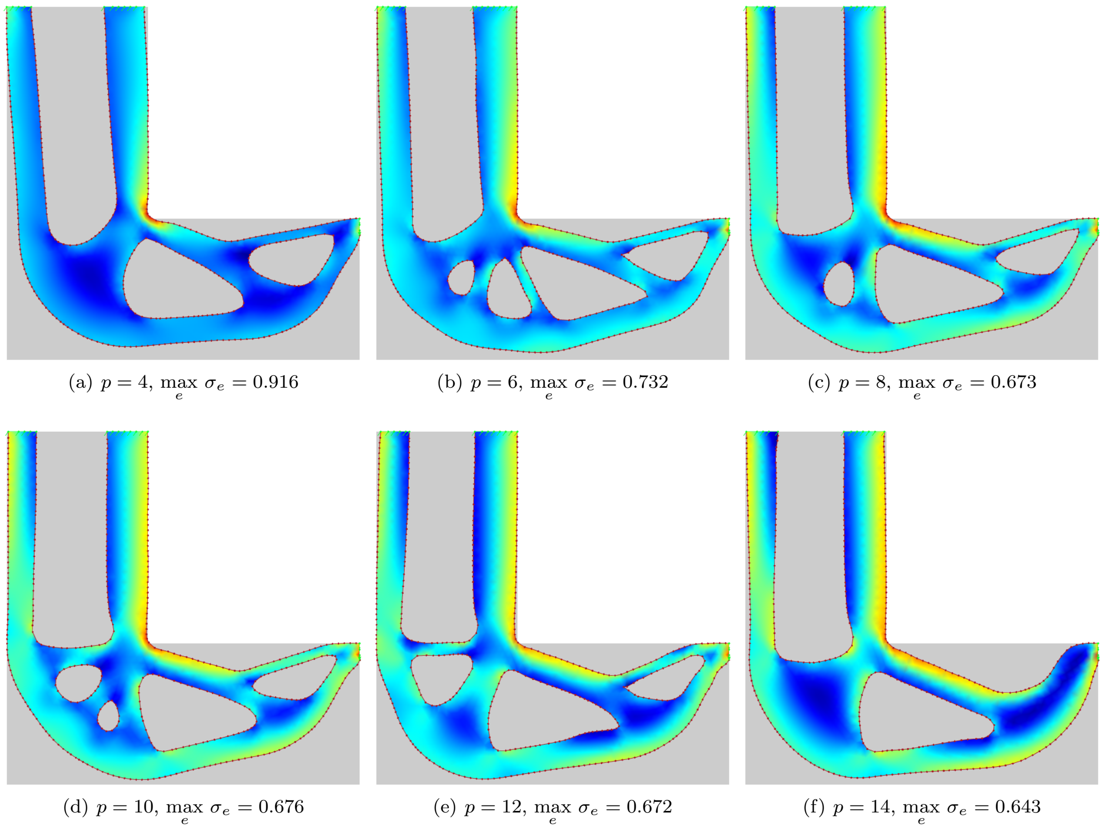
Objective	p	r	$\boldsymbol{u}^T\boldsymbol{K}\boldsymbol{u}$	$\max\limits_e \sigma_e$	$(\sum_e \sigma_e^8)^{\frac{1}{8}}$
Compliance	-	0	1.000	1.000	1.082
Compliance	-	$1.5 \cdot \delta$	1.022	1.028	1.111
Compliance	-	$2.5 \cdot \delta$	1.118	0.982	1.170
Compliance	-	$3.5 \cdot \delta$	1.353	0.949	1.259
p-norm stress	4	$2.5 \cdot \delta$	1.119	0.916	1.106
p-norm stress	6	$2.5 \cdot \delta$	1.054	0.732	0.949
p-norm stress	8	$2.5 \cdot \delta$	1.101	0.673	0.949
p-norm stress	10	$2.5 \cdot \delta$	1.110	0.676	0.936
p-norm stress	12	$2.5 \cdot \delta$	1.143	0.672	0.936
p-norm stress	14	$2.5 \cdot \delta$	1.177	0.643	0.954

Table 2 Scaled compliance $\boldsymbol{u}^T\boldsymbol{K}\boldsymbol{u}$, maximal von Mises stress $\max\limits_e \sigma_e$ and p-norm von Mises stress $(\sum_e \sigma_e^8)^{\frac{1}{8}}$ values for the L-shaped cantilever problem from figures 8 and 9.

3 Results

We now minimize the maximal von Mises stress using the proposed approach. In all figures, the von Mises stress has been scaled between its minimum and maximum and visualized by the 'jet' color map. Furthermore, it has been interpolated across elements to obtain a smooth visualization. The compliance values presented in the tables are scaled by compliance of the compliance optimized design (using $r = 0$). Also, the maximal stress and p-norm stress values are scaled by the maximal stress of the compliance optimized design.

We first study the L-shaped cantilever problem (Duysinx and Bendsøe 1998) which is initialized as depicted in Figure 1(b) and subject to a volume constraint V^* of 50% of the initial volume. For this problem, we will display graphs showing the von Mises stress in the nodes from the support to the load along the corner in Figure


(a) $p = 4$, $\max_e \sigma_e = 0.916$

(b) $p = 6$, $\max_e \sigma_e = 0.732$

(c) $p = 8$, $\max_e \sigma_e = 0.673$

(d) $p = 10$, $\max_e \sigma_e = 0.676$

(e) $p = 12$, $\max_e \sigma_e = 0.672$

(f) $p = 14$, $\max_e \sigma_e = 0.643$

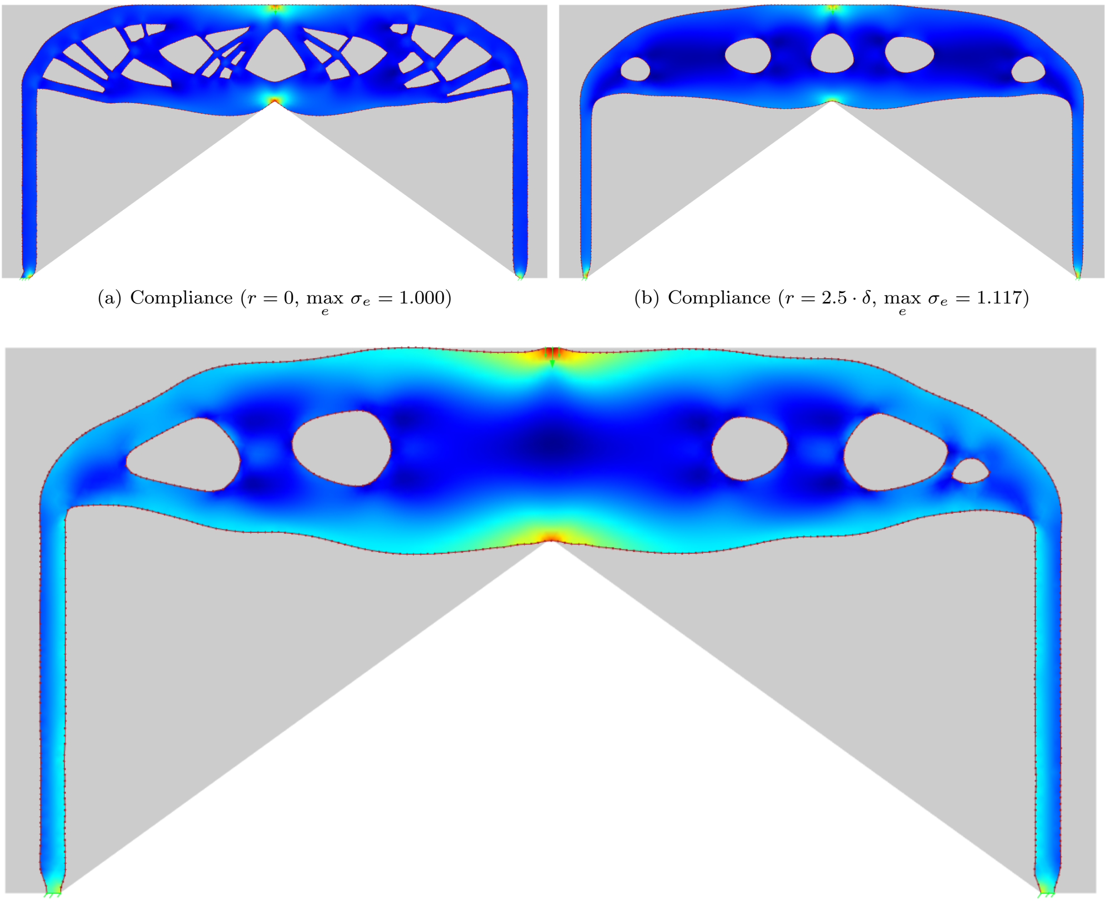
Fig. 9 Effect of changing the p-value when minimizing the p-norm von Mises stress of the L-shaped cantilever beam problem with fixed filter radius $r = 2.5 \cdot \delta$.

7. Here, the von Mises stress of a node is the average of the centroidal von Mises stress of the neighbouring elements. In these plots, we observe some wiggling. This is explained by the fact that we measure element stress in their centroids, meaning that elements sharing one or two nodes with the surface will have slightly different stress levels. This could potentially be avoided by a more elaborate stress recovery technique.

We compare designs minimizing the p-norm von Mises stress versus the compliance objective. Figures 8 and 7(a) and Table 2 present the effect of applying filtering when minimizing compliance. It is evident that applying a filter results in smoother surfaces with fewer holes and that it increases compliance. Furthermore, it is seen that applying a filter results in a slight decrease in the maximal von Mises stress which is attributed to the rounding of the corner where the von Mises stress is largest.

Next, we investigate the effect of the parameter p with fixed $r = 2.5 \cdot \delta$ as seen in figures 9 and 7(b) and Table 2. We see a more uniform distribution of von Mises stress and a reduced maximal von Mises stress as p is increased. However, for $p \geq 8$, the maximal von Mises stress is not improved significantly. Furthermore, the compliance increases as we increase p. Consequently, we use $p = 8$ in the remainder of this paper.

As expected, we achieve more uniformly distributed stresses when minimizing the p-norm von Mises stress versus minimizing compliance as seen in figures 8, 9 and 7(c). The results in Table 2 confirm that the maximal stress is reduced to 67.3% relative to the compliance optimized design. Furthermore, the maximal stress is reduced to 68.5% relative to the compliance based design when using a filter (with $r = 2.5 \cdot \delta$). It is also evident that compliance values for the stress based designs are comparable to those of the compliance based



(a) Compliance ($r = 0$, $\max_e \sigma_e = 1.000$)

(b) Compliance ($r = 2.5 \cdot \delta$, $\max_e \sigma_e = 1.117$)

(c) p-norm stress ($r = 2.5 \cdot \delta$, $p = 8$, $\max_e \sigma_e = 0.731$)

Fig. 10 Portal problem designs when using compliance (with and without filter) and the p-norm von Mises stress (with $p = 8$) as objective.

Objective	p	r	$\boldsymbol{u}^T\boldsymbol{K}\boldsymbol{u}$	$\max_e \sigma_e$	$(\sum_e \sigma_e^8)^{\frac{1}{8}}$
Compliance	-	0	1.000	1.000	1.202
Compliance	-	$2.5 \cdot \delta$	1.123	1.117	1.238
p-norm stress	8	$2.5 \cdot \delta$	1.147	0.731	0.955

Table 3 Scaled compliance $\boldsymbol{u}^T\boldsymbol{K}\boldsymbol{u}$, maximal von Mises stress $\max_e \sigma_e$ and p-norm von Mises stress $(\sum_e \sigma_e^8)^{\frac{1}{8}}$ values for the portal designs in Figure 10.

designs. This indicates that one can reduce stress levels without significant deterioration of compliance.

Finally, we present designs for the portal problem (Le et al 2010). We initialize the problem as seen in Figure 1(c) and use a volume constraint V^* of 50% of the initial volume. Again, Figure 10 and Table 3 illustrate that the von Mises stress is more uniformly distributed and that the maximal von Mises stress is decreased to 73.1% for the stress based versus the compliance based design. However, we also see that the maximal stress is increased by 12.3% for the compliance based design when a filter is applied compared to not using a filter. This is due to very thin supports in the filtered design which cause the maximal stress concentrations to be relocated from near the load and the kink to near the supports. For the p-norm stress based design, the supports are again widened to avoid large stress concentrations near the supports.

4 Conclusion

We have shown that the shape and topology compliance minimization method proposed by Christiansen et al (2014b) can be used to minimize the maximal von Mises stress. However, regularization is now needed to achieve a smooth design. Although this limits the search space of possible designs, it is important in order to ensure an accurate stress prediction by the FEA.

Subjects for future work are to extend this approach – which has already been used to generate 3D minimal compliance designs (Christiansen et al 2014a) – to 3D and to incorporate the topological derivatives to improve the convergence rate of the discrete optimization step. Finally, we note that the use of more advanced stress recovery schemes may improve convergence and consistency of the obtained results.

Acknowledgements

The authors appreciate the support from the Villum Foundation through the grant: ”NextTop”. Also, we would like to express our gratitude to Andreas Bærentzen and Morten Nobel-Jørgensen for their assistance.

References

Allaire G, Jouve F, Toader AM (2004) Structural optimization using sensitivity analysis and a level-set method. Journal of Computational Physics 194(1):363–393

Allaire G, Dapogny C, Frey P (2013) A mesh evolution algorithm based on the level set method for geometry and topology optimization. Structural and Multidisciplinary Optimization 48(4):711–715

Bendsøe MP (1989) Optimal shape design as a material distribution problem. Structural Optimization 1(4):193–202

Bendsøe MP, Sigmund O (2003) Topology Optimization - Theory, Methods, and Applications, second edition edn. Springer Verlag, Berlin

Bruggi M, Venini P (2008) A mixed FEM approach to stress-constrained topology optimization. International Journal for Numerical Methods in Engineering 73(12):1693–1714

Changwen X, Minghua Y (1990) Shape optimization of structures to minimize stress concentration. Computers & Structures 36(3):491 – 497

Christiansen AN, Bærentzen JA, Nobel-Jørgensen M, Aage N, Sigmund O (2014a) Combined shape and topology optimization of 3D structures. Computers & Graphics (0):–, Shape Modeling International 2014

Christiansen AN, Nobel-Jørgensen M, Aage N, Sigmund O, Bærentzen JA (2014b) Topology optimization using an explicit interface representation. Structural and Multidisciplinary Optimization 49(3):387–399

Duysinx P, Bendsøe MP (1998) Topology optimization of continuum structures with local stress constraints. International Journal for Numerical Methods in Engineering 43(8):1453–1478

Duysinx P, Sigmund O (1998) New developments in handling optimal stress constraints in optimal material distributions. In: Conference Proceedings of 7th Symposium in Multidisciplinary Analysis and Optimization, AIAA/USAF/NASA/ISSMO, St. Louis Missouri, USA, vol 3, pp 1501–1509

Francavilla A, Ramakrishnan CV, Zienkiewicz OC (1975) Optimization of shape to minimize stress concentration. The Journal of Strain Analysis for Engineering Design 10(2):63–70

Le C, Norato J, Bruns T, Ha C, Tortorelli D (2010) Stress-based topology optimization for continua. Structural and Multidisciplinary Optimization 41(4):605–620

Le C, Bruns T, Tortorelli D (2011) A gradient-based, parameter-free approach to shape optimization. Computer Methods in Applied Mechanics and Engineering 200(9-12):985–996

Misztal MK, Bærentzen JA (2012) Topology adaptive interface tracking using the deformable simplicial complex. ACM Transactions on Graphics 31(3):No. 24

Svanberg K (1987) The method of moving asymptotes – a new method for structural optimization. International Journal for Numerical Methods in Engineering 24(2):359–373

Svanberg K, Werme M (2007) Sequential integer programming methods for stress constrained topology optimization. Structural and Multidisciplinary Optimization 34(4):277–299

Wang M, Wang X, Guo D (2003) A level set method for structural topology optimization. Computer Methods in Applied Mechanics and Engineering 192(1):227–246

Appendix D

Paper D

Christiansen, A. N., Schmidt, R., and Bærentzen, J. A. (2014c). Automatic balancing of 3D models. *Computer-Aided Design*, 58(2015):236 – 241. Solid and Physical Modeling 2014

Computer-Aided Design 58 (2015) 236–241

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad

Automatic balancing of 3D models[☆]

Asger Nyman Christiansen [a,*], Ryan Schmidt [b], J. Andreas Bærentzen [a]

[a] *Technical University of Denmark, Denmark*
[b] *Autodesk Research, Canada*

HIGHLIGHTS

- We revisit a number of 3D print technologies and discuss their characteristics.
- We present an automatic, optimization based method for balancing 3D models.
- The balance is improved by creating internal cavities and by rotating the model.
- We pay special attention to make FDM printed models stand.

ARTICLE INFO

Keywords:
Rationalization
3D printing
Shape and topology optimization
Deformable Simplicial Complex method

ABSTRACT

3D printing technologies allow for more diverse shapes than are possible with molds and the cost of making just one single object is negligible compared to traditional production methods. However, not all shapes are suitable for 3D print. One of the remaining costs is therefore human time spent on analyzing and editing a shape in order to ensure that it is fit for production. In this paper, we seek to automate one of these analysis and editing tasks, namely improving the balance of a model to ensure that it stands. The presented method is based on solving an optimization problem. This problem is solved by creating cavities of air and distributing dense materials inside the model. Consequently, the surface is not deformed. However, printing materials with significantly different densities is often not possible and adding cavities of air is often not enough to make the model balance. Consequently, in these cases, we will apply a rotation of the object which only deforms the shape a little near the base. No user input is required but it is possible to specify manufacturing constraints related to specific 3D print technologies. Several models have successfully been balanced and printed using both polyjet and fused deposition modeling printers.

1. Introduction

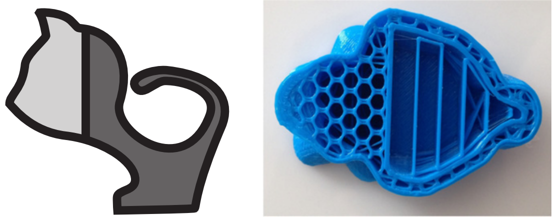
Injection molding has been one of the important pillars of mass production throughout the twentieth century, continuing to this day. It is a method that allows us to create vast numbers of plastic parts each of which takes mere fractions of a second to produce. Nevertheless, recent years have seen a growing excitement around a number of other fabrication technologies referred to as additive manufacturing, or simply 3D print. While these processes are very different, they tend to share the common trait that they are far slower than molding when many objects are to be made but much faster at producing a single object since no mold is needed. Another important advantage of 3D print is that we are generally quite unconstrained when it comes to what shapes that can be produced, the main restriction being on the size of the object. This is in stark contrast to objects produced using a mold since we have to be able to extract the object from the mold. Thus, as the speed of 3D printing increases, we are likely to face a future with much more variety in the shapes of manufactured objects.

Because of the variety of shapes and the low cost of producing few objects, the time consuming part shifts from manufacturing to modeling, and from producing the object to designing the object. Furthermore, it is often desired that the designed model has suitable geometric characteristics. Recent years have seen quite a few examples of work related to the aspect of making a 3D shape suitable for fabrication, a process known as *rationalization* in architecture. In this paper, we are specifically concerned with automatically ensuring that objects are balanced and thus able to stand without support after production.

[☆] This paper has been recommended for acceptance by Dr. Vadim Shapiro.

[*] Corresponding author. Tel.: +45 45255984.
E-mail address: asny@dtu.dk (A.N. Christiansen).

http://dx.doi.org/10.1016/j.cad.2014.07.009
0010-4485/© 2014 Elsevier Ltd. All rights reserved.



(a) Illustration. (b) Example.

Fig. 1. Figure (a) illustrates the situation when using an FDM 3D printer to produce a model with internal cavities. The thick black lines are the shell of the 3D model which is printed solid. The dark gray regions are infill whereas the light gray region is a cavity. Although such a cavity could be printed empty, in practice it may contain support structures with up to 20% aggregate density. Figure (b) shows an example, with 30% hexagonal-pattern infill and 10% support structures inside an interior cavity.

We make the following contributions.

1. We revisit a number of 3D print technologies and discuss their characteristics and affordances and how these pertain to the problem of producing objects that are balanced.
2. We present an automatic, optimization based method for balancing 3D models. The 3D model is embedded in an adaptive tetrahedral mesh. The balance is then improved by creating internal cavities and by rotating the model around its base. Apart from rotation the exterior of the model is not changed.
3. While this method may be used to balance 3D objects regardless of the production method, we pay special attention to fused deposition modeling (FDM). FDM is a common, cheap technology with characteristics that would confound a method that did not take these characteristics into account.

Prévost et al. [1] proposed a technique with the same capabilities as the presented method. However, it differs in nearly all particulars. With the presented method, cavities are generated by relabeling tetrahedra and moving internal surface nodes rather than labeling fixed cuboid voxels. Furthermore, Prévost et al. allow the user to equip the model with deformation handles used to perform automatic, affine transformations of parts of the model. While this strategy appears effective, it can make quite noticeable changes to the shape compared to a rotation around the base of the model which we propose.

2. 3D printing

The majority of 3D printing or *additive manufacturing* (AM) technologies operate by sequentially accumulating thin parallel layers of material in a vertical direction. As noted above, this provides great freedom in terms of 3D shape complexity. However, each different mechanism for realizing 3D printing involves quite different capabilities and constraints. These constraints are highly relevant to any algorithm which will attempt to alter a shape to satisfy some fabrication goal. Hence, we will review some properties relevant to the problem of making a shape stand.

The simplest case is when the model is printed completely solid. In this case, to balance a shape without deforming it our only recourse is to leave internal cavities. With printing technologies such as laser sintering (SLS), powder-bed, or stereolithography (SLA), printed *support* material will be trapped in any internal voids, and so escape holes (in some cases of considerable size) must be inserted into the model surface, or the model must be printed in parts and assembled. Each strategy is tedious and becomes increasingly intractable as the internal cavities grow in complexity.

A more complicated case is fused-deposition modeling (FDM), in which a thin stream of thermoplastic is extruded from a moving print head. This is the most common type of 3D printer today, in part because FDM has been rapidly commoditized in consumer hardware (Makerbot, RepRap, etc.). When using FDM, we have three types of regions (see Fig. 1):

1. Shell: exterior and interior surfaces are printed solid.
2. Infill: the interior is printed with a sparse pattern.
3. Cavity: internal cavities may be empty, or may contain support structure.

Although many FDM printers use a single material, they do not print in uniform density. To save material and print time, FDM printers generally print an outer *shell* several layers thick, and then fill the rest of the model with a sparse *infill* pattern. Printing internal *cavities* is also more complex with FDM printers. For most non-trivial objects, at some layers of the in-progress print there will be floating components which lack a direct connection to the print bed. Something must hold up each of these components, lest they succumb to gravitational forces. In FDM printing this is accomplished by adding *support* structures to the model. In addition to local height-minima, with FDM it is also necessary to support any parts of the model that have too shallow a *draft angle* relative to the print bed, as overlapping layers of the filament stream must have a sufficient area underneath them. Areas without adequate support will *droop*, which affects print quality and can even result in print failures. Generally, FDM support structures are snapped off after printing, but with internal cavities the support cannot be removed. Hence, internal cavities may have non-zero density. Since the density depends on the shape of the cavity, and on the particular support strategy in use, modeling it accurately is quite complex.

Clearly, to balance a 3D object, it is critical to take the difference in density between infill and cavity into account. Further compounding this issue is that internal cavities are also surrounding by solid shells, so adding a cavity can actually result in a local increase in density. This complicates both the analysis and optimization, and the previous work has not taken this variable density into account [1].

3. Method

In the following, we formulate the goal of balancing a 3D model, while making as few changes to the surface as possible, as an optimization problem. Consequently, the method is fully automatic. We will apply two optimization strategies. The first optimization strategy, hollowing (Section 3.1), creates cavities filled with a lighter or heavier material, for example air or copper inside a plastic model. Hollowing can also be used to simulate the infill and support structures created by FDM printers. Furthermore, it improves balance and does not deform the surface of the model. However, most 3D printers can only print in one material, or multiple materials with approximately the same density, and often cavities of air is not enough to make the model stand. Consequently, in these cases, a rotation around the base of the model will be applied (Section 3.2). The same rotation is applied to all surface nodes except the nodes which touch the ground. Therefore, the only deformation of the model will be close to the ground.

We assume that the initial 3D model is represented by a triangle surface mesh. Then, both the inside and the outside of the model are discretized into tetrahedral elements using TetGen [2]. Here, the tetrahedra do not overlap and each point in the domain is either inside or on the boundary between tetrahedra. Consequently, the mesh, illustrated in Fig. 2, is a simplicial complex. Furthermore, each tetrahedron has an associated material. The surface is then represented by the faces shared by two tetrahedra labeled with different materials. Therefore, the original and unchanged triangle mesh is embedded as a sub-complex in the tetrahedral mesh.

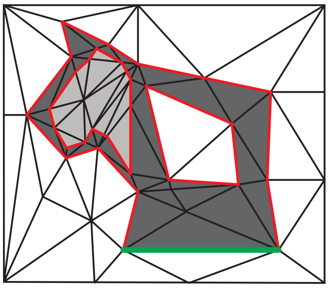


Fig. 2. 2D illustration of the discretization. The piecewise linear curves in red represent the surfaces which are sandwiched between two different materials (depicted in white, light gray and dark gray). Furthermore, the support surface, the surface between the model and the ground, is depicted in green. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this paper.)

To be able to introduce cavities and deform a 3D model while maintaining its tetrahedral mesh representation, we will use the Deformable Simplicial Complex (DSC) method [3]. The DSC method is a Lagrangian method for deformable surfaces, i.e. surface vertices are moved directly to deform the model. It then performs local updates to the tetrahedral mesh, moving, introducing, removing and reconnecting non-surface vertices to improve the quality. Thereby, the DSC method allows surfaces to deform while maintaining a sound tetrahedral mesh, even if topology changes occur, e.g. if two surfaces collide. Furthermore, the tetrahedra can be labeled with any number of labels which we will use to represent multiple materials. We also make use of the possibility to change the surface in discrete steps by relabeling tetrahedra. In 2D, the DSC method has previously been used for topology optimization [4], but minimizing compliance instead of improving balance.

To be able to model the balance of the shape, we need the contact surface between the model and the ground. We call this the support surface and it consists of a set of faces F_s on the surface of the model. The ground is represented by a plane spanned by the x- and z-axis such that the faces on the bottom of the shape coincide with this plane (see Fig. 2). We will then define the support point $\boldsymbol{s}$ as

$$\boldsymbol{s} = \frac{1}{\sum\limits_{f\in F_s} A_f} \sum_{f\in F_s} A_f \cdot \boldsymbol{c}_f \tag{1}$$

where A_f is the area and $\boldsymbol{c}_f$ is the center of face f. The support point always lies inside the convex hull of the support surface. This means that the model is guaranteed to stand if the center of gravity is directly above the support point. Consequently, we choose the horizontal squared distance between the center of gravity $\boldsymbol{c} = [c_x, c_y, c_z]^T$ and $\boldsymbol{s} = [s_x, s_y, s_z]^T$ as the objective function:

$$f(\boldsymbol{\rho}, \boldsymbol{p}) = (c_x(\boldsymbol{\rho}, \boldsymbol{p}) - s_x)^2 + (c_z(\boldsymbol{\rho}, \boldsymbol{p}) - s_z)^2. \tag{2}$$

Here, the center of gravity is

$$\boldsymbol{c}(\boldsymbol{\rho}, \boldsymbol{p}) = \frac{1}{\sum\limits_e \rho_e \cdot V_e(\boldsymbol{p})} \sum_e \rho_e \cdot V_e(\boldsymbol{p}) \cdot \boldsymbol{c}_e(\boldsymbol{p}) \tag{3}$$

where V_e is the volume and $\boldsymbol{c}_e$ is the barycenter of element e. The center of gravity is both a function of the densities of the tetrahedral elements $\boldsymbol{\rho} = [\dots, \rho_e, \dots]^T$ and the positions of the nodes on the surfaces between two materials $\boldsymbol{p} = [\dots, \boldsymbol{p}_n, \dots]^T$. We will change both the materials and surface positions to minimize the objective function.

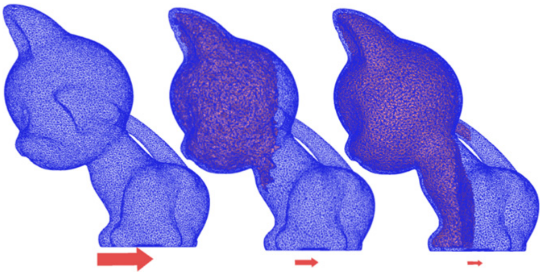


Fig. 3. Iterations 0, 1 and 10 of a hollowing process. A cavity of air is created in a solid plastic model by relabeling tetrahedra end deforming the shape of the internal cavity based on solving an optimization problem and using the DSC method. Red arrows point from the center of gravity to the support point.

3.1. Hollowing

Hollowing creates cavities filled with other materials than the material of the outer shell (Fig. 3). Furthermore, it does not change the shape of the model. This is ensured by applying a global constraint which enforces a minimum thickness of the outer shell of T^*:

$$g(\boldsymbol{\rho}, \boldsymbol{p}) = \frac{1}{N} \sum_{n=1}^{N} \max\left(T^* - t_n(\boldsymbol{\rho}, \boldsymbol{p}), 0\right)^2 \leq 0. \tag{4}$$

Here, $t_n(\boldsymbol{\rho}, \boldsymbol{p})$ is the thickness of the shell at surface node n. Note that $g(\boldsymbol{\rho}, \boldsymbol{p})$ is one time differentiable w.r.t. each variable in $\boldsymbol{p}$. Hollowing consists of a material optimization step followed by a shape optimization step. These two steps are then iterated until changes are small.

The material optimization step determines the optimal material densities $\boldsymbol{\rho}^* = [\dots, \rho_e^*, \dots]^T$ for all tetrahedral elements such that the objective function in Eq. (2) is minimized subject to the constraint in Eq. (4). To determine $\boldsymbol{\rho}^*$, we estimate the change in the objective function when the material in element e is changed from the current material with density ρ_e to material m with density ρ^m. To efficiently compute these discrete derivatives of the objective function, we will use the approximation

$$f(\boldsymbol{\rho}_e^m, \boldsymbol{p}) - f(\boldsymbol{\rho}, \boldsymbol{p}) \approx \frac{\partial f(\boldsymbol{\rho}, \boldsymbol{p})}{\partial \rho_e} \cdot (\rho^m - \rho_e) \tag{5}$$

where $\boldsymbol{\rho}_e^m$ equals $\boldsymbol{\rho}$ with ρ_e replaced by ρ^m. Then, among the N_m predefined material densities $\rho^m \in \left\{\rho^1, \dots, \rho^{N_m}\right\}$ which satisfies

$$g(\boldsymbol{\rho}_e^m, \boldsymbol{p}) - g(\boldsymbol{\rho}, \boldsymbol{p}) \leq 0 \tag{6}$$

we find

$$\rho_e^* = \arg\min_{\rho^m} \frac{\partial f(\boldsymbol{\rho}, \boldsymbol{p})}{\partial \rho_e} \cdot (\rho^m - \rho_e). \tag{7}$$

When $\boldsymbol{\rho}^*$ has been determined, the material in element e is changed to the material with density ρ_e^* by relabeling the element using the DSC method. Note that the positions of the surface nodes $\boldsymbol{p}$ are kept fixed during this step.

The shape optimization step first determines improved positions $\boldsymbol{p}^* = [\dots, \boldsymbol{p}_n^*, \dots]^T$ for all internal surface nodes such that the objective function is minimized subject to the shell thickness constraint in Eq. (4). Notice that the outer surface nodes are not a part of the optimization variables $\boldsymbol{p}$ in this step. Consequently, only the shape of the internal cavities is deformed. To determine $\boldsymbol{p}^*$, we first realize that changing the position of a surface node in the tangent directions will not change the shape much. Consequently, to

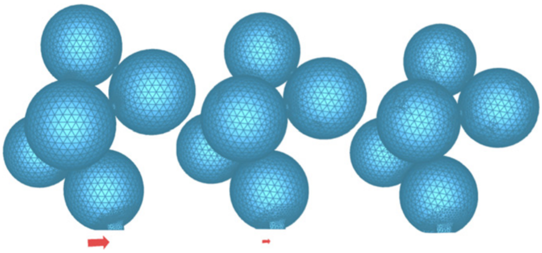


Fig. 4. Iterations 0, 20 and 40 of rotating an unbalanced model such that the center of gravity approaches the support point (illustrated by red arrows). The rotation angles are found by solving an optimization problem before all surface nodes except the nodes at the support surface are rotated by the DSC method.

reduce the number of optimization variables, we associate one design variable x_n with each surface node n. This variable is related to the position by

$$\boldsymbol{p}(\boldsymbol{x}) = [\ldots, \boldsymbol{p}_n + x_n \cdot \boldsymbol{n}_n, \ldots]^T \tag{8}$$

where $\boldsymbol{n}_n$ is the normal of node n and $\boldsymbol{x} = [\ldots, x_n, \ldots]^T$. To determine $\boldsymbol{x}^* = [\ldots, x_n^*, \ldots]^T$ and thereby $\boldsymbol{p}^* = \boldsymbol{p}(\boldsymbol{x}^*)$, we solve the smooth optimization problem

$$\begin{aligned} \boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} &: f(\boldsymbol{\rho}, \boldsymbol{p}(\boldsymbol{x})) \\ \text{subject to} &: g(\boldsymbol{\rho}, \boldsymbol{p}(\boldsymbol{x})) \leq 0 \\ &: \boldsymbol{x} \in [\boldsymbol{x}^{\min}, \boldsymbol{x}^{\max}] \end{aligned} \tag{9}$$

where the material densities $\boldsymbol{\rho}$ are fixed. Furthermore, the design variables are constrained such that the tetrahedra will not degenerate when solving the optimization problem. However, as a consequence, the deformations can only be small. This is remedied by repeating this and the material optimization step. We choose the gradient-based optimization algorithm *method of moving asymptotes* (MMA) [5] to solve the optimization problem in Eq. (9). Then, when $\boldsymbol{p}^* = \boldsymbol{p}(\boldsymbol{x}^*)$ is computed, each surface node n is moved from the current position $\boldsymbol{p}_n$ to $\boldsymbol{p}_n^*$ by the DSC method, i.e. without ruining the simplicial complex.

3.2. Rotation

In addition to hollowing, we will apply a rotation of the model except at the support surface such that the objective function is minimized (Fig. 4). This will introduce an undesired deformation at the base of the model. However, this problem is greatly mitigated by the fact that we rotate around the support point. Consequently, the surface movement is smallest near the support. Furthermore, an initial hollowing often means that only a very small angle of rotation is needed.

Rotation consists of repeating a rotation optimization step. This step determines an improved position $\boldsymbol{p}^* = [\ldots, \boldsymbol{p}_n^*, \ldots]^T$ for all surface nodes except those that are touching the ground. We will make a change of variable such that

$$\boldsymbol{p}(\phi, \theta) = [\ldots, \boldsymbol{s} + \boldsymbol{R}(\phi, \theta) \cdot (\boldsymbol{p}_n - \boldsymbol{s}), \ldots]^T. \tag{10}$$

Here, ϕ and θ are the rotation angles around the x- and z-axis respectively and $\boldsymbol{R}(\phi, \theta)$ is a 3×3 rotation matrix. The improved positions $\boldsymbol{p}^* = \boldsymbol{p}(\phi^*, \theta^*)$ are then found by solving the smooth optimization problem

$$\begin{aligned} \phi^*, \theta^* = \arg\min_{\phi,\theta} &: f(\boldsymbol{\rho}, \boldsymbol{p}(\phi, \theta)) \\ \text{subject to} &: \phi \in [\phi^{\min}, \phi^{\max}] \\ &: \theta \in [\theta^{\min}, \theta^{\max}]. \end{aligned} \tag{11}$$

Again, the densities $\boldsymbol{\rho}$ are fixed and the optimization variables ϕ and θ are constrained such that tetrahedra does not become degenerate when solving Eq. (11). The optimization problem is solved using MMA [5] before moving each surface node n from the current position $\boldsymbol{p}_n$ to $\boldsymbol{p}_n^*$ by the DSC method. Surface nodes which move below the ground plane are clamped to the ground plane while any discretization problems are handled by the DSC method.

4. Experiments

In this section we conduct a series of experiments using different printing technologies. We focused on three models: a deformed version of the standard kitten model, a variant of the well-known Armadillo model where we attached a larger head, and a model consisting of five connected spheres. In each case the model was sized to be initially 60 mm in the vertical direction. Each model was initially unbalanced, regardless of printing technology, and we use a $T^* = 1$ mm shell thickness in all optimizations.

To test solid/air configurations, we used a Stratasys Connex 260 printer. This printer fills internal cavities with a flexible support material which must be removed using a water-jet system. The support was removed by slicing the models with planar cuts that intersected the cavities, and then glued them back together after cleaning. In addition, we used a Stratasys Mojo printer. Here, the support material was dissolved in a water-based solution. This eliminates the need for assembling; instead, it requires holes in the model for the solution to enter the cavities. These holes are created in the model before printing but after optimization.

The kitten model (Fig. 5) was successfully balanced via internal cavities of air generated with our hollowing strategy. Balancing via rotation was also successful though it introduces large deformations to the base of the model. As expected based on the results of the optimization, the prints are very stable.

The armadillo–dog model (Fig. 6) could not be balanced strictly by creating cavities of air. However, a combination of cavities and rotation was successful. As expected, a rotation-only strategy was also effective but less rotation, and thereby less deformation of the feet is necessary if preceded by hollowing.

The five-spheres model (Fig. 7) was similar in that air cavities were insufficient due to the very small support surface. The combination of cavities and rotation was successful, as was the rotation-only case (Fig. 4).

To test our variable-density optimization, we used a Makerbot Replicator 2 FDM-style 3D printer, with standard Makerbot PLA filament. As previously discussed, use of this printer is complicated by the fact that the model does not have uniform density, and interior cavities may contain support structures. The printing software permits us to specify the infill density and the density of support structures as global values. However, since the Makerbot uses a single print material, these settings are only suggestions to the algorithm that generates the *tool path*. To validate the relationship between the infill density parameter and the resulting density of the infill structure, we performed an experiment in which we printed a 40 mm^3 half-cube (sliced diagonally). By comparing to a hollow print, we could observe the following relationships between infill parameter and physical infill density 5%/7%, 10%/12%, 20%/22% for three test cases. Based on these results we assume that the infill parameter is an adequate estimate of infill density and use it in the optimization. For the support structures, the actual toolpath is highly context-dependent. The filament pattern depends not only on the cavity shape, but also alignment with an underlying grid. Furthermore, to save material and make the support easier to break away, the software randomly skips layers in the support structure. The result is extremely complicated to analyze without actually generating the infill toolpath, which

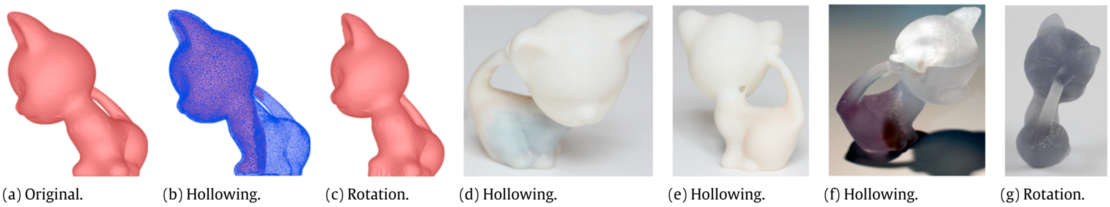

(a) Original. (b) Hollowing. (c) Rotation. (d) Hollowing. (e) Hollowing. (f) Hollowing. (g) Rotation.

Fig. 5. The large head of the kitten model (a) causes it to topple. We make it stand by inserting cavities of air using hollowing (b) and print it using a Stratasys Mojo printer (d, e) and a Stratasys Connex printer (f). We also apply rotation to make it stand (c) and print it using a Stratasys Connex printer (g).

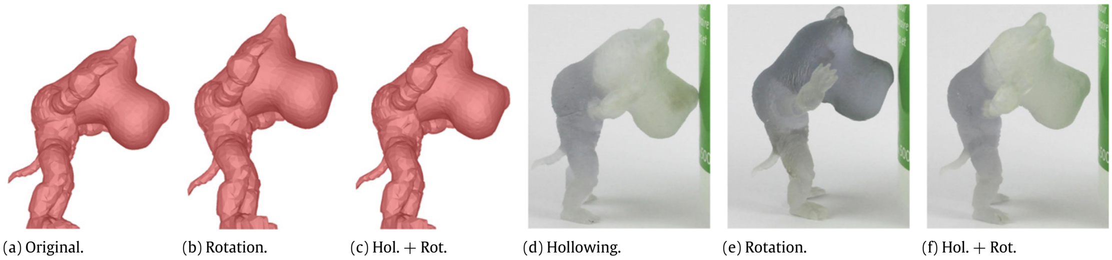

(a) Original. (b) Rotation. (c) Hol. + Rot. (d) Hollowing. (e) Rotation. (f) Hol. + Rot.

Fig. 6. The armadillo–dog model (a) does not balance when printed solid on a Stratasys Connex printer or when printed with cavities of air generated by hollowing (d). However, by rotating (b, e) or by creating cavities of air and rotating (c, f) the model stands.

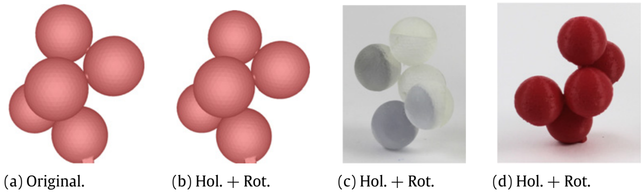

(a) Original. (b) Hol. + Rot. (c) Hol. + Rot. (d) Hol. + Rot.

Fig. 7. Hollowing is not enough to make the statue of spheres model (a) stand unless it is combined with rotation (b). The optimized model is printed on a Stratasys Connex printer (c). Our method also manages to balance a Makerbot print (d) of this model by hollowing (with 25% infill and 0% cavity density) and rotation.

was intractable to include in our optimization. Hence, we assume that the specified support density parameter is a worst-case value and use it directly in the optimization as the density of cavities of air. We could also disable support structures entirely; however, this introduces a risk of print failure.

We performed extensive experimentation on *the kitten model* (Fig. 8), varying both the infill and cavity density. At zero cavity density, we could successfully balance the kitten even at 10% infill. At 5% cavity density, infill had to be increased to 30%, and even at that point the actual print was on the edge of being unbalanced. At 10% cavity density, we had to increase infill to 50% before we could achieve balance in the printed result.

The five-spheres model (Fig. 7) did not allow for balancing with non-zero cavity density using hollowing. However if we included rotation, we could easily achieve balance with 10% cavity and 25% infill density and with zero cavity and 25% infill density.

5. Conclusion

We have presented and illustrated an automatic method for deforming a 3D model to ensure it will stand when manufactured using a 3D printer. Our method embeds the original model directly in an adaptive tetrahedral mesh. Consequently, the shape of the model is not deformed at initialization and neither when the method creates cavities of lighter and heavier materials inside the model using hollowing. Furthermore, the ability to take multiple materials into account in the hollowing process makes it possible to simulate the infill and support structures of the popular FDM type of 3D printers. Unfortunately, hollowing is often not sufficient to make a model stand and for FDM type printers the issue is generally worse. Therefore, deformation of the outer surface of the model is often essential as previously reported [1]. Our approach has been to rotate around the base of the model. Fortunately, the needed rotations are often small, especially when hollowing has already been applied. Therefore, the rotation strategy has proven to be quite effective, despite its simplicity. Furthermore, the objective function is convex, the method iteratively approaches an optimum and the rotation step is unconstrained. Hence, we claim that the optimization will always converge and, as a consequence, the method is able to balance any 3D model. This claim is supported by the presented test cases; however, it has not been proven.

Should any kind of change to the exterior surface be undesired, the future may bring us some more options. In particular, we hope it will be possible to print in materials with significantly different densities. Three examples are presented in Fig. 9 where cavities of glass, copper and gold respectively are created inside a plastic model in addition to cavities of air.

A limitation of our method is that we do not model the support structures inside a model intended for FDM print accurately. One extension that could allow for fully void cavities inside FDM prints would be to control the shape of the upper portions of the cavities. If all surfaces on the cavity boundary satisfied the overhang draft angle constraint, then no support would be needed to print them. Alternatively, cavities with flat tops could be printed without

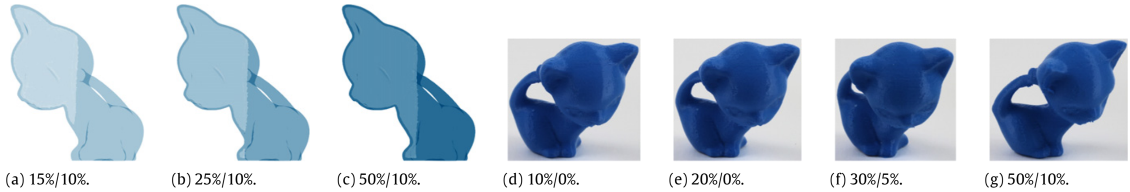

(a) 15%/10%. (b) 25%/10%. (c) 50%/10%. (d) 10%/0%. (e) 20%/0%. (f) 30%/5%. (g) 50%/10%.

Fig. 8. The infill (shown dark)/cavity (shown light) of a Makerbot print are simulated in the hollowing optimization by a percentage of solid material (a–c). The optimized results using the parameters given in the captions have been printed on a Makerbot (d–g). Although the 25% infill version (b) is borderline balanced in the optimization, in an actual print only the 50% infill (c, g) would stand.

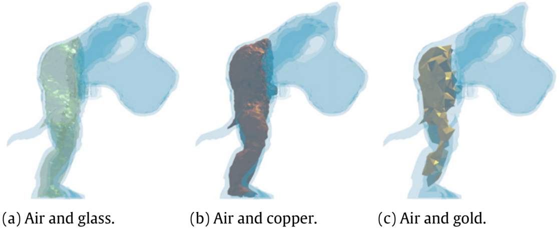

(a) Air and glass. (b) Air and copper. (c) Air and gold.

Fig. 9. Combining air cavities with cavities of a denser material than plastic will eliminate the need to change the shape at all. Here, we show three examples using glass, copper and gold. Using glass makes the model borderline balanced while using copper or gold ensures a completely balanced model. Furthermore, only a small cavity of gold is needed to make the model balance.

support using the *bridging* capabilities of FDM printing, in which the print head moves rapidly to lay down a horizontal scaffold on which to print additional layers.

Acknowledgment

The authors appreciate the support from the Villum Foundation through the grant: "NextTop".

References

[1] Prévost Romain, Whiting Emily, Lefebvre Sylvain, Sorkine-Hornung Olga. Make it stand: balancing shapes for 3D fabrication. ACM Trans Graph 2013;32(4): 81:1–81:10. (proceedings of ACM SIGGRAPH).

[2] Si Hang. TetGen: a quality tetrahedral mesh generator and a 3D delaunay triangulator, 2013.

[3] Misztal Marek Krzysztof, Bærentzen Jakob Andreas. Topology adaptive interface tracking using the deformable simplicial complex. ACM Trans Graph 2012; 31(3):24.

[4] Christiansen Asger Nyman, Nobel-Jørgensen Morten, Aage Niels, Sigmund Ole, Bærentzen Jakob Andreas. Topology optimization using an explicit interface representation. Struct Multidiscip Optim 2014;49(3):387–99.

[5] Svanberg Krister. The method of moving asymptotes—a new method for structural optimization. Internat J Numer Methods Engrg 1987;24(2):359–73.

Bibliography

Aage, N., Andreassen, E., and Lazarov, B. S. (2014). Topology optimization using petsc: An easy-to-use, fully parallel, open source topology optimization framework. *Structural and Multidisciplinary Optimization*, pages 1–8.

Aage, N., Nobel-Jørgensen, M., Andreasen, C. S., and Sigmund, O. (2013). Interactive topology optimization on hand-held devices. *Structural and Multidisciplinary Optimization*, 47(1):1–6.

Aage, N., Sigmund, O., and Breinbjerg, O. (2011). *Topology optimization of radio frequency and microwave structures.* PhD thesis, Technical University of Denmark (DTU).

Allaire, G., Dapogny, C., Delgado, G., and Michailidis, G. (2013a). Multi-phase structural optimization via a level set method. Technical report, Sorbonne University. 33 pages.

Allaire, G., Dapogny, C., and Frey, P. (2011). Topology and geometry optimization of elastic structures by exact deformation of simplicial mesh. *Comptes Rendus Mathematique*, 349(17-18):999–1003.

Allaire, G., Dapogny, C., and Frey, P. (2013b). A mesh evolution algorithm based on the level set method for geometry and topology optimization. *Structural and Multidisciplinary Optimization*, 48(4):711–715.

Allaire, G., Jouve, F., and Michailidis, G. (2014). Thickness control in structural optimization via a level set method. (Submitted).

Allaire, G., Jouve, F., and Toader, A.-M. (2004). Structural optimization using sensitivity analysis and a level-set method. *Journal of Computational Physics*, 194(1):363–393.

Amir, O., Aage, N., and Lazarov, B. S. (2014). On multigrid-cg for efficient topology optimization. *Structural and Multidisciplinary Optimization*, 49(5):815–829.

Arnout, S., Firl, M., and Bletzinger, K.-U. (2012). Parameter free shape and thickness optimisation considering stress response. *Structural and Multidisciplinary Optimization*, 45(6):801–814.

Bærentzen, J. A., Gravesen, J., Anton, F., and Aanæs, H. (2012). *Guide to Computational Geometry Processing: Foundations, Algorithms, and Methods.* Springer.

Bærentzen, J. A., Revall Frisvad, J., and Aanæs, H. (2013). GEometry and Linear algebra (GEL) library.

Bendsøe, M. P. (1989). Optimal shape design as a material distribution problem. *Structural Optimization*, 1(4):193–202.

Bendsøe, M. P. and Kikuchi, N. (1988). Generating optimal topologies in structural design using a homogenization method. *Computer Methods in Applied Mechanics and Engineering*, 71(2):197 – 224.

Bendsøe, M. P. and Sigmund, O. (2003). *Topology Optimization - Theory, Methods, and Applications.* Springer Verlag, Berlin, second edition.

Bucur, D. and Buttazzo, G. (2006). *Variational Methods in Shape Optimization Problems.* Progress in Nonlinear Differential Equations and Their Applications. Birkhäuser.

Céa, J., Garreau, S., Guillaume, P., and Masmoudi, M. (2000). The shape and topological optimizations connection. *Computer Methods in Applied Mechanics and Engineering*, 188(4):713–726.

Chen, J., Shapiro, V., Suresh, K., and Tsukanov, I. (2007). Shape optimization with topological changes and parametric control. *International Journal for Numerical Methods in Engineering*, 71(3):313–346.

Cheng, S., Dey, T., and Shewchuk, J. (2012). *Delaunay Mesh Generation.* Chapman & Hall/CRD Computer and Information Science. CRC PressINC.

Choi, K. and Kim, N. (2010). *Structural Sensitivity Analysis and Optimization 1: Linear Systems.* Number vb. 1 in Mechanical Engineering Series. Springer.

Christensen, P. W. and Klarbring, A. (2008). *An Introduction to Structural Optimization.* Solid mechanics and its applications. Springer.

Christiansen, A. N. and Bærentzen, J. A. (2013). Generic graph grammar: A simple grammar for generic procedural modelling. In *Proceedings of the 28th Spring Conference on Computer Graphics*, SCCG '12, pages 85–92, New York, NY, USA. ACM.

Christiansen, A. N., Bærentzen, J. A., Nobel-Jørgensen, M., Aage, N., and Sigmund, O. (2014a). Combined shape and topology optimization of 3D structures. *Computers & Graphics*, 46(2015):25 – 35. Shape Modeling International 2014.

Christiansen, A. N., Carstensen, J. M., Møller, F., and Nielsen, A. A. (2012). Monitoring the change in colour of meat: A comparison of traditional and kernel-based orthogonal transformations. *Journal of Spectral Imaging*, 3(1).

Christiansen, A. N., Nobel-Jørgensen, M., Aage, N., Sigmund, O., and Bærentzen, J. A. (2014b). Topology optimization using an explicit interface representation. *Structural and Multidisciplinary Optimization*, 49(3):387–399.

Christiansen, A. N., Nobel-Jørgensen, M., Bærentzen, J. A., Aage, N., and Sigmund, O. (2013). Topology optimization using an explicit interface representation. Presented at the 10th World Congress on Structural and Multidisciplinary Optimization (WCSMO-10).

Christiansen, A. N., Schmidt, R., and Bærentzen, J. A. (2014c). Automatic balancing of 3D models. *Computer-Aided Design*, 58(2015):236 – 241. Solid and Physical Modeling 2014.

Christiansen, A. N., Tortorelli, D. A., Aage, N., and Sigmund, O. (2014d). Combined shape and topology optimization for minimization of maximal von Mises stress. (Submitted).

Christiansen, A. N., Tortorelli, D. A., Aage, N., and Sigmund, O. (2014e). Combined shape and topology optimization for minimization of von Mises stress. Presented at the 4th International Conference on Engineering Optimization (EngOpt 2014).

Cook, R. D., Malkus, D. S., Plesha, M. E., and Witt, R. J. (2007). *Concepts and Applications of Finite Element Analysis*. John Wiley & Sons.

Dahl, V. A., Christiansen, A. N., and Bærentzen, J. A. (2014). Multiphase image segmentation using the deformable simplicial complex method. In *Proceedings of the 22nd International Conference on Pattern Recognition*.

Davis, T. A., Hager, W. W., and Duff, I. S. (2013). SuiteSparse.

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

Ding, Y. (1986). Shape optimization of structures: a literature survey. *Computers & Structures*, 24(6):985–1004.

Duysinx, P. and Sigmund, O. (1998). New developments in handling optimal stress constraints in optimal material distributions. In *Conference Proceedings of 7th Symposium in Multidisciplinary Analysis and Optimization, AIAA/USAF/NASA/ISSMO*, volume 3, pages 1501–1509, St. Louis Missouri, USA.

Enright, D., Fedkiw, R., Ferziger, J., and Mitchell, I. (2002). A hybrid particle level set method for improved interface capturing. *Journal of Computational Physics*, 183(1):83 – 116.

Eschenauer, H. A., Kobelev, V. V., and Schumacher, A. (1994). Bubble method for topology and shape optimization of structures. *Structural and Multidisciplinary Optimization*, 8:42–51.

Feijóo, R. A., Novotny, A. A., Taroco, E., and Padra, C. (2003). The topological derivative for the poisson's problem. *Mathematical Models and Methods in Applied Sciences*, 13(12):1825–1844.

Field, D. A. (1988). Laplacian smoothing and delaunay triangulations. *Communications in Applied Numerical Methods*, 4(6):709–712.

Garreau, S., Guillaume, P., and Masmoudi, M. (2000). The topological asymptotic for pde systems: The elasticity case. *SIAM J. Control Optim.*, 39(6):1756–1778.

Guest, J. K., Prévost, J. H., and Belytschko, T. (2004). Achieving minimum length scale in topology optimization using nodal design variables and projection functions. *International Journal for Numerical Methods in Engineering*, 61(2):238–254.

Ha, S.-H. and Cho, S. (2008). Level set based topological shape optimization of geometrically nonlinear structures using unstructured mesh. *Computers & Structures*, 86(13-14):1447–1455.

Hashin, Z. and Shtrikman, S. (1963). A variational approach to the theory of the elastic behaviour of multiphase materials. *Journal of the Mechanics and Physics of Solids*, 11(2):127 – 140.

Kim, D.-H., Lee, S. B., Kwank, B. M., Kim, H.-G., and Lowther, D. (2008). Smooth boundary topology optimization for electrostatic problems through the combination of shape and topological design sensitivities. *IEEE Transactions on Magnetics*, 44(6):1002–1005.

Klincsek, G. T. (1980). Minimal triangulations of polygonal domains. *Annals of Discrete Mathematics*, 9:121–123.

Le, C., Bruns, T., and Tortorelli, D. (2011). A gradient-based, parameter-free approach to shape optimization. *Computer Methods in Applied Mechanics and Engineering*, 200(9-12):985–996.

Le, C., Norato, J., Bruns, T., Ha, C., and Tortorelli, D. (2010). Stress-based topology optimization for continua. *Structural and Multidisciplinary Optimization*, 41(4):605–620.

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

Misztal, M. K. and Bærentzen, J. A. (2012). Topology adaptive interface tracking using the deformable simplicial complex. *ACM Transactions on Graphics*, 31(3):No. 24.

Misztal, M. K., Bærentzen, J. A., and Anton, F. (2010a). *Deformable Simplicial Complexes*. PhD thesis, Technical University of Denmark, Department of Informatics and Mathematical Modeling.

Misztal, M. K., Bridson, R., Erleben, K., Bærentzen, J. A., and Anton, F. (2010b). Optimization-based fluid simulation on unstructured meshes. *7th Workshop on Virtual Reality Interaction and Physical Simulation VRIPHYS*, page 10.

Misztal, M. K., Erleben, K., Bargteil, A., Fursund, J., Christensen, B. B., Bærentzen, J. A., and Bridson, R. (2012). Multiphase flow of immiscible fluids on unstructured moving meshes. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '12, pages 97–106, Aire-la-Ville, Switzerland. Eurographics Association.

Mohammadi, B. and Pironneau, O. (2001). *Applied shape optimization for fluids*, volume 28. Oxford University Press, Oxford.

Nobel-Jørgensen, M., Aage, N., Christiansen, A. N., Igarashi, T., Bærentzen, J. A., and Sigmund, O. (2014). 3D interactive topology optimization on handheld devices. (Submitted).

Nobel-Jørgensen, M., Christiansen, A. N., Bærentzen, J. A., Aage, N., and Sigmund, O. (2013). Improving topology optimization using games. Presented at the 10th World Congress on Structural and Multidisciplinary Optimization (WCSMO-10).

Oñate, E., Perazzo, F., and Miquel, J. (1999). Advances in the stabilized finite point meted for structural mechanics. In *Publication No. 164, CIMNE*, Barcelona, Spain.

Osher, S. J. and Fedkiw, R. P. (2002). *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 1 edition.

Parthasarathy, V. N., Graichen, C. M., and Hathaway, A. F. (1994). A comparison of tetrahedron quality measures. *Finite Elem. Anal. Des.*, 15(3):255–261.

Pironneau, O. (1982). Optimal shape design for elliptic systems. In Drenick, R. and Kozin, F., editors, *System Modeling and Optimization*, volume 38 of *Lecture Notes in Control and Information Sciences*, pages 42–66. Springer Berlin Heidelberg.

Shewchuk, J. R. (2002). Two discrete optimization algorithms for the topological improvement of tetrahedral meshes. Technical report, University of California at Berkeley.

Si, H. (2013). TetGen: A quality tetrahedral mesh generator and a 3d delaunay triangulator.

Sigmund, O. and Guest, J. K. (2014). On the relation between material bounds, topological derivatives and structural topology optimization methods. (In preparation).

Sigmund, O. and Maute, K. (2013). Topology optimization approaches. *Structural and Multidisciplinary Optimization*, 48(6):1031–1055.

Sokolowski, J. and Zochowski, A. (1999). On the topological derivative in shape optimization. *SIAM Journal on Control and Optimization*, 37(4):1251–1272.

Svanberg, K. (1987). The method of moving asymptotes – a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24(2):359–373.

Trottenberg, U., Oosterlee, C. W., and Schuller, A. (2000). *Multigrid*. Academic press.

Vassilevski, P. S. (2008). *Multilevel Block Factorization Preconditioners, Matrix-based Analysis and Algorithms for Solving Finite Element Equations.* Springer, New York.

Wang, M., Wang, X., and Guo, D. (2003). A level set method for structural topology optimization. *Computer Methods in Applied Mechanics and Engineering*, 192(1):227–246.

Wojtan, C., Thürey, N., Gross, M., and Turk, G. (2009). Deforming meshes that split and merge. In *ACM SIGGRAPH 2009 papers*, SIGGRAPH '09, pages 76:1–76:10, New York, NY, USA. ACM.

Xia, Q., Shi, T., Liu, S., and Wang, M. Y. (2012). A level set solution to the stress-based structural shape and topology optimization. *Computers & Structures*, 90 - 91(0):55–64.

Yamasaki, S., Nomura, T., Kawamoto, A., Sato, K., and Nishiwaki, S. (2011). A level set-based topology optimization method targeting metallic waveguide design problems. *International Journal for Numerical Methods in Engineering*, 87(9):844–868.

Zhang, W.-H., Beckers, P., and Fleury, C. (1995). A unified parametric design approach to structural shape optimization. *International Journal for Numerical Methods in Engineering*, 38(13):2283–2292.