Shape Optimization Using the Cut Finite Element Method*

Erik Burman† Daniel Elfverson‡ Peter Hansbo§
Mats G. Larson¶ Karl Larsson‖

August 25, 2017

Abstract

We present a cut finite element method for shape optimization in the case of linear elasticity. The elastic domain is defined by a level-set function, and the evolution of the domain is obtained by moving the level-set along a velocity field using a transport equation. The velocity field is the largest decreasing direction of the shape derivative that satisfies a certain regularity requirement and the computation of the shape derivative is based on a volume formulation. Using the cut finite element method no re–meshing is required when updating the domain and we may also use higher order finite element approximations. To obtain a stable method, stabilization terms are added in the vicinity of the cut elements at the boundary, which provides control of the variation of the solution in the vicinity of the boundary. We implement and illustrate the performance of the method in the two–dimensional case, considering both triangular and quadrilateral meshes as well as finite element spaces of different order.

*This research was supported in part by the Swedish Foundation for Strategic Research Grant No. AM13-0029, the Swedish Research Council Grants Nos. 2011-4992, 2013-4708, the Swedish Research Programme Essence, and EPSRC, UK, Grant Nr. EP/J002313/1.

†Department of Mathematics, University College London, Gower Street, London WC1E 6BT, UK, e.burman@ucl.ac.uk

‡Department of Mathematics and Mathematical Statistics, Umeå University, SE–901 87 Umeå, Sweden, daniel.elfverson@umu.se

§Department of Mechanical Engineering, Jönköping University, SE-551 11 Jönköping, Sweden, peter.hansbo@ju.se

¶Department of Mathematics and Mathematical Statistics, Umeå University, SE–901 87 Umeå, Sweden, mats.larson@umu.se

‖Department of Mathematics and Mathematical Statistics, Umeå University, SE–901 87 Umeå, Sweden, karl.larsson@umu.se

Contents

1 Introduction **3**

2 Model Problem **4**
- 2.1 Linear Elasticity . . . 4
- 2.2 Minimization Problem . . . 5

3 Cut Finite Element Method **6**
- 3.1 The Mesh and Finite Element Spaces . . . 6
- 3.2 The Method . . . 7
- 3.3 Geometry Description . . . 9

4 Shape Calculus **11**
- 4.1 Definition of the Shape Derivative . . . 11
- 4.2 Shape Derivative . . . 12
- 4.3 Finite Element Approximation of the Shape Derivative . . . 14

5 Domain Evolution **14**
- 5.1 Shape Evolution . . . 14
- 5.2 Optimization Algorithm . . . 15

6 Numerical Results **16**
- 6.1 Model Problems . . . 16
- 6.2 Implementation Aspects . . . 16
- 6.3 Numerical Experiments . . . 16

7 Conclusions **27**

1 Introduction

Optimization of elastic structures is an important and active research field of significant interest in engineering. There are two common approaches to represent the domain which we seek to optimize: (i) *A density function.* This approach, common in topology optimization [4, 8], is very general and computationally convenient, but the boundary representation is not sharp and thus typically fine grids and low order approximation spaces are employed. (ii) *An implicit or explicit representation of the boundary.* This approach is common in shape optimization [18] where the boundary is typically described by a level-set function or a parametrization, but topological changes can also be handled for instance using an implicit level-set representation of the boundary. Given the boundary representation we need to generate a discretization of the domain when it is updated. This can be done using a standard meshing approach based on mesh motion and/or re-meshing or alternatively using a fictitious domain method, see [3, 16, 20] for different approaches.

In this contribution we focus on the fictitious domain approach using the recently developed cut finite element method CutFEM [5, 7], extending our previous work on the Bernoulli free boundary value problem [6] to linear elasticity. The key components in CutFEM are: (i) Use of a fixed background mesh and a sharp boundary representation that is allowed to cut through the background mesh in arbitrary fashion. (ii) Weak enforcement of the boundary conditions. (iii) Stabilization of the cut elements in the vicinity of the boundary using a consistent stabilization term which leads to optimal order accuracy and conditioning of the resulting algebraic system of equations. CutFEM also allows higher order finite element spaces and rests on a solid theoretical foundation including stability bounds, optimal order a priori error bounds, and optimal order bounds for the condition numbers of the stiffness and mass matrices, see [5] and the references therein.

In order to support large changes in the shape and topology of the domain during the optimization process we employ a level-set representation of the boundary. The evolution of the domain is obtained by moving the level-set along a velocity field using a Hamilton-Jacobi transport equation, see [2, 3]. The velocity field is the largest decreasing direction of the shape derivative that satisfies a certain regularity requirement together with a boundary conditions on the boundary of the design volume. The computation of the shape derivative is based on a volume formulation, see [12, 13] for similar approaches. In this context CutFEM provides an accurate and stable approximation of the linear elasticity equations which completely avoids the use of standard meshing procedures when updating the domain. In this contribution we focus on standard Lagrange elements, but a wide range of elements may be used in CutFEM, including discontinuous elements, isogeometric elements with higher order regularity, and mixed elements.

When using a higher order finite element space in shape optimization we use a finer representation of the domain than the computational mesh, i.e., the level-set representing the geometry is defined on a finer mesh. For computational convenience and efficiency we use a piecewise linear geometry description on the finer grid. This approach of course leads to loss of optimal order but the main purpose of using the finer grid is to allow the

domain to move more freely on the refined grid despite using larger higher order elements for the approximation of the solution field. If necessary, once a steady design has been found, a more accurate final computation can be done which uses a higher order geometry description. Our approach leads to a convenient and efficient implementation since the solution to the elasticity equations and the level-set function are represented using the fixed background mesh and a uniform refinement thereof.

An outline of the paper is as follows: in Section 2 we formulate the equations of linear elasticity and the optimization problem, in Section 3 we formulate the CutFEM, in Section 4 we recall the necessary results from shape calculus, in Section 5 we formulate the transport equation for the level-set and the optimization algorithm, and finally in Section 6 we present numerical results.

2 Model Problem

2.1 Linear Elasticity

Let $\Omega \in \mathbb{R}^d$, for $d = 2, 3$, be a bounded domain with boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$ such that $\Gamma_D \cap \Gamma_N = \emptyset$, and let n be the exterior unit normal to $\partial\Omega$. Assuming a linear elastic isotropic material the constitutive relationship between the symmetric stress tensor σ and the strain tensor ϵ is given by Hooke's law

$$\sigma = 2\mu\epsilon + \lambda \operatorname{tr}(\epsilon) I \tag{2.1}$$

where μ and λ are the Lamé parameters and I is the $d \times d$ identity matrix. Also, assuming small strains we may use the linear strain tensor $\epsilon(u) = (\nabla u + \nabla u^T)/2$, where u is the displacement field, as a strain measure. In this expression for $\epsilon(u)$ we take the gradient of vector fields which we define as $\nabla u = u \otimes \nabla$, i.e. the Jacobian of u. Under these assumptions the governing equations for the stress σ and displacements u of an elastic body in equilibrium are

$$-\operatorname{Div}\sigma = f \qquad \text{in } \Omega \tag{2.2}$$

$$\sigma = 2\mu\epsilon(u) + \lambda \operatorname{tr}(\epsilon(u)) I \qquad \text{in } \Omega \tag{2.3}$$

$$u = 0 \qquad \text{on } \Gamma_D \tag{2.4}$$

$$\sigma \cdot n = g \qquad \text{on } \Gamma_N \tag{2.5}$$

where $\operatorname{Div}\sigma$ is the row-wise divergence on σ and f and g are given body and surface force densities, respectively.

In shape optimization we seek to minimize some objective functional $J(\Omega; u(\Omega))$ with respect to the domain Ω where $u(\Omega)$ is the displacement field obtained by solving (2.2)–(2.5). For the present work we choose to minimize the compliance, i.e. the internal energy of the elastic body, and for clarity of presentation we also choose $f = 0$.

Weak Form. The weak form of the problem (2.2)–(2.5) reads: find $u \in V(\Omega) = \{u \in [H^1(\Omega)]^d : u|_{\Gamma_D} = 0\}$ such that

$$a(\Omega; u, q) = L(q) \qquad \forall q \in V(\Omega) \tag{2.6}$$

with bilinear form $a(\Omega; \cdot, \cdot)$ and linear functional $L(\cdot)$ given by

$$a(\Omega; v, q) = 2\mu(\epsilon(v), \epsilon(q))_{L^2(\Omega)} + \lambda(\nabla \cdot v, \nabla \cdot q)_{L^2(\Omega)} \tag{2.7}$$

$$L(q) = (g, q)_{L^2(\Gamma_N)} \tag{2.8}$$

where $(\cdot, \cdot)_{L^2(\Omega)}$ denotes the usual $L^2(\Omega)$ inner product. Specifically, for tensors $A, B : \Omega \to \mathbb{R}^{d \times d}$ the $L^2(\Omega)$ inner product is $(A, B)_{L^2(\Omega)} = \int_\Omega A : B$ where the contraction operator ":" is defined $A : B = \sum_{i,j=1}^d A_{ij} B_{ij}$ and we recall that $\text{tr}(\epsilon(v)) = I : \epsilon(v) = \nabla \cdot v$. Note that by choosing $f = 0$ and Γ_N fixed the linear functional L is independent of Ω.

2.2 Minimization Problem

Given some functional $J(\Omega; v)$ and $v \in V(\Omega)$ we define the objective functional

$$J(\Omega) = J(\Omega; u(\Omega)) \tag{2.9}$$

where $u(\Omega) \in V(\Omega)$ is the solution to (2.6). Letting $\mathcal{O}$ denote the set of all admissible domains we formulate the minimization problem: find $\Omega_{\min} \in \mathcal{O}$ such that

$$J(\Omega_{\min}) = \min_{\Omega \in \mathcal{O}} J(\Omega) \tag{2.10}$$

Lagrangian Formulation. The minimization problem can also be expressed as finding an extreme point $(\Omega_{\min}; u(\Omega_{\min}), p(\Omega_{\min}))$ to the Lagrangian

$$\mathcal{L}(\Omega; v, q) = J(\Omega; v) - a(\Omega; v, q) + L(q) \tag{2.11}$$

and we note that $J(\Omega; u(\Omega)) = \mathcal{L}(\Omega; u(\Omega), q)$ for all $q \in V(\Omega)$ and $\Omega \in \mathcal{O}$ by (2.6). For a fixed domain Ω, the saddle point (u, p) fulfilling

$$\mathcal{L}(\Omega; u, p) = \inf_{v \in V} \sup_{q \in V} \mathcal{L}(\Omega; v, q) \tag{2.12}$$

can be determined by the problems: find $u \in V(\Omega)$ and $p \in V(\Omega)$ such that

$$\langle \partial_q \mathcal{L}(\Omega; u, p), \delta q \rangle = 0 \qquad \forall \delta q \in V(\Omega) \tag{2.13}$$

$$\langle \partial_v \mathcal{L}(\Omega; u, p), \delta v \rangle = 0 \qquad \forall \delta v \in V(\Omega) \tag{2.14}$$

where $\langle \partial_q \mathcal{L}(\Omega, u, p), \delta q \rangle$ denotes the partial derivative of $\mathcal{L}$ with respect to q in the direction of δq, cf. [3]. From (2.13) we can derive the primal problem (2.6); thus u is the solution to (2.6), and from (2.14) we deduce that p is the solution to the following dual problem: find $p \in V(\Omega)$ such that

$$a(\Omega; v, p) = \langle \partial_v J(\Omega, u), v \rangle \quad \forall v \in V(\Omega) \tag{2.15}$$

Recalling that $J(\Omega; v) = \mathcal{L}(\Omega; v, q)$ we can thus express the objective functional (2.9) in terms of the Lagrangian as

$$J(\Omega) = \mathcal{L}(\Omega; u(\Omega), p(\Omega)) = \inf_{v \in V(\Omega)} \sup_{q \in V(\Omega)} \mathcal{L}(\Omega; v, q) \tag{2.16}$$

where $u(\Omega) \in V(\Omega)$ and $p(\Omega) \in V(\Omega)$ are the solutions to

$$a(\Omega; u(\Omega), v) = L(v) \qquad \forall v \in V(\Omega) \tag{2.17}$$
$$a(\Omega; v, p(\Omega)) = \langle \partial_v J(\Omega; u(\Omega)), v \rangle \qquad \forall v \in V(\Omega) \tag{2.18}$$

i.e., the solutions to the primal and dual problem, respectively.

Objective Functional. In the present work we choose to minimize the compliance, i.e. the elastic energy, which is given by

$$\frac{1}{2} a(\Omega; u(\Omega), u(\Omega)) \tag{2.19}$$

where $u(\Omega)$ is the solution to (2.6). Since we also want to constrain the volume $|\Omega|$ we form the functional $J(\cdot; \cdot)$ as the sum of the compliance and a penalty term

$$J(\Omega; v) = \frac{1}{2} a(\Omega; v, v) + \kappa |\Omega| \tag{2.20}$$

where κ acts as a material cost which we in the present work keep fixed. For this choice of objective functional the dual problem (2.15) coincides with the primal problem (2.6) since $\langle \partial_v J(\Omega; u), v \rangle = \frac{1}{2}(a(\Omega; v, u) + a(\Omega; u, v))$ and the bilinear form is symmetric, and hence $p = u$.

Remark 2.1. If a fixed amount of material is desired, i.e., $|\Omega| = \gamma |\Omega_0|$ for some fixed $0 < \gamma < 1$, it is possible to determine κ by some adaptive update strategy of the material cost κ.

Remark 2.2. When considering other functionals $J(\Omega; v)$, typically $u \neq p$ and the dual problem (2.15) must be computed.

3 Cut Finite Element Method

We will use a cut finite element method to solve the primal (and dual) equation (2.6). An analysis of this method for linear elasticity is presented in [11].

3.1 The Mesh and Finite Element Spaces

Consider a fixed polygonal domain Ω_0, with $\Omega \subset \Omega_0$ as illustrated in Figure 1a, and let $\mathcal{K}_{h,0}$ and $\mathcal{F}_{h,0}$ denote a subdivision of Ω_0 into a family of quasi-uniform triangles/tetrahedrons

or a uniform quadrilaterals/bricks with mesh parameter $h \in (0, h_0]$, illustrated in Figure 1b, respectively the set of interior faces in $\mathcal{K}_{h,0}$. Let P_k be the space of full polynomials in $\mathbb{R}^d$ up to degree k on triangular/tetrahedral elements and tensor products of polynomials up to degree k on quadrilateral/brick elements. We define the finite element space of continuous piecewise polynomial functions on $\mathcal{K}_{h,0}$ by

$$V_{h,k,0} = \{v \in [H^1(\Omega_0)]^d \cap [C^0(\Omega_0)]^d \, : \, v|_K \in [P_k(K)]^d \, , \, K \in \mathcal{K}_{h,0}\} \tag{3.1}$$

The active part of the mesh is given by all elements in $\mathcal{K}_{h,0}$ which has a non-zero intersection with the domain Ω. We define the active mesh and its interior faces

$$\mathcal{K}_h = \{K \in \mathcal{K}_{h,0} : \overline{K} \cap \Omega \neq \emptyset\} \quad \text{and} \quad \mathcal{F}_h = \{F \in \mathcal{F}_{h,0} : F \cap \Omega \neq \emptyset\} \tag{3.2}$$

the union of all active elements

$$\Omega_h = \cup_{K \in \mathcal{K}_h} K \tag{3.3}$$

illustrated in Figure 1c, and the finite element space on the active mesh

$$V_{h,k}(\Omega_h) = V_{h,k,0}|_{\Omega_h} \tag{3.4}$$

We also define the sets of all elements that are cut by Γ_D and Γ_N, respectively

$$\Omega_{h,D} = \{\cup K : \overline{K} \cap \Gamma_D \neq \emptyset, \, K \in \mathcal{K}_h\} \tag{3.5}$$

$$\Omega_{h,N} = \{\cup K : \overline{K} \cap \Gamma_N \neq \emptyset, \, K \in \mathcal{K}_h\} \tag{3.6}$$

and the sets of interior faces belonging to elements in $\mathcal{K}_h$ that are cut by Γ_D respectively cut by Γ_N but not by Γ_D

$$\mathcal{F}_{h,D} = \{F \in \mathcal{F}_h : F \cap \overline{\Omega}_{h,D} \neq \emptyset\} \tag{3.7}$$

$$\mathcal{F}_{h,N} = \{F \in \mathcal{F}_h \setminus \mathcal{F}_{h,D} : F \cap \overline{\Omega}_{h,N} \neq \emptyset\} \tag{3.8}$$

which are illustrated in Figure 1d. Note that $\mathcal{F}_{h,D} \cap \mathcal{F}_{h,N} = \emptyset$. For each face $F \in \mathcal{F}_h$ we choose to denote one of the two elements sharing F by K_+ and the other element by K_-. We thus have $F = \overline{K}_+ \cap \overline{K}_-$ and we define the face normal $n_F = n_{\partial K_+}$ and the jump over the face

$$[\![v]\!] = v|_{K_+} - v|_{K_-} \tag{3.9}$$

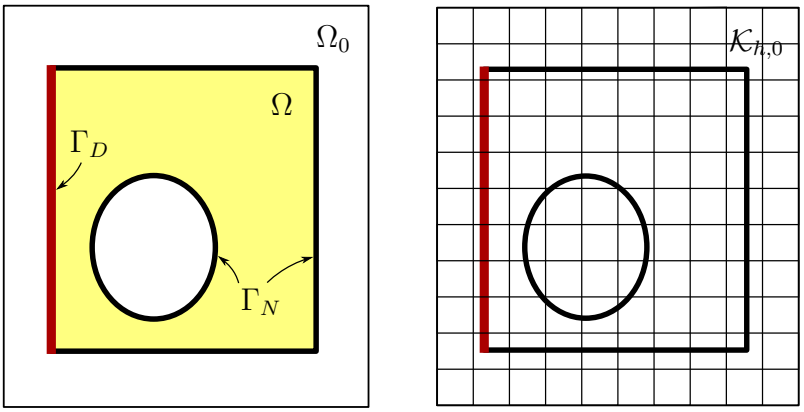
3.2 The Method

Following the procedure in [11] we introduce the stabilized bilinear form

$$a_h(\Omega; u, v) = a(\Omega; u, v) + s_h(\mathcal{F}_{h,D}; u, v) + h^2 s_h(\mathcal{F}_{h,N}; u, v) \tag{3.10}$$

The stabilization form s_h is given by

$$s_h(\mathcal{F}; u, v) = \sum_{F \in \mathcal{F}} \sum_{j=1}^{k} \gamma_j h^{2j-1} ([\![\partial_{n_F}^j u]\!], [\![\partial_{n_F}^j v]\!])_{L^2(F)} \tag{3.11}$$



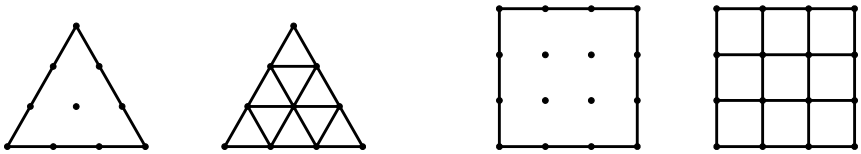
(a) Domain and boundaries

(b) Background grid

(c) Domain of active elements $\mathcal{K}_h$

(d) Sets of faces

Figure 1: Illustration of notation for meshes and faces.



(a) Triangles **(b)** Quadrilaterals

Figure 2: P_3 Lagrange elements and the corresponding mesh for a P_1-iso-P_3 finite element space consisting of P_1 Lagrange elements with vertices at the positions of the P_3 Lagrange nodes.

where $\partial^j_{n_F}$ denotes the j:th derivative in the direction of the face normal n_F and $\{\gamma_j\}_{j=1}^k$ are positive parameters. We also introduce the stabilized Nitsche form

$$A_h(\Omega; u, v) = a_h(\Omega; u, v) - (\sigma(u) \cdot n, v)_{L^2(\Gamma_D)} - (u, \sigma(v) \cdot n)_{L^2(\Gamma_D)} + \gamma_D h^{-1} \left(2\mu(u, v)_{L^2(\Gamma_D)} + \lambda(u \cdot n, v \cdot n)_{L^2(\Gamma_D)}\right) \tag{3.12}$$

where the additional terms give the weak enforcement of the Dirichlet boundary conditions via Nitsche's method [14] and $\gamma_D > 0$ is a parameter.

The cut finite element method for linear elasticity can now be formulated as the following problem: find $u_h \in V_{h,k}(\Omega_h)$ such that

$$A_h(\Omega; u_h, v) = L(v) \qquad \forall v \in V_{h,k}(\Omega_h) \tag{3.13}$$

Theoretical Results. To summarize the main theoretical results from [11] the cut finite element method for linear elasticity has the following properties:

- The stabilized form a_h enjoys the same coercivity and continuity properties with respect to the proper norms on Ω_h as the standard form a on Ω equipped with a fitted mesh.
- Optimal order approximation holds in the relevant norms since there is a stable interpolation operator with an extension operator that in a H^s stable way extends a function from Ω to a neighborhood of Ω containing Ω_h.

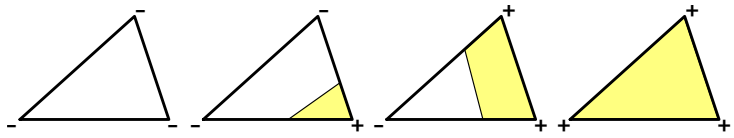
Using these results a priori error estimates of optimal order can be derived using standard techniques of finite element analysis.

3.3 Geometry Description

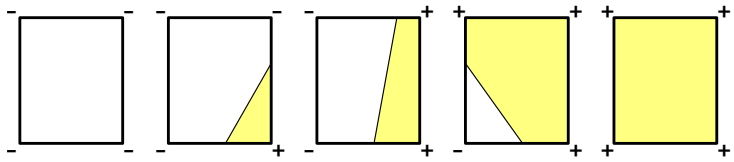
Let the geometry be described via a level-set function $\phi(x) : \Omega_0 \to \mathbb{R}$ where the domain Ω and the domain boundary $\partial\Omega$ are given by

$$\Omega = \{\, x \in \overline{\Omega}_0 \,:\, \phi(x) > 0 \,\} \qquad \text{and} \qquad \partial\Omega = \{\, x \in \overline{\Omega}_0 \,:\, \phi(x) = 0 \,\} \tag{3.14}$$

For convenience we use P_1 finite elements for the level-set representation. However, as we use P_k finite elements to approximate the solution, we improve the geometry representation when using higher order finite elements ($k \geq 2$) by letting the level-set be defined

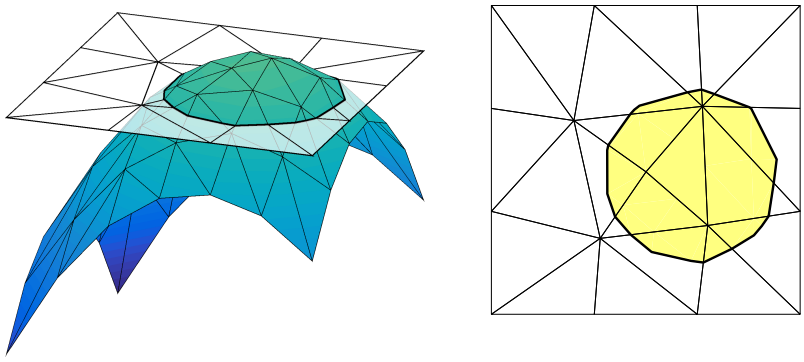


(a) Triangles



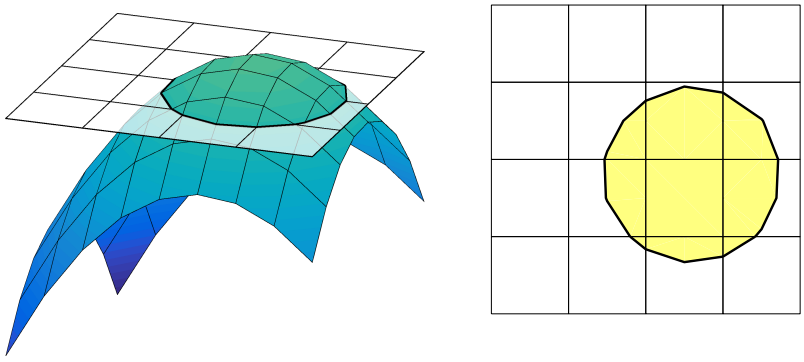
(b) Quadrilaterals

Figure 3: Cases of extracting piecewise linear geometry from a level-set on an element $K \in \mathcal{K}_{h/k,0}$ with the signs of the level-set at the vertices as indicated.



(a) P_1-iso-P_2 level-set

(b) P_1-iso-P_2 geometry



(c) P_1-iso-P_2 level-set

(d) P_1-iso-P_2 geometry

Figure 4: Piecewise linear geometry represented by a discrete level-set using P_1-iso-P_k finite elements.

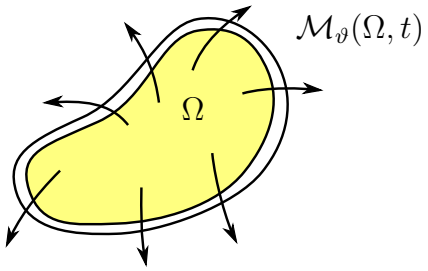


Figure 5: Illustration of the mapping $\mathcal{M}_\vartheta(\Omega, t)$.

on a refined mesh. The refined mesh $\mathcal{K}_{h/k,0}$ is constructed by uniform refinement of $\mathcal{K}_{h,0}$ such that the Lagrange nodes of a P_k finite element in $\mathcal{K}_{h,0}$ coincides with the vertices of $\mathcal{K}_{h/k,0}$, as illustrated in Figure 2. The finite element space of the level-set is the scalar valued so called P_1-iso-P_k finite element space on $\mathcal{K}_{h,0}$, defined as

$$W_{h/k} = \{v \in H^1(\Omega_0) \cap C^0(\Omega_0) \,:\, v|_K \in P_1(K),\ K \in \mathcal{K}_{h/k,0}\} \tag{3.15}$$

Geometry Extraction. We extract the domain Ω from the level-set function $\phi \in W_{h/k}$ by traversing all elements in $\mathcal{K}_{h/k,0}$ and checking the value and sign of ϕ in the element vertices to derive the intersection between the element and the domain. This procedure results in a number of simple cases which for triangles and quadrilaterals are illustrated in Figure 3. Note that in the case of quadrilaterals the boundary intersection $\partial\Omega \cap K$ where $K \in \mathcal{K}_{h/k,0}$ is actually not a linear function as bilinear basis functions include a quadratic cross term, but we employ linear interpolation between detected edge intersections to produce the P_1 boundary illustrated. Example geometry extractions from P_1-iso-P_2 level-sets on triangles and quadrilaterals are shown in Figure 4.

Quadrature. As the possible geometry intersections with elements in the refined mesh $\mathcal{K}_{h/k,0}$ consists of a small number of cases, as illustrated in Figure 3, we construct quadrature rules for exact integration of products of P_k polynomials for each case.

4 Shape Calculus

4.1 Definition of the Shape Derivative

For $\Omega \in \mathcal{O}$ we let $\mathcal{W}(\Omega, \mathbb{R}^d)$ denote the space of sufficiently smooth vector fields on Ω and for a vector field $\vartheta \in \mathcal{W}$ we define the mapping

$$\mathcal{M}_\vartheta : \Omega \times I \ni (x, t) \mapsto x + t\vartheta(x) \in \mathcal{M}_\vartheta(\Omega, t) \subset \mathbb{R}^d \tag{4.1}$$

where $I = (-\delta, \delta)$, $\delta > 0$. This mapping is illustrated in Figure 5. For small enough δ, the mapping $\Omega \mapsto \mathcal{M}_\vartheta(\Omega, t)$ is a bijection and $\mathcal{M}_\vartheta(\Omega, 0) = \Omega$. We also assume that the vector field ϑ is such that $\mathcal{M}_\vartheta(\Omega, t) \in \mathcal{O}$ for $t \in I$ with δ small enough.

Let $J(\Omega)$ be a shape functional, i.e. a mapping $J : \mathcal{O} \ni \Omega \mapsto J(\Omega) \in \mathbb{R}$. We then have the composition $I \ni t \mapsto J \circ \mathcal{M}(\Omega, t) \in \mathbb{R}$ and we define the shape derivative $D_{\Omega,\vartheta}$ of J in the direction ϑ by

$$D_{\Omega,\vartheta} J(\Omega) = \frac{d}{dt} J \circ \mathcal{M}_\vartheta(\Omega, t)|_{t=0} = \lim_{t \to 0} \frac{J(\mathcal{M}_\vartheta(\Omega, t)) - J(\Omega)}{t} \tag{4.2}$$

Note that if $\mathcal{M}_\vartheta(\Omega, t) = \Omega$ we have $D_{\Omega,\vartheta} J = 0$, even if $\mathcal{M}_\vartheta(\omega, t) \neq \omega$ for a proper subspace $\omega \subset \Omega$. We finally define the shape derivative $D_\Omega J : W(\Omega, \mathbb{R}^d) \to \mathbb{R}$ by

$$D_\Omega J(\vartheta) = D_{\Omega,\vartheta} J(\Omega) \tag{4.3}$$

If the functional J depends on other arguments we use ∂_Ω to denote the partial derivative with respect to Ω and $\partial_{\Omega,\vartheta}$ to denote the partial derivative with respect to Ω in the direction ϑ. See e.g. [17, 18] or [10] and references therein.

For $f : \Omega \times I \to \mathbb{R}^d$ we also define the material time derivative in the direction ϑ by

$$D_{t,\vartheta} f = \lim_{t \to 0} \frac{f(\mathcal{M}_\vartheta(x, t), t) - f(x, 0)}{t} \tag{4.4}$$

and recall the identity

$$\partial_{\Omega,\vartheta} \int_\Omega f = \int_\Omega (D_{t,\vartheta} f + (\nabla \cdot \vartheta) f) \tag{4.5}$$

see for example [10].

4.2 Shape Derivative

We want to take the derivative of the inf-sup formulation of the objective function (2.16) with respect to the domain. The Correa–Seeger theorem [9] or [10] states that for $q, v \in V(\Omega)$ we have

$$D_\Omega J(\vartheta) = \partial_{\Omega,\vartheta} \mathcal{L}(\Omega; q, v)|_{v=u(\Omega), q=p(\Omega)} \tag{4.6}$$

where $u(\Omega)$ is the primal solution (2.6), and $p(\Omega)$ is the dual solution (2.15). For the special case when $u(\Omega) = p(\Omega)$ we have

$$D_\Omega J(\vartheta) = \partial_{\Omega,\vartheta} \mathcal{L}(\Omega; v, v)|_{v=u(\Omega)} \tag{4.7}$$

For notational simplicity we below omit the dependency on Ω, i.e., $u = u(\Omega)$ and $p = p(\Omega)$.

Theorem 4.1. *The shape derivative of the compliance objective functional $J(\Omega) = \mathcal{L}(\Omega; u, u)$ defined via (2.20) is given by*

$$\begin{aligned} D_{\Omega,\vartheta} J(\vartheta) = \int_\Omega & 2\mu(\epsilon_\vartheta(u) : \epsilon(u)) + \lambda \mathrm{tr}(\epsilon_\vartheta(u)) \mathrm{tr}(\epsilon(u)) \\ & + (\nabla \cdot \vartheta) \Big(\kappa - \mu(\epsilon(u) : \epsilon(u)) - \frac{1}{2} \lambda \, \mathrm{tr}(\epsilon(u))^2 \Big) \end{aligned} \tag{4.8}$$

where $\epsilon_\vartheta(u)$ is defined as

$$\epsilon_\vartheta(u) = \frac{1}{2} \left(\nabla u \nabla \vartheta + \nabla \vartheta^T \nabla u^T \right) \tag{4.9}$$

Proof. We have

$$\partial_{\Omega,\vartheta}\mathcal{L}(\Omega; v, v) = \partial_{\Omega,\vartheta}\left(\kappa|\Omega| - \frac{1}{2}a(\Omega; v, v)\right) \tag{4.10}$$

from (2.11) and (2.20) and by (4.5) we obtain

$$\partial_{\Omega,\vartheta}\mathcal{L}\left(\kappa|\Omega| - \frac{1}{2}a(v, v)\right) = \partial_{\Omega,\vartheta}\int_\Omega \kappa - \mu\epsilon(v) : \epsilon(v) - \frac{1}{2}\lambda\,\mathrm{tr}(\epsilon(v))^2 \tag{4.11}$$

$$= \int_\Omega D_{t,\vartheta}\left(-\mu\epsilon(v) : \epsilon(v) - \frac{1}{2}\lambda\,\mathrm{tr}(\epsilon(v))^2\right) \tag{4.12}$$

$$+ (\nabla\cdot\vartheta)\left(\kappa - \mu\epsilon(v) : \epsilon(v) - \frac{1}{2}\lambda\,\mathrm{tr}(\epsilon(v))^2\right)$$

where we used $D_{t,\vartheta}\kappa = 0$. We introduce the compact notation $\mathcal{M}_t = \mathcal{M}_\vartheta(\cdot, t)$ for the mapping and $\Omega_t = \mathcal{M}_t\Omega = \mathcal{M}_\vartheta(\Omega, t)$ for the perturbed domain. Letting ∇_t denote the gradient on Ω_t we have the identity

$$\nabla_t(v\circ\mathcal{M}_t^{-1})\circ\mathcal{M}_t = \nabla v(\nabla\mathcal{M}_t)^{-1} \tag{4.13}$$

and thus $\epsilon(v)$ can be parametrized by t as

$$\epsilon_t(v) = \frac{1}{2}\left(\nabla_t(v\circ\mathcal{M}_t^{-1}) + (v\circ\mathcal{M}_t^{-1})^T\nabla_t v^T\right)\circ\mathcal{M}_t \tag{4.14}$$

$$= \frac{1}{2}\left(\nabla v(\nabla\mathcal{M}_t)^{-1} + (\nabla\mathcal{M}_t)^{-T}\nabla v^T\right) \tag{4.15}$$

Using that

$$D_{t,\vartheta}\nabla v(\nabla\mathcal{M}_t)^{-1} = \nabla v D_{t,\vartheta}(\nabla\mathcal{M}_t)^{-1} = -\nabla v\nabla\vartheta \tag{4.16}$$

we obtain

$$D_{t,\vartheta}\epsilon_t(v) = D_{t,\vartheta}\frac{1}{2}\left(\nabla v\nabla\mathcal{M}_t^{-1} + \nabla\mathcal{M}_t^{-T}\nabla v^T\right) \tag{4.17}$$

$$= -\frac{1}{2}\left(\nabla v\nabla\vartheta + \nabla\vartheta^T\nabla v^T\right) \tag{4.18}$$

$$= -\epsilon_\vartheta(v) \tag{4.19}$$

and as a result

$$D_{t,\vartheta}\left(-\mu\epsilon(v) : \epsilon(v) - \frac{1}{2}\lambda\,\mathrm{tr}(\epsilon(v))^2\right) = 2\mu\epsilon_\vartheta(v) : \epsilon(v) + \lambda\,\mathrm{tr}(\epsilon_\vartheta(v))\,\mathrm{tr}(\epsilon(v)) \tag{4.20}$$

which concludes the proof by setting $v = u$. □

4.3 Finite Element Approximation of the Shape Derivative

We compute a discrete approximation to the shape derivative by inserting the cut finite element approximation u_h to (2.6), defined by (3.13), into (4.8). This yields the following expression for the shape derivative approximation

$$\partial_{\Omega,\vartheta}\mathcal{L}(\Omega; u_h, u_h) = \int_\Omega 2\mu(\epsilon_\vartheta(u_h) : \epsilon(u_h)) + \lambda \mathrm{tr}(\epsilon_\vartheta(u_h))\mathrm{tr}(\epsilon(u_h)) \\ + (\nabla\cdot\vartheta)\Big(\kappa - \mu(\epsilon(u_h) : \epsilon(u_h)) - \frac{1}{2}\lambda(\nabla\cdot u_h)(\nabla\cdot u_h)\Big) \tag{4.21}$$

5 Domain Evolution

To construct a robust method for evolving the domain we need the discrete level-set function $\phi \in W_{h/k}$ to be a good approximation to a signed distance function, at least close to the boundary, i.e.,

$$\phi(x) \approx \begin{cases} \rho(x, \partial\Omega), & x \notin \Omega \\ -\rho(x, \partial\Omega), & x \in \Omega \end{cases} \tag{5.1}$$

where $\rho(x, \partial\Omega)$ is the smallest Euclidean distance between the point x and the boundary $\partial\Omega$. We formulate a method to evolve ϕ using the transport equation

$$\partial_t\phi + \beta\cdot\nabla\phi = 0 \tag{5.2}$$

where β is a velocity field which we compute based on the shape derivative. After some time ϕ will no longer represent a signed distance function and ϕ and so called reinitialization needs to be performed. There are several algorithms for redistancing level-set functions, see e.g. [1, 15, 19, 21].

5.1 Shape Evolution

Computing the Velocity Field $\boldsymbol{\beta}$. Consider the bilinear form

$$b(\beta, \vartheta) = (\beta, \vartheta)_{L^2(\Omega_0)} + c_1(\nabla\beta, \nabla\vartheta)_{L^2(\Omega_0)} \tag{5.3}$$

where $c_1 > 0$ is a parameter used for setting the amount of regularization. We want to find the velocity field β that satisfy

$$\min_{\sqrt{b(\beta,\beta)}=1} D_{\Omega,\beta}J(\Omega) \tag{5.4}$$

This is equivalent to solving: find $\beta' \in [W_{h/k}]^d$ such that

$$b(\beta', \vartheta) = -D_{\Omega,\vartheta}J(\Omega) \qquad \forall\vartheta \in [W_{h/k}]^d \tag{5.5}$$

and set

$$\beta = \beta' / \sqrt{b(\beta', \beta')} \tag{5.6}$$

see for example [6]. As boundary conditions on β' we use

$$\beta' \cdot n = 0 \quad \text{and} \quad (I - n \otimes n)(\nabla \beta')n = 0 \qquad \text{on } \partial\Omega_0 \tag{5.7}$$

where $(I - n \otimes n)$ is the projection onto the tangential plane of the boundary. This condition allows for a velocity tangential to the boundary.

Evolving the Level-Set ϕ. The evolution of the domain Ω over a pseudo-time step T is computed by solving the following convection equation

$$(\partial_t \phi, v)_{L^2(\Omega_0)} + (\beta \cdot \nabla \phi, v)_{L^2(\Omega_0)} + c_2 \sum_{F \in \mathcal{F}_h} h^2([\![\partial_{n_F} \phi]\!], [\![\partial_{n_F} v]\!])_{L^2(F)} = 0, \quad t \in (0, T] + t_0 \tag{5.8}$$

where c_2 is a stabilization parameter. For time integration we use a Crank–Nicolson method and the time step T is chosen via the optimization algorithm described in the next section.

5.2 Optimization Algorithm

In this section we propose an algorithm that solve (2.10) and give a overview of the optimization procedure. To find the descent direction of the boundary we first use sensitivity analysis to compute a discrete shape derivative (4.21). Next we compute a velocity corresponding to the greatest descent direction of shape velocity subject to some regularity constraints (5.4). Finally, we use use the velocity field to evolve the domain by moving the level-set function (5.8). This procedure is then repeated in the optimization algorithm.

Algorithm 1 Optimization algorithm

1: Compute the velocity field/decent direction β_h
2: **if** $J(\Omega_{t+T}) < J(\Omega_t)$ **then**
3: $\quad t = t + T$
4: $\quad T = 2T$
5: **else**
6: $\quad$ **while** $J(\Omega_{t+T}) > J(\Omega_t)$ **do**
7: $\quad\quad T = T/2$
8: $\quad$ **end while**
9: $\quad t = t + T$
10: **end if**
11: Reinitialize ϕ_h to a distance function on Ω_{t+T}
12: Go back to 1

6 Numerical Results

6.1 Model Problems

For our numerical experiments we consider three model problems where we perform shape optimization using CutFEM to optimize the following designs with respect to compliance:

- Cantilever beam under traction load
- L-shape beam under traction load
- Wheel under imposed displacements

The design domains, boundary conditions and initial states for both traction problems are described in Figure 6 and for the wheel in Figure 7. We assume that the material is linear elastic isotropic with a Young's modulus $E = 10^4$ and a Poisson's ratio 0.3. The traction load density in the first two problems is $g = [0, -20]^T$ N/m. As objective functional we use compliance (2.20) with a material penalty $\kappa = 35$.

6.2 Implementation Aspects

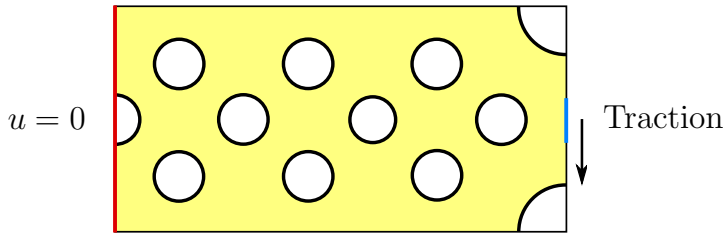
Finite Elements. The CutFEM formulation works independently of the type of $H^1(\Omega_0)$ finite element and in the experiments below we use Lagrange elements of order $k \in \{1, 2, 4\}$ on both triangles and quadrilaterals. We have decided to use P_1-iso-P_k finite elements for the level-set representation. It is possible to use higher order approximations of the level-set, however for the level-set to both describe a quadratic (or higher order) approximation of the interface and a distance function is not possible in straightforward manner, which in turn can affects the accuracy of the transport equation.

Parameter Choices. To produce our numerical results we set $c_1 = 3(h/k)^2$ in (5.3) and $c_2 = 0.1$ in (5.8). Note that h/k in the expression for c_1 is the mesh size of the level-set mesh. We set the penalty parameter $\gamma_D = 10k^2(\mu + \lambda)$ in (3.12) and $\gamma_j = 10^{-7}(\mu + \lambda)$ in (3.11).

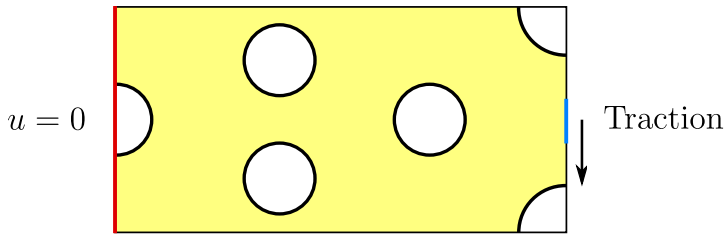
Disconnected Domains. As the level-set description allow for topological changes it is possible for small parts of the domain to become disjoint. To remedy this we use a simple filtering strategy to remove these disjoint parts which is based on properties of the direct solver. It is however possible to construct other filtering strategies instead, for example based on the connectivity of the stiffness matrix.

6.3 Numerical Experiments

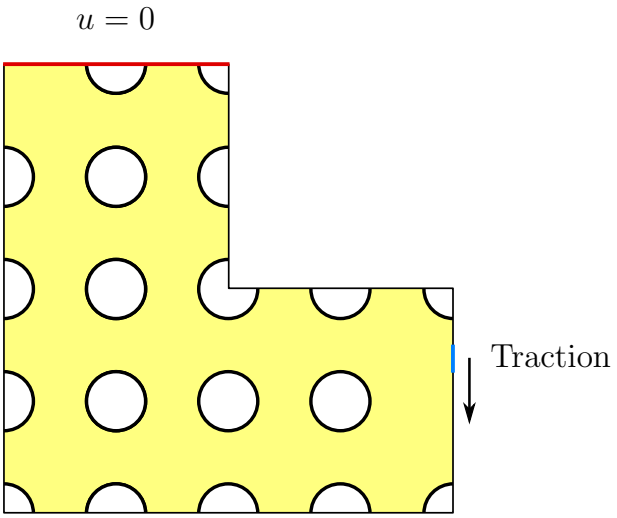
Cantilever Beam. For the cantilever beam with initial data described in Figure 6a we give results for different k when h/k is kept constant on triangles in Figure 8 and on quadrilaterals in Figure 9. The plotted meshes are those used for representing the solution



(a) Cantilever beam 1

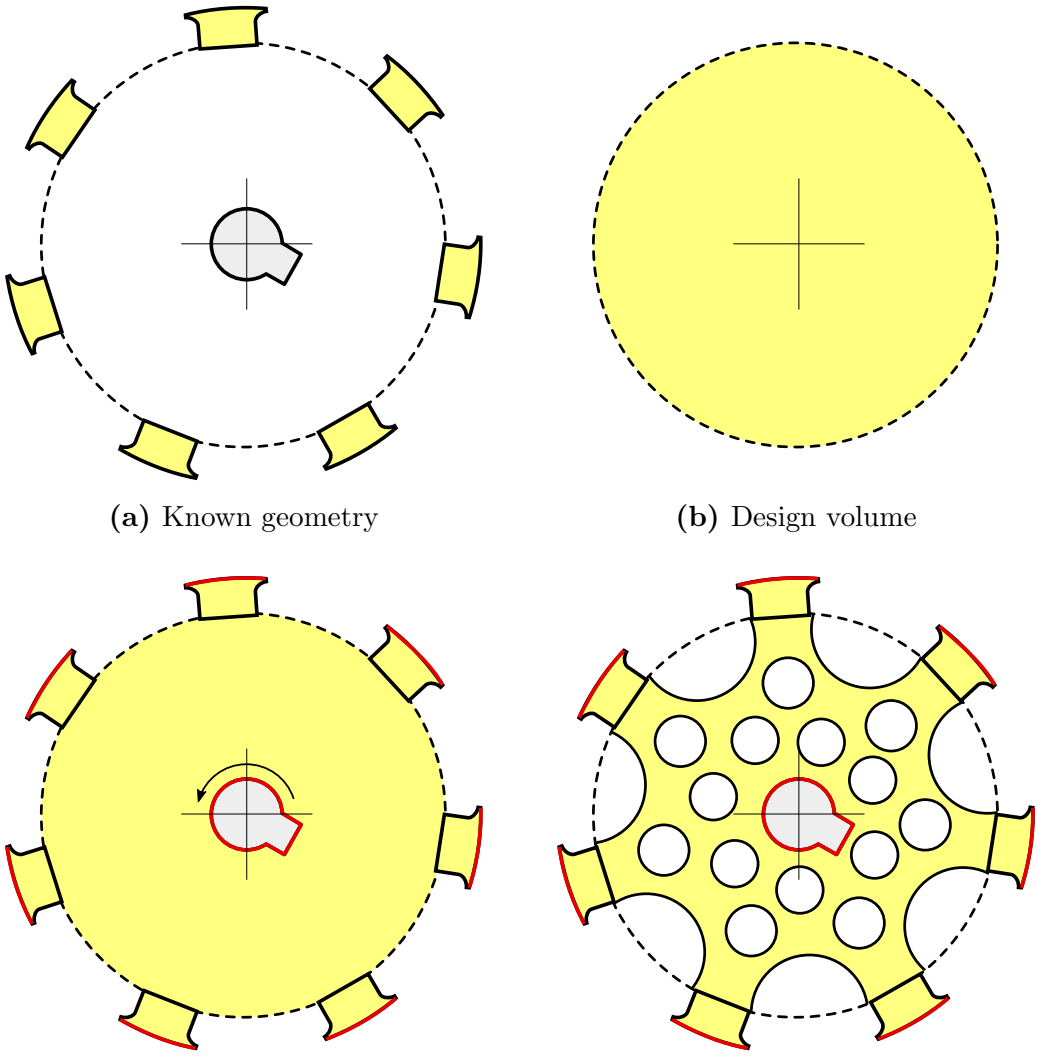


(b) Cantilever beam 2



(c) L-shape beam

Figure 6: Schematic figures of the design domains, boundary conditions and initial states used in the two traction model problems. (a) *Cantilever beam:* The design domain is a $2\,\mathrm{m} \times 1\,\mathrm{m}$ rectangle and the traction load density on the right is evenly distributed within $\pm 0.1\,\mathrm{m}$ from the horizontal center line. (b) *L-shape beam:* The design domain is a $2\,\mathrm{m} \times 2\,\mathrm{m}$ square with a $1\,\mathrm{m} \times 1\,\mathrm{m}$ square void in the top right corner. The traction load density on the right is evenly distributed in the interval $[5/16, 1/2]\,\mathrm{m}$ from the bottom.



(a) Known geometry

(b) Design volume

(c) Computational domain

(d) Initial guess

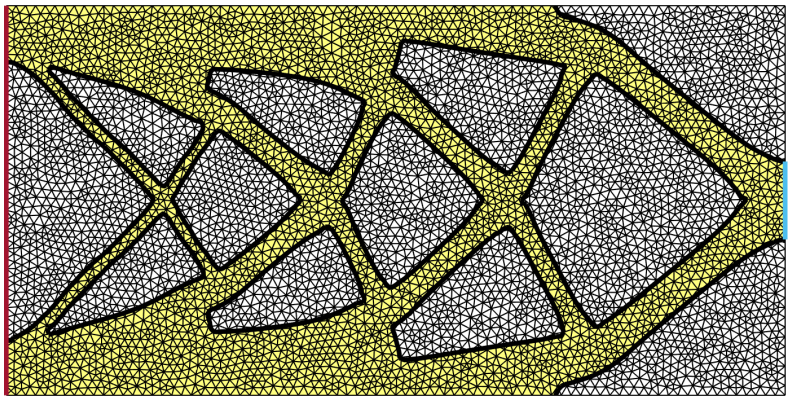
Figure 7: Schematic figures illustrating the wheel model problem where certain parts of the design is known. (a) The known geometric features in this example are the seven "feet" which are always added to the computational domain and the center "keyhole" which is always removed from the domain. (b) The design volume which we seek to optimize is the illustrated disc. While we also could remove the keyhole from the design volume we in practice only remove elements completely inside the keyhole. Thus, the geometric removal operation would still have to be performed in each iteration. (c) The total computational domain consists of the design volume with the known geometric features imposed, here illustrated with boundary conditions. On the outer part of the feet we impose homogeneous Dirichlet conditions, $u = 0$, while we on the keyhole impose the non-homogeneous Dirichlet condition $u = 0.3[-x_2, x_1]^T$. As indicated in the figure the latter condition on the displacement field corresponds to a linearized rotation of the boundary. (d) A number of holes in the design volume is added as an initial guess for the iteration procedure.

u while the mesh resolution for the level-set description of the geometry is constant in all the examples in each figure. Note that while the solutions are symmetric around the horizontal centerline we do not impose symmetry in the algorithm. We also give an example of the resulting stresses and displacements in Figure 8c. In the case of the final geometry obtain using $k = 2$ and $h = 0.025$ in Figure 9b we present the decay of the objective function in Figure 10. We perform 50 iterations for each example.

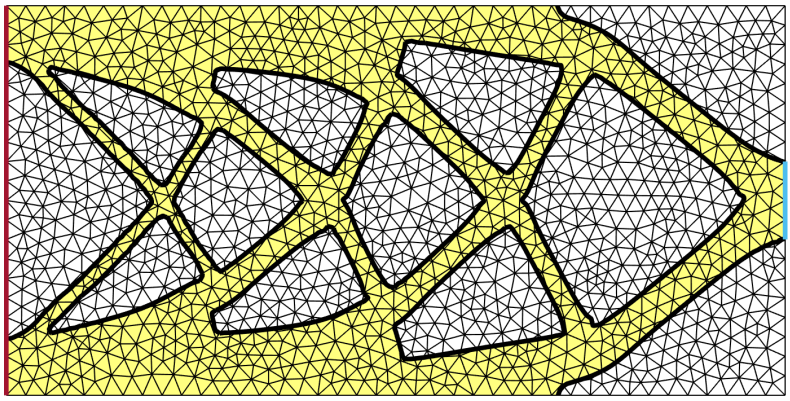
For the cantilever beam with initial data described in Figure 6b we give results for different h when $k = 1$ on quadrilaterals in Figure 11. We note that the results are stable with respect to refinement of h as soon as the mesh is sufficiently fine to be able to represent the thin structures occuring in the optimal solution. When the mesh is too coarse, see Figure 11a, we can not represent the geometry and we therefore get a solution with a different topology. Note also that we have imposed a cost for material in the objective functional which is the same on all meshes and since the topology is different we get a larger total amount of material in this case.

L-shape Beam. For the L-shape beam problem described in Figure 6c we give results on quadratic triangles in Figure 12. Note that in this example we actually have topological changes between the initial and final geometries. We illustrate this in the sequence presented in Figure 13. We perform 50 iterations.

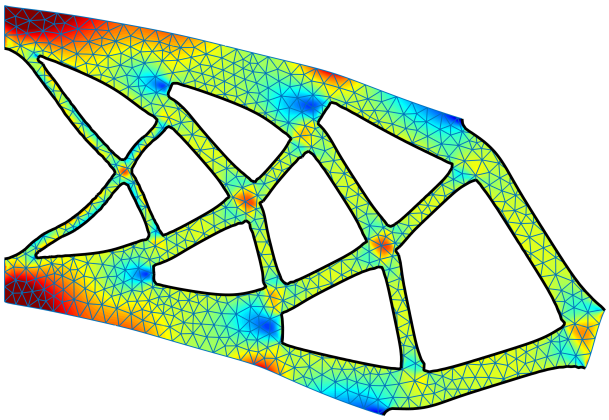
Wheel. For the wheel displacement problem described in Figure 7 we give results on quadratic quadrilaterals in Figure 14. We illustrate that final volume in this in the sequence presented in Figure 13. We perform 50 iterations.



(a) $k = 2$, $h = 0.025$

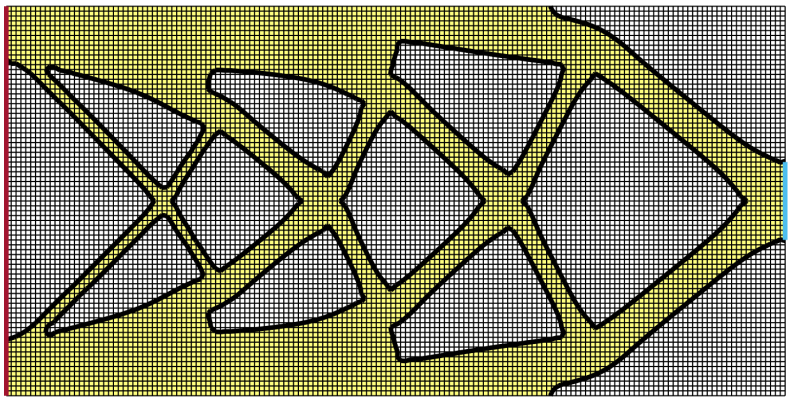


(b) $k = 4$, $h = 0.05$

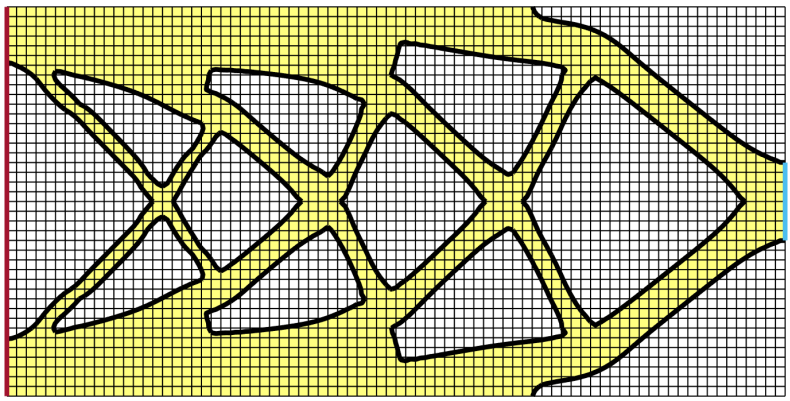


(c) $k = 4$, $h = 0.05$

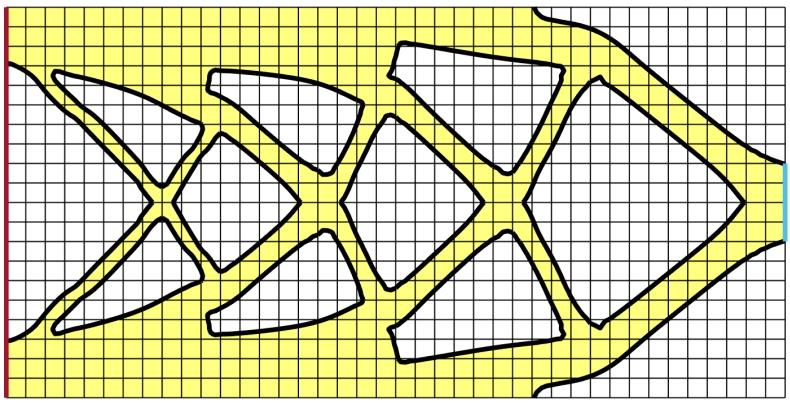
Figure 8: (a)–(b) Final geometries for the cantilever beam problem using triangle elements. The mesh size for the level-set geometry description is h/k and kept constant between the two examples. (c) Visualization of (exaggerated) displacements and von-Mises stresses when using the final geometry in (b).



(a) $k = 1$, $h = 0.0125$



(b) $k = 2$, $h = 0.025$



(c) $k = 4$, $h = 0.05$

Figure 9: Final geometries for the cantilever beam problem using quadrilateral elements. The mesh size for the level-set geometry description is h/k and kept constant between the three examples.

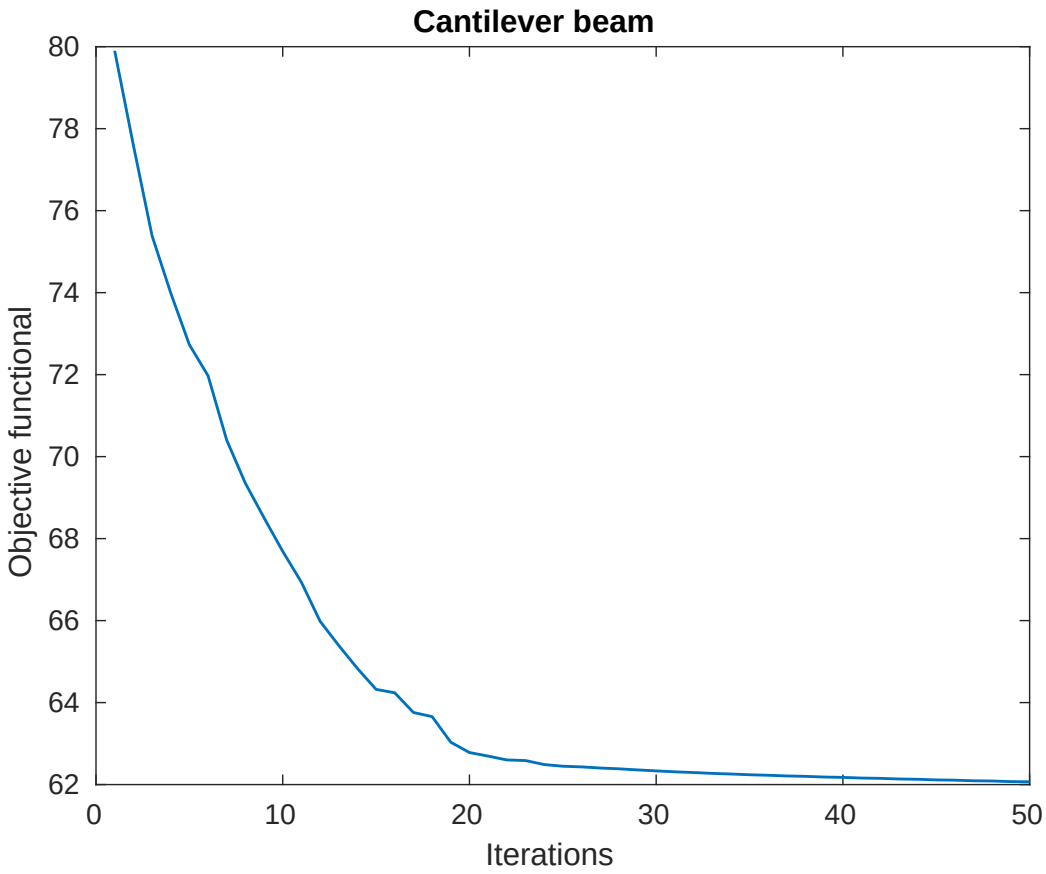
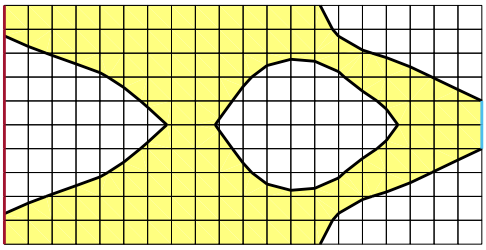
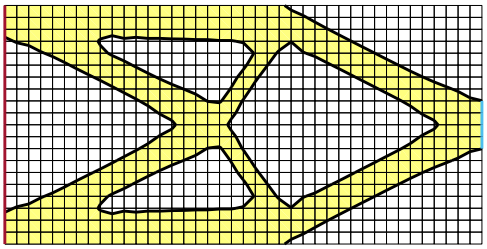


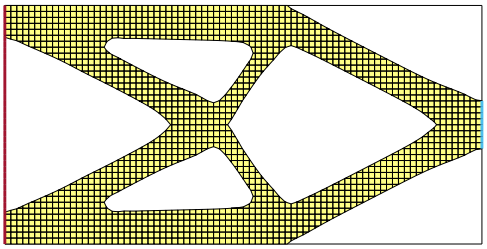
Figure 10: The decay of the objective function for the cantilever beam using $k = 2$ and $h = 0.025$. The final geometry of the cantilever beam is given in Figure 9b.



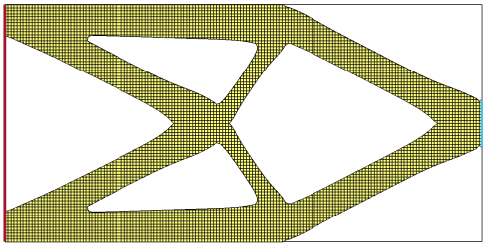
(a) $k = 1$, $h = 0.1$



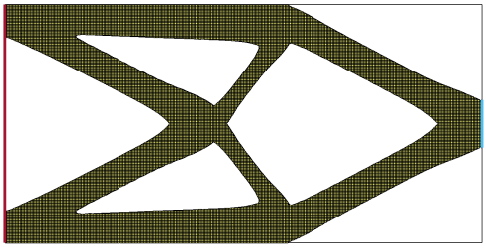
(b) $k = 1$, $h = 0.5$



(c) $k = 1$, $h = 0.025$

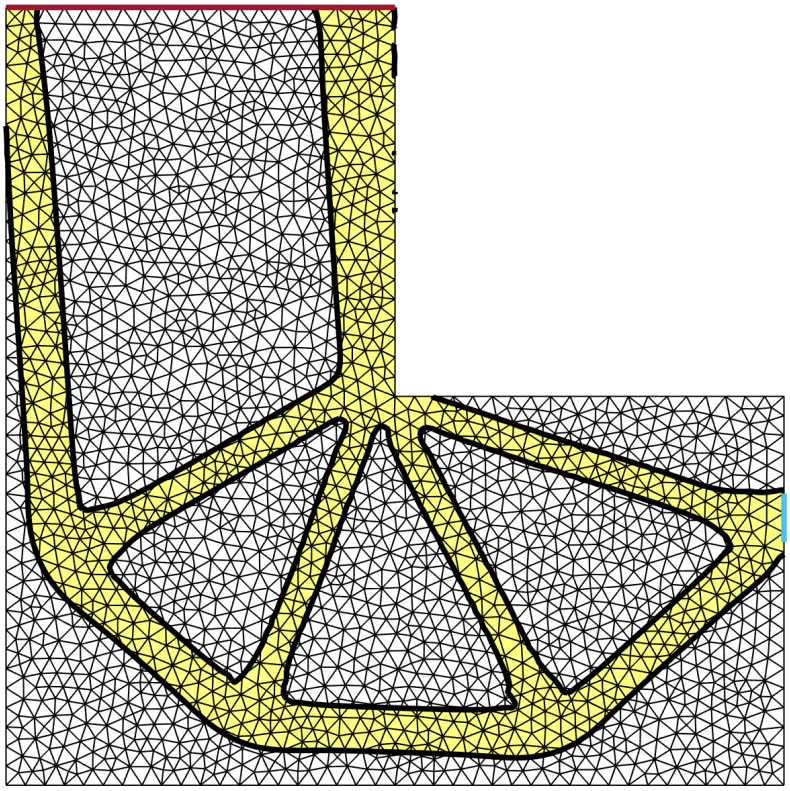


(d) $k = 1$, $h = 0.00125$



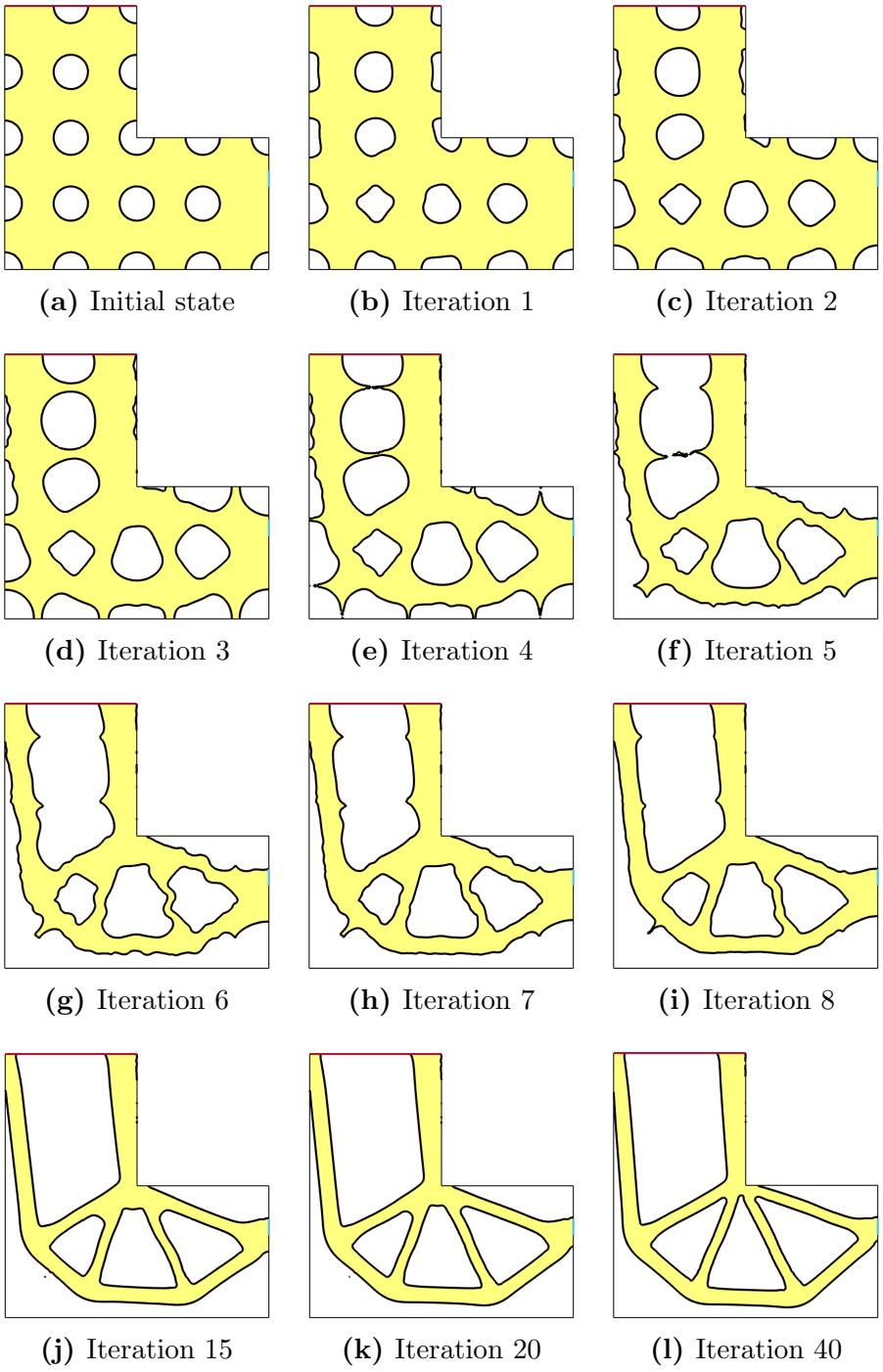
(e) $k = 1$, $h = 0.000625$

Figure 11: Final geometries for the cantilever beam problem using quadrilateral elements. The polynomial degree $k = 1$ is kept constant and the mesh size h is decreased in the examples. Note that in (a) the mesh is too coarse to represent the optimal solution and therefore we get a solution with a simpler topology. As soon as the mesh is sufficiently fine we get a stable solution as in (b)–(e).



(a) $k = 2$, $h = 0.05$

Figure 12: Final geometry for the L-shape beam problem on triangles.



(a) Initial state **(b)** Iteration 1 **(c)** Iteration 2

(d) Iteration 3 **(e)** Iteration 4 **(f)** Iteration 5

(g) Iteration 6 **(h)** Iteration 7 **(i)** Iteration 8

(j) Iteration 15 **(k)** Iteration 20 **(l)** Iteration 40

Figure 13: Sample iteration states in the L-shape beam problem ($k = 2$, $h = 0.05$). Note the topology changes occurring when void separations break.

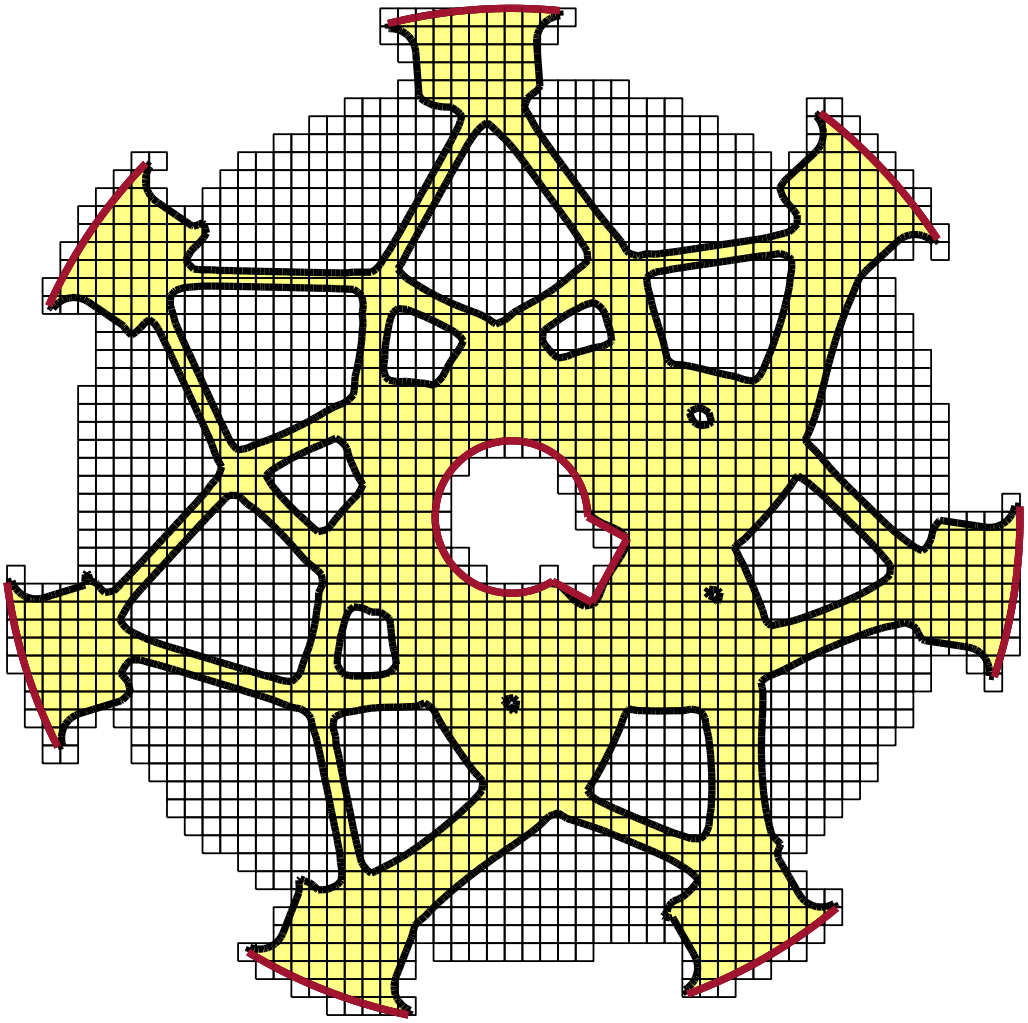


Figure 14: Final geometry for the wheel model on a quadrilateral mesh.

7 Conclusions

We developed, implemented, and demonstrated a shape and topology optimization algorithm for the linear elasticity compliance problem based on:

- A cut finite element method for linear elasticity with higher order polynomials on triangles and quadrilaterals.
- A piecewise linear or bilinear level-set representation of the boundary. In the case of higher order polynomials we use a refined mesh for the level-set representation to allow a more flexible and complex geometric variation and utilize the additional accuracy of the higher order elements.
- A Hamilton-Jacobi transport equation to update the geometry with a velocity field given by the largest descending direction of the shape derivative with a certain regularity requirement.

Our numerical examples demonstrate the performance of the method and show, in particular, that when using higher order elements and a level-set on a refined mesh fine scale geometric features of thickness smaller than the element size can be represented and stable and accurate solutions are produced by the cut finite element method in each of the iterations.

Directions for future work include:

- Fine tuning of the local geometry based on stress measures.
- More general design constraints.
- Use of adaptive mesh refinement to enhance local accuracy of the solution and geometry representation.

References

[1] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269 – 277, 1995.

[2] G. Allaire, C. Dapogny, and P. Frey. Shape optimization with a level set based mesh evolution method. *Comput. Methods Appl. Mech. Engrg.*, 282:22–53, 2014.

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

[4] M. P. Bendsøe and O. Sigmund. *Topology optimization*. Springer-Verlag, Berlin, 2003. Theory, methods and applications.

[5] E. Burman, S. Claus, P. Hansbo, M. G. Larson, and A. Massing. CutFEM: discretizing geometry and partial differential equations. *Internat. J. Numer. Methods Engrg.*, 104(7):472–501, 2015.

[6] E. Burman, D. Elfverson, P. Hansbo, M. G. Larson, and K. Larsson. A cut finite element method for the bernoulli free boundary value problem. *Computer Methods in Applied Mechanics and Engineering*, 317:598 – 618, 2017.

[7] E. Burman and P. Hansbo. Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method. *Appl. Numer. Math.*, 62(4):328–341, 2012.

[8] P. W. Christensen and A. Klarbring. *An introduction to structural optimization*, volume 153 of *Solid Mechanics and its Applications*. Springer, New York, 2009.

[9] R. Correa and A. Seeger. Directional derivates in minimax problems. *Numer. Funct. Anal. Optim.*, 7(2-3):145–156, 1984/85.

[10] M. Delfour and J. Zolésio. *Shapes and Geometries*. Society for Industrial and Applied Mathematics, second edition, 2011.

[11] P. Hansbo, M. G. Larson, and K. Larsson. Cut finite element methods for linear elasticity problems. Technical report, Umeå University, Dept. Mathematics and Mathematical Statistics, 2016.

[12] R. Hiptmair and A. Paganini. Shape optimization by pursuing diffeomorphisms. *Comput. Methods Appl. Math.*, 15(3):291–305, 2015.

[13] R. Hiptmair, A. Paganini, and S. Sargheini. Comparison of approximate shape gradients. *BIT*, 55(2):459–485, 2015.

[14] J. Nitsche. Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abh. Math. Sem. Univ. Hamburg*, 36:9–15, 1971.

[15] J. A. Sethian. Fast-marching level-set methods for three-dimensional photolithography development, 1996.

[16] J. A. Sethian and A. Wiegmann. Structural boundary design via level set and immersed interface methods. *J. Comput. Phys.*, 163(2):489–528, 2000.

[17] J. Simon. Differentiation with respect to the domain in boundary value problems. *Numerical Functional Analysis and Optimization*, 2(7-8):649–687, 1980.

[18] J. Sokolowski and J.-P. Zolesio. *Introduction to shape optimization*. Springer, 1992.

[19] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flow. *Journal of Computational Physics*, 114(1):146 – 159, 1994.

[20] M. Y. Wang, X. Wang, and D. Guo. A level set method for structural topology optimization. *Comput. Methods Appl. Mech. Engrg.*, 192(1-2):227–246, 2003.

[21] H. Zhao. A fast sweeping method for eikonal equations. *Math. Comp.*, 74(250):603–627, 2005.