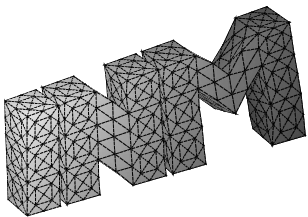

Boundary element based multiresolution shape
optimisation in electrostatics

K. Bandara, F. Cirak, G. Of, O. Steinbach, J. Zapletal



**Berichte aus dem
Institut für Numerische Mathematik**

Technische Universität Graz

Boundary element based multiresolution shape optimisation in electrostatics

K. Bandara, F. Cirak, G. Of, O. Steinbach, J. Zapletal

**Berichte aus dem
Institut für Numerische Mathematik**

Bericht 2014/1

Technische Universität Graz
Institut für Numerische Mathematik
Steyrergasse 30
A 8010 Graz

WWW: <http://www.numerik.math.tu-graz.at>

© Alle Rechte vorbehalten. Nachdruck nur mit Genehmigung des Autors.

Boundary element based multiresolution shape optimisation in electrostatics

K. Bandara¹, F. Cirak¹, G. Of², O. Steinbach², J. Zapletal³

¹Department of Engineering, University of Cambridge,
Trumpington Street, Cambridge CB2 1 PZ, United Kingdom
kkmb2@cam.ac.uk, f.cirak@eng.cam.ac.uk

²Institute of Computational Mathematics, Graz University of Technology,
Steyrergasse 30, 8010 Graz, Austria
of@tugraz.at, o.steinbach@tugraz.at

³Department of Applied Mathematics, VŠB Technical University of Ostrava,
17. listopadu 15/2172, 708 33 Ostrava-Poruba, Czech Republic
jan.zapletal@vsb.cz

Abstract

We consider the shape optimisation of high-voltage devices subject to electrostatic field equations by combining fast boundary elements with multiresolution subdivision surfaces. The geometry of the domain is described with subdivision surfaces and different resolutions of the same geometry are used for optimisation and analysis. The primal and adjoint problems are discretised with the boundary element method using a sufficiently fine control mesh. For shape optimisation the geometry is updated starting from the coarsest control mesh with increasingly finer control meshes. The multiresolution approach effectively prevents the appearance of non-physical geometry oscillations in the optimised shapes. Moreover, there is no need for mesh regeneration or smoothing during the optimisation due to the absence of a volume mesh. We present several numerical experiments and one industrial application to demonstrate the robustness and versatility of the developed approach.

1 Introduction

The shape optimisation of high-voltage electrical devices, such as switchgear or transformers, serves as the driving application for our work. The prevention of electrical breakdown is one of the key considerations in the design of high-voltage devices [1, 2]. In a first approximation,

limiting the electric field strength on critical components can reduce a device's susceptibility to electric breakdown. The electric field strength is determined with the electrostatic field equations, which in absence of space charges reduce to the Laplace equation with Dirichlet boundary conditions [3]. By optimising the shape of critical components the maximum electric field strength on the surface, i.e., the normal flux, can often be considerably reduced. This may make it possible to shrink the size of a device and in turn lead to cost savings. In the approach introduced in this paper we systematically optimise the geometry of a device such that a cost functional consisting of the L_2 norm of the electric field strength is minimised. The boundary element method (BEM) has clear advantages when applied to shape optimisation of high-voltage devices, see [4, 5, 6, 7, 8] for an introduction to BEM. First of all, BEM relies only on a surface discretisation so that there is no need to maintain an analysis-suitable volume discretisation during the shape optimisation process. Moreover, BEM is ideal for solving problems in unbounded domains that occur in electrostatic field analysis. In gradient-based shape optimisation the shape derivative of the cost functional with respect to geometry perturbations is needed [9, 10, 11]. To this purpose, we use the adjoint approach and solve the primary and the adjoint boundary value problems with BEM. The associated linear systems of equations are dense and an acceleration technique, such as the fast multipole method [12, 13], is necessary for their efficient solution. For some recent applications of fast BEM in shape optimisation and Bernoulli-type free-boundary problems we refer to [14, 15, 16].

The geometry parameterisation and its interplay with the BEM surface discretisation plays a crucial role in shape optimisation. When the BEM surface mesh is used for geometry parameterisation it leads to non-physical oscillations in the optimised geometry, as already known in the finite element literature [17, 18]. In addition, the BEM mesh may become severely distorted after a few optimisation steps so that auxiliary mesh smoothing procedures become necessary. To remedy both difficulties, geometries in shape optimisation are commonly parameterised with b-splines or related techniques, such as NURBS and subdivision surfaces [17, 18, 19, 20, 21]. In this paper we represent geometries with subdivision surfaces, which are the generalisation of splines to arbitrary connectivity meshes. Specifically, we use the Loop scheme based on triangular meshes and quartic box-splines [22].

In subdivision schemes a limit surface is obtained through the repeated refinement of a coarse control mesh [23]. In practice, there are closed form expressions for computing the limit surface for a given control mesh [24, 25]. The hierarchy of control meshes underlying a subdivision surface lends itself naturally to multiresolution editing [26, 27]. The coarse control mesh vertex positions are modified to perform large-scale editing and the fine control mesh vertex positions are modified to add localised changes. In the introduced multiresolution shape optimisation approach we use a fine control mesh for BEM discretisation and coarser control meshes for geometry modification. More precisely, we start optimising with the coarsest control mesh and progress to optimise increasingly finer control meshes. As our numerical examples demonstrate, the multiresolution optimisation approach does not lead

to non-physical oscillations in geometry. Moreover, the occurrence of mesh pathologies, like inverted elements, is greatly reduced because the support size of the geometry modifications and the element sizes are well coordinated.

This paper is organised as follows. In Section 2 we introduce the electrostatic shape optimisation problem and the required shape derivatives. We then discuss in Section 3 the discretisation of the state and adjoint boundary value problems with the BEM. Subsequently, in Section 4 the multiresolution subdivision surfaces for geometry parameterisation are explained. The multiresolution optimisation algorithm is introduced in Section 5. Finally, in Section 6 we present several numerical examples with increasing complexity to demonstrate the efficiency and robustness of the proposed approach.

2 Electrostatic shape optimisation problem

The electrostatic field equations in absence of space charges lead to a Dirichlet boundary value problem for the Laplace equation

$$\begin{cases} -\Delta u = 0 & \text{in } \Omega, \\ u = 0 & \text{on } \Gamma_0, \\ u = 1 & \text{on } \Gamma_f, \end{cases} \quad (2.1)$$

where u is the electric potential or voltage, $\Omega \subset \mathbb{R}^3$ denotes a multiply connected, bounded Lipschitz domain with the boundary $\Gamma := \partial\Omega$ consisting of a free part Γ_f and a fixed part Γ_0 . In this paper, we assume that the potentials on Γ_f and Γ_0 are constant. The geometry of the free boundary Γ_f is to be determined with shape optimisation. We assume that the topology of Ω is as shown in Figure 1, i.e., that Γ_0 and Γ_f are disconnected parts of the boundary and that Γ_f is interior to Γ_0 . It is straightforward to generalise our approach to other situations. Moreover, it is well known that the Dirichlet problem (2.1) admits a unique solution $u \in H^1(\Omega)$. For the purposes of the shape calculus introduced below, however, we assume a higher regularity of the solution u in the vicinity of the free part of the boundary Γ_f .

In electrostatic shape optimisation one may seek to minimise the pointwise maximum of the normal flux on the free part of the boundary Γ_f . The corresponding cost functional reads

$$J_{\max}(\Omega, u) := \sup_{\mathbf{x} \in \Gamma_f} \left| \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) \right| \quad (2.2)$$

with the normal flux $\partial u / \partial \mathbf{n} = \langle \nabla u, \mathbf{n} \rangle$ and the exterior unit boundary normal \mathbf{n} . Recall that the physical interpretation of the normal flux is the electric field in normal direction. Due to the non-smooth nature of the max function in (2.2) we decided to consider an alternative

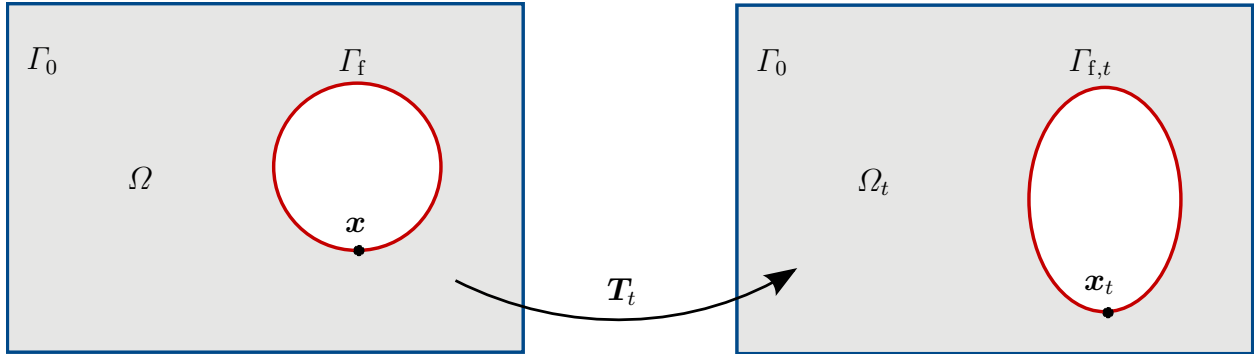


Figure 1: Topology of the domain Ω and its transformation to the domain Ω_t .

differentiable functional in order to be able to exploit methods of the first order calculus; namely

$$J(\Omega, u) := \frac{1}{2} \left\| \frac{\partial u}{\partial \mathbf{n}} - Q \right\|_{L^2(\Gamma_f)}^2 = \frac{1}{2} \int_{\Gamma_f} \left(\frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) - Q \right)^2 \mathrm{d}\mathbf{s}_{\mathbf{x}} \quad (2.3)$$

where $Q \geq 0$ is a prescribed constant expected value. Similar cost functionals are also considered in the context of Bernoulli-type free-boundary problems [16, 28].

During the iterative shape optimisation (see Section 5) the derivatives of the cost functional (2.3) with respect to domain perturbations are needed. To this end, we make use of shape calculus methods as introduced, e.g., in [9, 11]. First, in order to describe geometry changes of the domain Ω , we define a family of mappings $\mathbf{T}_t : \Omega \rightarrow \mathbb{R}^3$ as the perturbation of identity, i.e., $\mathbf{T}_t := \mathbf{I} + t\mathbf{V}$ with the pseudo-time parameter $t \in [0, \tau)$ and some velocity field \mathbf{V} . The new configuration of the domain Ω at pseudo-time t is as illustrated in Figure 1 given by

$$\Omega_t := \mathbf{T}_t(\Omega) := \{\mathbf{T}_t(\mathbf{x}) : \mathbf{x} \in \Omega\}.$$

Under reasonable regularity assumptions on the velocity field \mathbf{V} , say $\mathbf{V} \in \mathbf{W}^{1,\infty}(\Omega)$, and for τ small enough, \mathbf{T}_t can be shown to be a one-to-one mapping of Ω onto Ω_t preserving the Lipschitz regularity, see, e.g., [10, 11]. The shape optimisation problem is thus transformed to a search for a suitable velocity field \mathbf{V} that reduces the cost functional. Following the structure theorem by Hadamard and Zolésio, see, e.g., [9, Chapter 9], one only has to define the mapping on the boundary. Moreover, since Γ_0 is fixed, we can define $\mathbf{V}|_{\Gamma_0} := \mathbf{0}$, i.e., $\Gamma_{0,t} := \mathbf{T}_t(\Gamma_0) = \Gamma_0$. The shape optimisation problem is solved by an iterative procedure and a suitable velocity field \mathbf{V} has to be determined in every iteration step.

Together with the original boundary value problem (2.1) we consider the problem

$$\begin{cases} -\Delta u_t = 0 & \text{in } \Omega_t, \\ u_t = 0 & \text{on } \Gamma_0, \\ u_t = 1 & \text{on } \Gamma_{f,t} \end{cases} \quad (2.4)$$

in the transformed domain. The value of the cost functional (2.3) in this configuration changes to

$$J(\Omega_t, u_t) = \frac{1}{2} \int_{\Gamma_{f,t}} \left(\frac{\partial u_t}{\partial \mathbf{n}_t}(\mathbf{x}_t) - Q \right)^2 d\mathbf{s}_{\mathbf{x}_t}.$$

To be able to use first order minimisation techniques, the shape derivative of the cost functional in the direction of the velocity field \mathbf{V} given by

$$J'(\Omega, u)(\mathbf{V}) := \left. \frac{d}{dt} J(\Omega_t, u_t) \right|_{t=0} = \lim_{t \rightarrow 0^+} \frac{J(\Omega_t, u_t) - J(\Omega, u)}{t}$$

is needed. Under reasonable regularity assumptions (see [9, 11]), the shape derivative can be represented in the so called Hadamard–Zolésio form

$$J'(\Omega, u)(\mathbf{V}) = \int_{\Gamma_f} g(\mathbf{x}) \langle \mathbf{V}(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle d\mathbf{s}_{\mathbf{x}}$$

where $\langle \mathbf{V}(\mathbf{x}), \mathbf{n}(\mathbf{x}) \rangle$ denotes the scalar product of the two involved vectors and g is some kernel function which is independent of the velocity field \mathbf{V} . Note that setting

$$\mathbf{V}|_{\Gamma_f} := -g\mathbf{n} \quad \text{implies} \quad J'(\Omega, u)(\mathbf{V}) = - \int_{\Gamma_f} g^2(\mathbf{x}) d\mathbf{s}_{\mathbf{x}} \leq 0 \quad (2.5)$$

and thus such a velocity field \mathbf{V} defines a steepest descent direction. This structure will be exploited in Section 5 for iterative shape optimisation.

Finally, in the case of the considered cost functional (2.3) the kernel function g takes the form

$$g(\mathbf{x}) := -\frac{\partial p}{\partial \mathbf{n}}(\mathbf{x}) \frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) - \frac{H(\mathbf{x})}{2} \left(\left(\frac{\partial u}{\partial \mathbf{n}}(\mathbf{x}) \right)^2 - Q^2 \right) \quad (2.6)$$

according to [9, 16, 28]. Here H is the additive curvature, u is the primal solution given by (2.1), and p is the solution of the adjoint boundary value problem

$$\begin{cases} -\Delta p = 0 & \text{in } \Omega, \\ p = 0 & \text{on } \Gamma_0, \\ p = \frac{\partial u}{\partial \mathbf{n}} - Q & \text{on } \Gamma_f. \end{cases} \quad (2.7)$$

Notice that the only difference between the primal and adjoint boundary value problems concerns the Dirichlet datum on Γ_f . The normal flux of the primal solution serves as the Dirichlet datum for the adjoint problem.

3 Boundary element discretisation

During the iterative shape optimisation the primal and the adjoint boundary value problems (2.1) and (2.7), respectively, have to be solved to compute the cost functional (2.3) and the steepest descent direction (2.5). The actual domain boundaries are described with multiresolution subdivision surfaces as will be introduced in Section 4. For all the computations we apply the boundary element method to avoid remeshing or smoothing of the volume mesh in each step of the optimisation process.

We apply a direct boundary integral formulation, as we need the normal fluxes $\frac{\partial p}{\partial \mathbf{n}}$ and $\frac{\partial u}{\partial \mathbf{n}}$ for the evaluation of the kernel function g in (2.6). For details on boundary integral equations, the properties of the involved boundary integral operators and boundary element methods, see, e.g., [4, 5, 6, 7, 8].

The solution of the original boundary value problem (2.1) is given by the representation formula

$$u(\tilde{\mathbf{x}}) = \int_{\Gamma} U^*(\tilde{\mathbf{x}}, \mathbf{y}) t(\mathbf{y}) \, d\mathbf{s}_{\mathbf{y}} - \int_{\Gamma} \frac{\partial}{\partial \mathbf{n}_{\mathbf{y}}} U^*(\tilde{\mathbf{x}}, \mathbf{y}) u(\mathbf{y}) \, d\mathbf{s}_{\mathbf{y}} \quad \text{for } \tilde{\mathbf{x}} \in \Omega \quad (3.1)$$

where $\Gamma = \partial\Omega = \Gamma_{\text{f}} \cup \Gamma_0$ is the boundary of the domain Ω and $t := \frac{\partial u}{\partial \mathbf{n}}$ denotes the normal flux. The fundamental solution $U^*(\mathbf{x}, \mathbf{y})$ of the Laplacian in three dimensions reads

$$U^*(\mathbf{x}, \mathbf{y}) := \frac{1}{4\pi} \frac{1}{|\mathbf{x} - \mathbf{y}|}.$$

The limiting case $\Omega \ni \tilde{\mathbf{x}} \rightarrow \mathbf{x} \in \Gamma$ provides the boundary integral equation

$$u(\mathbf{x}) = (Vt)(\mathbf{x}) + \frac{1}{2}u(\mathbf{x}) - (Ku)(\mathbf{x}) \quad \text{for almost all } \mathbf{x} \in \Gamma, \quad (3.2)$$

where $V: H^{-1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$ denotes the single layer boundary integral operator

$$(Vt)(\mathbf{x}) := \int_{\Gamma} U^*(\mathbf{x}, \mathbf{y}) t(\mathbf{y}) \, d\mathbf{s}_{\mathbf{y}} \quad \text{for } \mathbf{x} \in \Gamma,$$

and $K: H^{1/2}(\Gamma) \rightarrow H^{1/2}(\Gamma)$ denotes the double layer boundary integral operator

$$(Ku)(\mathbf{x}) := \int_{\Gamma} \frac{\partial}{\partial \mathbf{n}_{\mathbf{y}}} U^*(\mathbf{x}, \mathbf{y}) u(\mathbf{y}) \, d\mathbf{s}_{\mathbf{y}} \quad \text{for } \mathbf{x} \in \Gamma.$$

To determine the unknown normal fluxes $t := \frac{\partial u}{\partial \mathbf{n}}$ and $q := \frac{\partial p}{\partial \mathbf{n}}$ for the boundary value problems (2.1) and (2.7), respectively, we use the boundary integral equation (3.2) to obtain

$$(Vt)(\mathbf{x}) = \frac{1}{2}u(\mathbf{x}) + (Ku)(\mathbf{x}), \quad (3.3a)$$

$$(Vq)(\mathbf{x}) = \frac{1}{2}p(\mathbf{x}) + (Kp)(\mathbf{x}) \quad (3.3b)$$

for almost all $\mathbf{x} \in \Gamma$. For the original boundary value problem (2.1), the boundary integral equation (3.3a) can be simplified to

$$(Vt)(\mathbf{x}) = \begin{cases} 0 & \text{for } \mathbf{x} \in \Gamma_0, \\ 1 & \text{for } \mathbf{x} \in \Gamma_f \end{cases}$$

due to the kernel properties of the double layer boundary integral operator. As the boundary integral operators are bounded and the single layer boundary integral operator V is $H^{-1/2}(\Gamma)$ -elliptic, all boundary integral equations are uniquely solvable.

For the boundary element discretisation we use the Galerkin variational formulation based on piecewise constant approximations

$$t_h := \sum_{i=1}^N t_i \psi_i \quad \text{and} \quad q_h := \sum_{i=1}^N q_i \psi_i. \quad (3.4)$$

The related constant basis functions

$$\psi_i(\mathbf{x}) := \begin{cases} 1 & \text{for } \mathbf{x} \in \tau_i, \\ 0 & \text{elsewhere} \end{cases}$$

are defined with respect to a decomposition of the surface Γ into N planar triangular shape regular boundary elements τ_i . Thus, we have to solve the following two systems of linear equations

$$\mathbf{V}_h \mathbf{t} = \mathbf{M}_h \mathbf{f}, \quad (3.5a)$$

$$\mathbf{V}_h \mathbf{q} = \left(\frac{1}{2} \mathbf{M}_h + \mathbf{K}_h\right) \mathbf{h} \quad (3.5b)$$

to determine the normal flux vectors $\mathbf{t} \in \mathbb{R}^N$ and $\mathbf{q} \in \mathbb{R}^N$ with the coefficients t_i and q_i . The Galerkin matrices are given by

$$\begin{aligned} \mathbf{V}_h[j, i] &:= \int_{\tau_j} \int_{\tau_i} U^*(\mathbf{x}, \mathbf{y}) \, d\mathbf{s}_y \, d\mathbf{s}_x, & \mathbf{K}_h[j, i] &:= \int_{\tau_j} \int_{\tau_i} \frac{\partial}{\partial \mathbf{n}_y} U^*(\mathbf{x}, \mathbf{y}) \, d\mathbf{s}_y \, d\mathbf{s}_x, \\ \mathbf{M}_h[j, i] &:= \int_{\tau_j} \psi_i(\mathbf{x}) \, d\mathbf{s}_x \end{aligned}$$

for $i, j = 1, \dots, N$. The coefficients of the two vectors \mathbf{f} and \mathbf{h} on the right hand side of (3.5) are given by

$$f_i := \begin{cases} 0 & \text{for } \tau_i \in \Gamma_0, \\ 1 & \text{for } \tau_i \in \Gamma_f, \end{cases} \quad h_i := \begin{cases} 0 & \text{for } \tau_i \in \Gamma_0, \\ t_i - Q & \text{for } \tau_i \in \Gamma_f. \end{cases}$$

During shape optimisation we need the shape gradient at the mesh nodes \mathbf{x}_j , cf. Sections 4, 5. To this end, we approximate the kernel function g of the shape gradient (2.6) with

$$\tilde{g}_j := -\tilde{q}_j \tilde{t}_j - \frac{H(\mathbf{x}_j)}{2} \left((\tilde{t}_j)^2 - Q^2 \right) \quad (3.6)$$

in the nodes \mathbf{x}_j . We estimate the required additive curvature H with discrete differential operators given in [29]. Instead with the discontinuous normal fluxes in (3.4) we compute the shape gradients \tilde{g}_j with the nodally weighted averages

$$\tilde{t}_j := \frac{\sum_{i \in I(j)} t_i A_i}{\sum_{i \in I(j)} A_i} \quad \text{and} \quad \tilde{q}_j := \frac{\sum_{i \in I(j)} q_i A_i}{\sum_{i \in I(j)} A_i},$$

where the index set $I(j)$ contains all the element indices connected to node j and A_i denotes the surface area of the element τ_i . This can be interpreted as a quasi-interpolation, see, e.g., [30].

As the fundamental solution $U^*(\mathbf{x}, \mathbf{y})$ is non-local, the entries of the matrices \mathbf{V}_h and \mathbf{K}_h are non-zero in general. As a result, both are fully populated. Therefore, acceleration techniques, such as the fast multipole method [13], are crucial for the fast and efficient computation of matrix operations. A description of our implementation of the boundary element method using the fast multipole method and a detailed error analysis is given in [12].

4 Multiresolution subdivision surfaces

Next, we introduce the multiresolution subdivision surfaces for describing the geometry of the computational domain. Subdivision is a powerful geometric modelling technique for generating smooth surfaces on arbitrary connectivity meshes. The smooth surface is recursively generated through repeated refinement of an initial coarse control mesh. In this paper, we limit ourselves to the review of the elementary properties of subdivision surfaces. In particular, we only consider subdivision schemes that lead to cubic b-splines in the univariate setting and quartic box-splines in the bivariate setting with triangular elements. For more details we refer to [23, 25, 20, 31].

4.1 Subdivision refinement

To begin with, we consider the univariate subdivision refinement of polygons, as illustrated in Figure 2. The initial polygon is referred to as the *control polygon* and determines the shape of the *limit curve* resulting from repeated subdivision. It is instructive to think that each subdivision step consists of a refinement and an averaging step. In the refinement step each segment of the polygon is subdivided into two segments, see Figure 3a. Subsequently, the vertex coordinates of the refined polygon are determined by averaging the coarse vertex

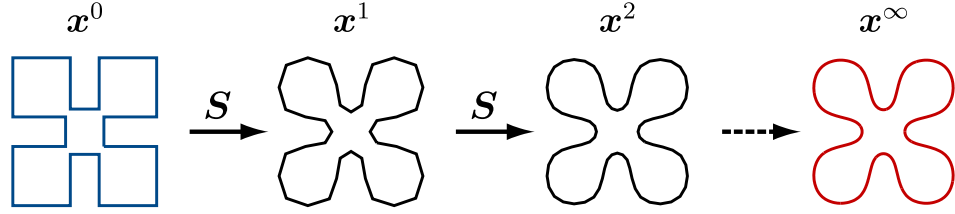


Figure 2: Subdivision refinement of a given control polygon (shown left). The three polygons to the right are generated by repeated subdivision. Notice the increasing smoothness of the subdivided polygons.

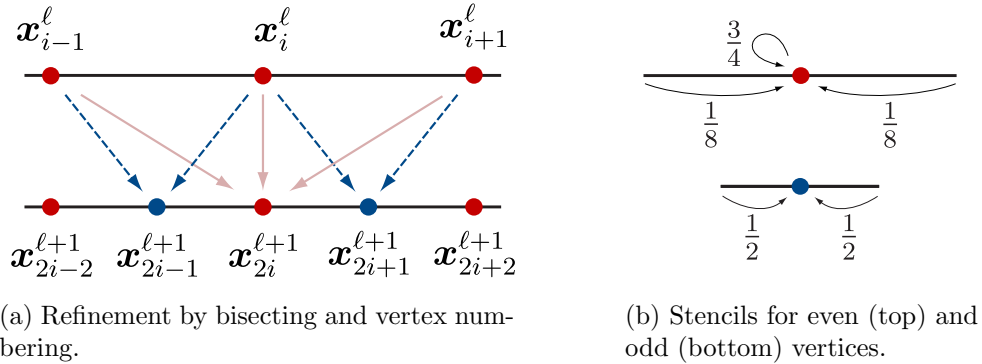


Figure 3: Univariate subdivision refinement.

coordinates with the two stencils shown in Figure 3b. The even vertex stencil applies to vertices that are already present in the coarse polygon and the odd vertex stencil applies to vertices that are only present in the refined polygon. The naming odd and even is motivated by the consecutive numbering of vertices in a polygon where newly inserted vertices receive odd numbers. According to Figure 3b, for a given coarse polygon of level ℓ with vertex coordinates \mathbf{x}_i^ℓ a refined polygon of level $\ell + 1$ with vertex coordinates $\mathbf{x}_i^{\ell+1}$ is computed with

$$\mathbf{x}_{2i}^{\ell+1} = \frac{1}{8}\mathbf{x}_{i-1}^\ell + \frac{3}{4}\mathbf{x}_i^\ell + \frac{1}{8}\mathbf{x}_{i+1}^\ell, \quad (4.1a)$$

$$\mathbf{x}_{2i+1}^{\ell+1} = \frac{1}{2}\mathbf{x}_i^\ell + \frac{1}{2}\mathbf{x}_{i+1}^\ell. \quad (4.1b)$$

By definition we assign the coarse control mesh the level $\ell = 0$. Without going into the details, we note that the weights in (4.1) have been chosen such that the limit curve for $\ell \rightarrow \infty$ is a uniform cubic b-spline. The weights can be modified in order to change the interpolation and/or smoothness properties of the curve. Being a cubic b-spline the limit curve is C^2 -continuous which can also be reduced to C^0 -continuous by modifying the stencil weights. For the sake of completeness we note that other choices of stencil weights may also

lead to smooth curves [23].

It is convenient to express the subdivision equations (4.1) as a matrix vector multiplication

$$\mathbf{x}^{\ell+1} = \mathbf{S}\mathbf{x}^{\ell}, \quad (4.2)$$

where $\mathbf{x}^{\ell+1}$ and \mathbf{x}^{ℓ} are the vectors of all vertex coordinates of the coarse and refined polygons, respectively, and \mathbf{S} is the subdivision matrix. The subdivision matrix \mathbf{S} is a banded sparse matrix and its entries are the stencil weights given in Figure 3b. The dimensions of \mathbf{S} , for instance, for the 2D example in Figure 2 are $(4N \times 2N)$, where N is the number of vertices on level ℓ with each having two coordinates. Although the dimensions of \mathbf{S} increase with increasing ℓ we denote all subdivision matrices with \mathbf{S} since the same subdivision stencils are used in each step.

In the bivariate surface setting, we use the subdivision scheme introduced by Loop [22]. The notion of control polygon is now replaced with *control mesh* to reflect the higher dimension. The Loop scheme is based on triangular meshes and yields quartic box-splines for meshes with regular vertices. In this context a vertex is regular when it is inside the domain and incident to six edges, or is on the boundary of the domain and incident to four edges. The number of incident edges to a vertex is usually referred to as the *valence* of that vertex. It is known that the Loop scheme yields a C^1 -continuous surface on non-regular vertices and a C^2 -continuous surface everywhere else.

In the refinement step of the Loop scheme, each triangle of the control mesh is subdivided into four triangles by introducing new vertices at the edge midpoints, as seen in Figure 4a. Subsequently, the vertex coordinates of the refined mesh are computed with the stencils shown in Figure 4b. Similar to univariate subdivision there are two different types of stencils. The even vertex stencil applies to vertices that already existed on the coarse mesh and the odd vertex stencil applies to newly introduced vertices on the edges. Notice that the weights in the stencil for even vertices depend on the valence v of the vertex.

In the present work, we use the extended Loop subdivision scheme introduced by Biermann et al. [32]. In contrast to the original Loop scheme, the extended scheme allows the modelling of non-smooth and non-manifold features, such as creases or T-sections, which are crucial for many engineering applications. It is clear that the limit surface along creases is only C^0 -continuous. Figure 5 shows a mechanical connector geometry containing sharp features described with the extended subdivision surfaces. For implementation details we refer to [31].

4.2 Multiresolution subdivision surfaces

Subdivision surfaces represent a limit surface with a nested hierarchy of control meshes of increasing resolution. As known in computer graphics, this property lends itself to efficient multiresolution editing of surfaces [26, 27]. The basic idea in multiresolution editing is to modify coarse mesh vertex coordinates to perform large-scale changes (to the limit surface) and to modify fine mesh coordinates to add localised changes. By way of example, this is

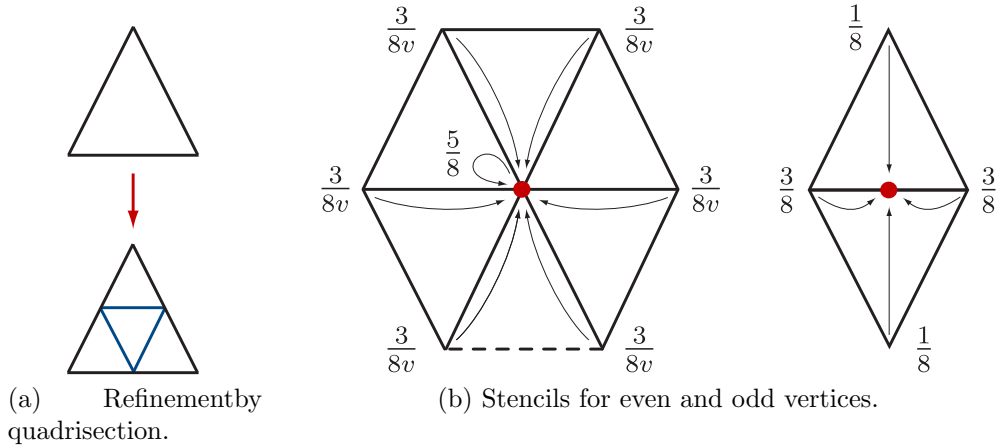


Figure 4: Bivariate Loop subdivision refinement.

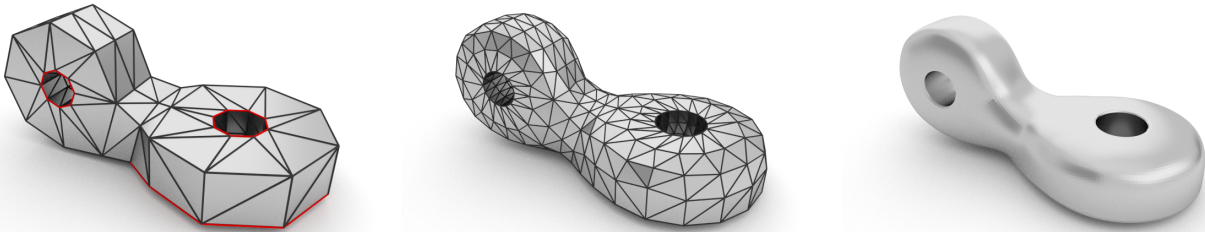


Figure 5: Subdivision refinement of a connector. On the coarse control mesh on the left the edges shown in red are tagged as crease edges. The control mesh in the centre is generated with extended Loop subdivision. The geometry on the right is a rendering of the limit surface.

illustrated in Figure 6 for the connector geometry previously introduced in Figure 5. First the control mesh coordinates \mathbf{x}^0 are modified with $\mathbf{x}^0 + \mathbf{d}^0$, where \mathbf{d}^0 can be thought as a user given perturbation vector. In the considered example, \mathbf{d}^0 applies displacements only to the vertices placed on one of the hole edges. Subsequent computation of the limit surface (by repeated subdivision) leads to a geometry with rather large scale changes. Alternatively, the edge of the hole can be displaced on level $\ell = 1$, i.e., $\mathbf{x}^1 + \mathbf{d}^1 = \mathbf{S}\mathbf{x}^0 + \mathbf{d}^1$. This results in more localised changes. It can be shown that the area of influence for each vertex extends over two rings of adjacent triangles.

The multiresolution editing algorithms available in computer graphics allow us to alternately edit coarse and fine resolutions, see, e.g., [26]. This is achieved by a wavelet-like decomposition of the geometry into a low resolution part and a collection of wavelet coefficients expressing perturbations from the low resolution part [26, 27]. To compute such a decomposition it is necessary to define, in addition to the subdivision refinement, a coarsening

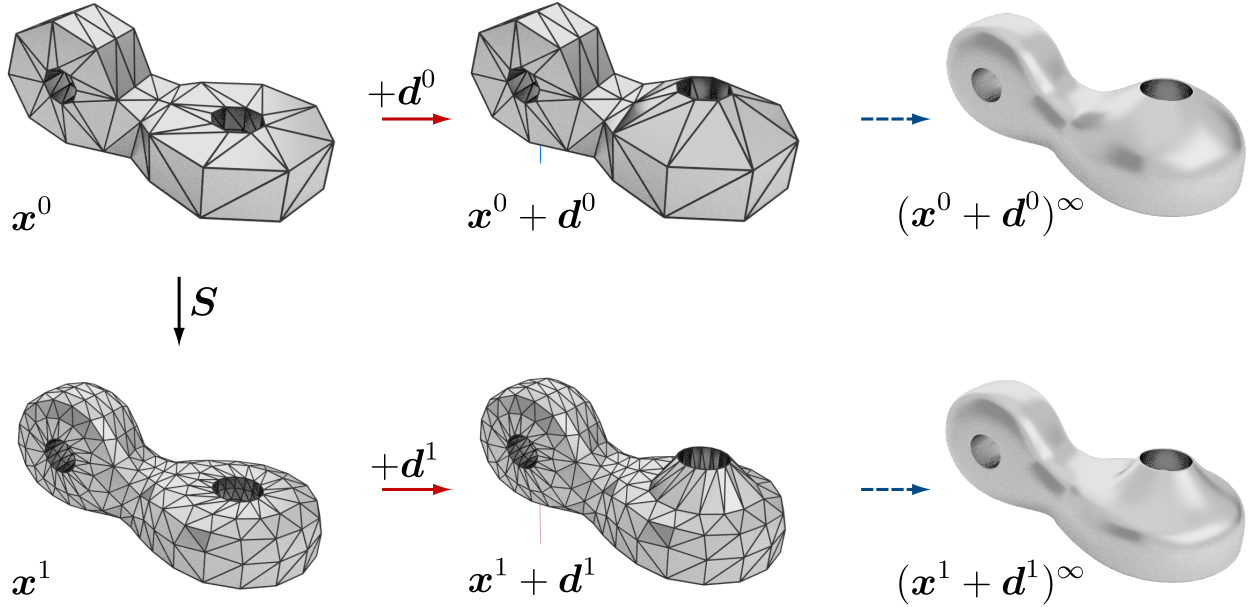


Figure 6: Multiresolution editing of the connector geometry introduced in Figure 5. The geometry is modified by moving the edge of one of the holes in the vertical direction. On the first row the modification is performed on level $\ell = 0$ and on the second row it is performed on level $\ell = 1$. Notice the effect of the modification level on the limit surface (last column).

operation

$$\mathbf{x}^\ell = \mathbf{R}\mathbf{x}^{\ell+1}. \quad (4.3)$$

The coarsening matrix \mathbf{R} enables the computation of the coarse representation \mathbf{x}^ℓ corresponding to a given edited fine representation $\mathbf{x}^{\ell+1}$. In contrast, recall that the subdivision matrix \mathbf{S} enables to compute for a given coarse representation the corresponding fine representation, cf. (4.2). Different choices for the matrix \mathbf{R} are possible. For instance, it can be determined by the least squares fitting

$$\mathbf{x}^\ell = \operatorname{argmin}_{\mathbf{y}^\ell} \|\mathbf{x}^{\ell+1} - \mathbf{S}\mathbf{y}^\ell\|^2, \quad (4.4)$$

which leads to

$$\mathbf{S}^\top \mathbf{S} \mathbf{x}^\ell = \mathbf{S}^\top \mathbf{x}^{\ell+1}. \quad (4.5)$$

By comparison with (4.3) we observe that the coarsening matrix has to be $\mathbf{R} = (\mathbf{S}^\top \mathbf{S})^{-1} \mathbf{S}^\top$. In Figure 7 the functioning of the coarsening matrix \mathbf{R} is illustrated. We reconsider the one-dimensional subdivision refinement example previously introduced in Figure 2 and investigate the coarsening of two limit curves. On the top row of Figure 7, the limit curve previously obtained via subdivision refinement in Figure 2 is successively coarsened until

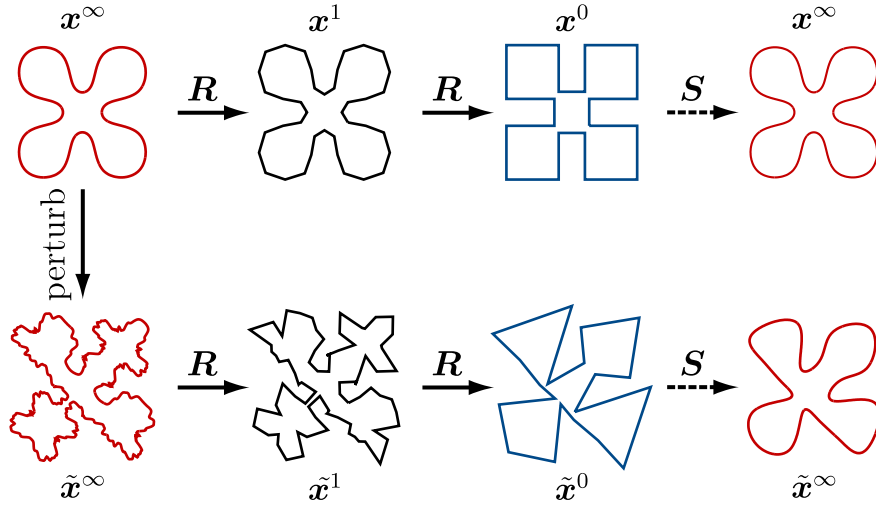


Figure 7: Coarsening of the limit curve previously obtained with subdivision refinement, see Figure 2. Top row shows the coarsening of the unperturbed limit surface. For a vertex with index i the perturbation is of the form $\tilde{\mathbf{x}}_i^\infty = \mathbf{x}_i^\infty + \mathbf{n}(\mathbf{x}_i)a \sin(f\langle \mathbf{x}_i, \mathbf{e}_0 \rangle)$, where \mathbf{n} is the normal to the curve, \mathbf{e}_0 is the basis vector $(1, 0, 0)^\top$, a is the amplitude, and f is the frequency.

the original control polygon is recovered. On the bottom row the coarsening of a perturbed limit polygon is shown. As can be seen, the coarsening operation successively removes the high-frequency oscillations from the geometry. The resulting control polygon represents a limit curve which is a visually faithful smooth representation of the perturbed original curve. In multiresolution analysis of surfaces the refinement and coarsening matrices \mathbf{S} and \mathbf{R} , respectively, are used to construct a wavelet-like decomposition of the geometry. The resulting algorithms allow the concurrent editing of coarse and fine levels. As will be discussed in Section 5, in multiresolution shape optimisation it is sufficient to start from a coarse control mesh and successively add details to a surface by editing increasingly finer subdivided meshes. This can be achieved without a wavelet-like decomposition of the surface.

5 Multiresolution shape optimisation

We are now in a position to discuss the multiresolution optimisation of the electrostatic problem introduced in Section 2 by combining subdivision surfaces with the shape gradients computed with BEM. The basic idea in our approach is to use a fine mesh for computing the shape gradients and to use a coarser mesh for geometry modification [20, 33]. The underlying subdivision representation ensures that the same limit surface is always considered, independently of the actual refinement level. Moreover, it is straightforward to transfer

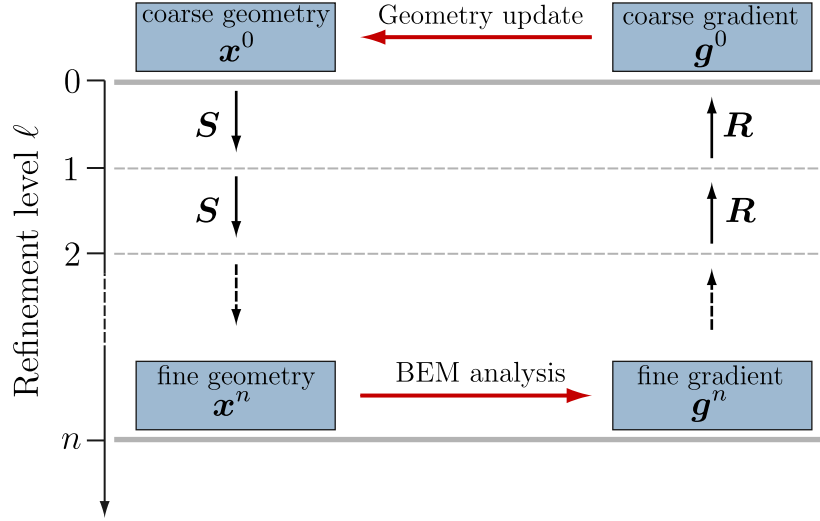


Figure 8: Multiresolution optimisation algorithm.

any geometry or field data between the different levels using the introduced subdivision refinement and coarsening operations, see (4.2) and (4.3), respectively.

In the following we briefly summarise our multiresolution optimisation algorithm. This description should be read in conjunction with Figure 8. For simplicity, we assume that the discretised optimisation problem is solved using the steepest descent algorithm and no constraints are present.

1. Initialise the optimisation level with $\ell_o = 0$, the computational level with $\ell_c = n$ and the discretised cost functional with $J(\Omega^{\ell_c}, u^{\ell_c}) = \infty$. Here, n is user given and has to be large enough such that the accuracy of the numerical solution is sufficient for practical purposes.
2. Repeatedly subdivide the optimisation mesh at level ℓ_o until the computation level ℓ_c is reached.
3. Solve the primal boundary element system (3.5a) and the adjoint boundary element system (3.5b) using the computational mesh with $\ell_c = n$.
4. Evaluate the cost functional $J(\Omega^{\ell_c}, u^{\ell_c})$, i.e., the discrete counterpart to (2.3), and the shape derivative kernel $\tilde{g}_i^{\ell_c}$ at each vertex \mathbf{x}_i according to (3.6).
5. If the cost functional $J(\Omega^{\ell_c}, u^{\ell_c})$ is increasing in relation to the previous value, increment the optimisation level $\ell_o \leftarrow (\ell_o + 1)$.
6. If the optimisation level has reached the computational level, i.e., $\ell_o = \ell_c$, terminate the optimisation procedure.

7. Repeatedly coarsen the computational mesh and the associated vertex shape derivative kernel \tilde{g} until the optimisation level ℓ_o together with \tilde{g}^{ℓ_o} is reached.
8. Perturb the vertex coordinates of the optimisation mesh according to (2.5) with

$$\mathbf{x}_i^{\ell_o} \leftarrow (\mathbf{x}_i^{\ell_o} - \alpha \tilde{g}_i^{\ell_o} \mathbf{n}(\mathbf{x}_i^{\ell_o}))$$

where the index i denotes vertex id, the kernel values $\tilde{g}_i^{\ell_o}$ are computed with (3.6), and $\alpha \geq 0$ is a suitably chosen step length parameter.

9. Go to step 2.

In our actual implementation we use the method of moving asymptotes (MMA) proposed by Svanberg [34, 35] as implemented in the NLopt library [36]. This changes in particular the Step 8 in the above algorithm. Using a more sophisticated optimisation algorithm than steepest descent significantly reduces the number of optimisation iterations. For further details, such as the consideration of constraints, we refer to [33].

6 Examples

We present three examples to demonstrate the functioning of the proposed BEM based multiresolution shape optimisation approach. In all examples, the Dirichlet boundary value problem for the Laplace equation (2.1) is considered. Recall that the domain boundary Γ is split into the two disjoint parts Γ_0 and Γ_f , with the prescribed potentials $u = 0$ on Γ_0 and $u = 1$ on Γ_f . During the optimisation only the shape of the boundary Γ_f is updated with multiresolution shape optimisation described in Section 5. The boundary Γ_0 and its mesh resolution remains unchanged. As mentioned, the discretised optimisation problem is solved with the MMA algorithm as implemented in the NLopt library [36]. The input to the NLopt library consists of the cost functional $J(\Omega^{\ell_c}, u^{\ell_c})$ and for each vertex on the optimisation level ℓ_o its position $\mathbf{x}_i^{\ell_o}$ and gradient $\tilde{g}_i^{\ell_o} \mathbf{n}(\mathbf{x}_i^{\ell_o})$.

6.1 Box in a sphere

As an introductory example we optimise the shape of a box placed inside a sphere, see Figure 9, with the expected normal flux density Q in (2.3) set to 20. It can be shown that the optimal shape for the inner box is a sphere with half the diameter of the outer sphere [37]. The box, representing the to be optimised boundary Γ_f , is of size $0.16 \times 0.2 \times 0.24$ and the outer sphere, representing the fixed boundary Γ_0 , has radius 0.2. The coarse mesh for the box contains 48 elements which increases to 768 elements in the twice subdivided fine mesh at level $\ell_c = 2$. During the subdivision refinement, the creases in the coarse mesh are maintained as creases using the extended subdivision stencils mentioned in Section 4.1, see

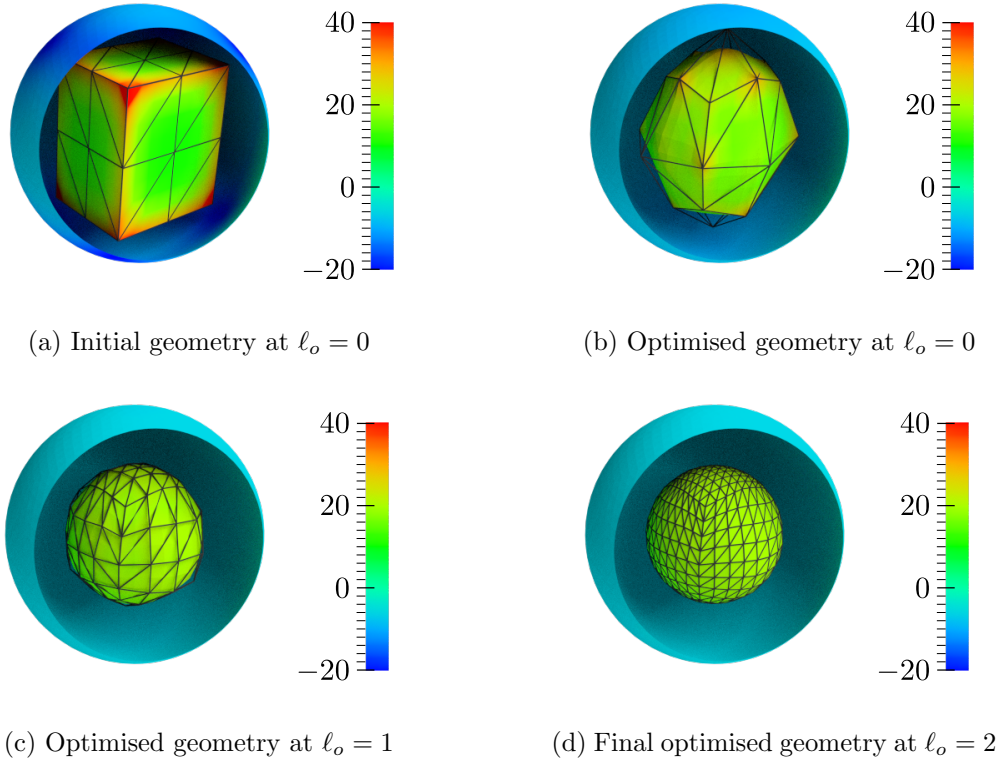


Figure 9: Box in a sphere. Initial and optimised geometries with isocontours of the normal flux. The meshes indicate the optimisation level ℓ_o . The isocontours belong to the fine computational mesh at level $\ell_c = 2$. The geometries shown in (b) and (c) represent intermediate results and (d) represents the final result.

also [32]. Note that on the limit surface the creases are only C^0 -continuous and not at least C^1 -continuous. The resolution of the outer sphere remains fixed with 3840 elements. Hence, the meshes for the boundary element analysis of the cube and sphere consist of 768 and 3840 elements, respectively.

Figure 9a shows the initial coarse geometry yielding a cost functional value of $J(\Omega^{\ell_c}, u^{\ell_c}) = 15.38$. First we select this coarse geometry as optimisation level, i.e., $\ell_o = 0$, and obtain the optimised geometry shown in Figure 9b. After consecutively selecting $\ell_o = 1$ and $\ell_o = 2$ and optimising we obtain the final optimised geometry shown in Figure 9d. This final shape of the initial box is nearly a sphere of diameter 0.215 and the cost functional value is $J(\Omega^{\ell_c}, u^{\ell_c}) = 8.46 \cdot 10^{-3}$, which represents a reduction of 99.95%. The maximum normal flux in the initial geometry is $J_{\max}(\Omega^{\ell_c}, u^{\ell_c}) = 81.49$ and reduces to $J_{\max}(\Omega^{\ell_c}, u^{\ell_c}) = 21.43$ in the final geometry. As to be expected, the optimisation leads to a geometry with nearly uniform distribution of normal flux as seen in Figure 9d. In Figure 10 a rendering of the limit surface of the final optimised geometry is shown. Notice the high smoothness of the

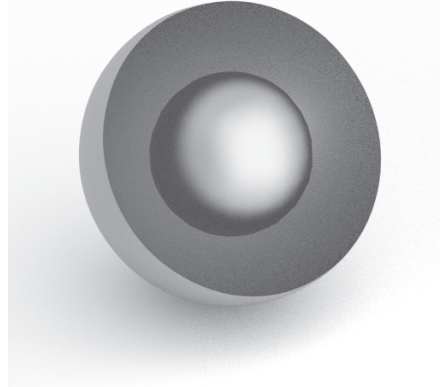


Figure 10: Box in a sphere. Rendering of the limit surface of the final optimised shape.

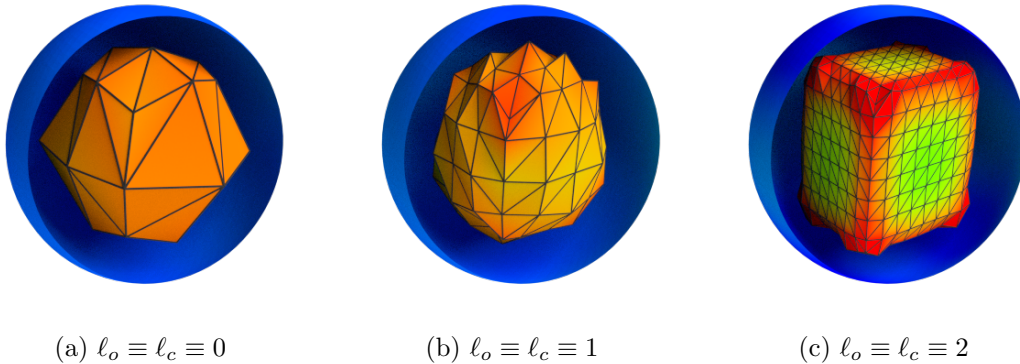


Figure 11: Box in a sphere. Optimisation without multiresolution using the same level for shape control and computation, i.e., $\ell_o \equiv \ell_c$. The meshes indicate the level $\ell_o \equiv \ell_c$ and the isocontours indicate the normal flux.

optimised geometry.

In order to demonstrate the robustness and benefits of the proposed multiresolution optimisation approach we computed this example using the same level for optimisation and computation, i.e., $\ell_o \equiv \ell_c$. Figure 11 shows the geometries obtained with $\ell_o \equiv \ell_c \equiv 0$, $\ell_o \equiv \ell_c \equiv 1$ and $\ell_o \equiv \ell_c \equiv 2$. Notice in Figure 11 the stark differences in the three obtained geometries and the non-physical geometry oscillations for the finer meshes. Moreover, the result obtained with $\ell_o \equiv \ell_c \equiv 0$ is highly questionable because of the coarseness of the computational mesh. The non-physical oscillations obtained in Figure 11b are reminiscent of problems reported in the finite element context [17, 18].

6.2 L-shaped domain

As a second example we consider the optimisation of an L-shaped domain placed inside a larger L-shaped domain, see Figure 12. The geometry contains convex and non-convex corners which usually represent challenges for geometry updating during optimisation. The inner L-shaped boundary is the boundary Γ_f to be optimised and the outer L-shaped boundary is the fixed boundary Γ_0 . The number of elements on Γ_f increases from initially 28 to 448 in the twice subdivided mesh at level $\ell_c = 2$. The resolution of the boundary Γ_0 remains fixed with 7168 elements. During the subdivision refinement of the inner L-shaped boundary Γ_f the creases are not retained as visible in Figure 13a. As previously mentioned, in order to maintain the creases it is necessary to use special subdivision stencils [32, 31].

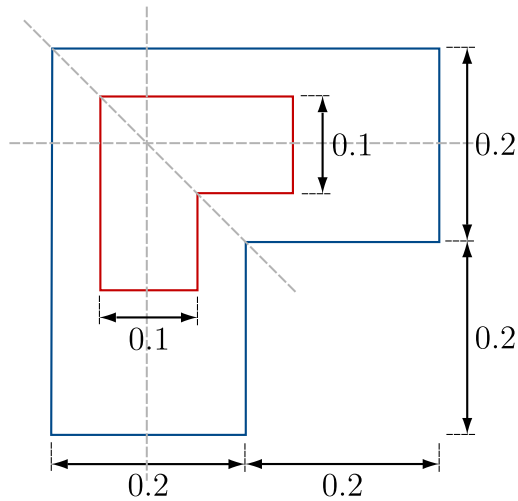


Figure 12: L-shaped domain. Plan view of the initial geometry with the outer polygon representing the boundary Γ_0 and the inner polygon representing the boundary Γ_f to be optimised. The out-of-plane length of each part is 0.1 and 0.2, respectively.

Figure 13a shows the initial geometry with the maximum normal flux $J_{\max}(\Omega^{\ell_c}, u^{\ell_c}) = 62.04$ and the cost functional $J(\Omega^{\ell_c}, u^{\ell_c}) = 1.32$ (with the expected value Q set to 30). Notice the relatively low value for $J_{\max}(\Omega^{\ell_c}, u^{\ell_c})$ resulting from the smoothness of the computational geometry at level $\ell_c = 2$. The ellipsoidal geometry resulting from the multiresolution shape optimisation is shown in Figure 13d. As can be deduced from the isocontour plots on the optimised boundary, the normal flux is nearly uniformly distributed. Moreover, during optimisation $J_{\max}(\Omega^{\ell_c}, u^{\ell_c})$ reduces by 48.69% to the value of 31.83. The cost functional $J(\Omega^{\ell_c}, u^{\ell_c})$ reduces to $8.65 \cdot 10^{-3}$ (99.35% reduction).

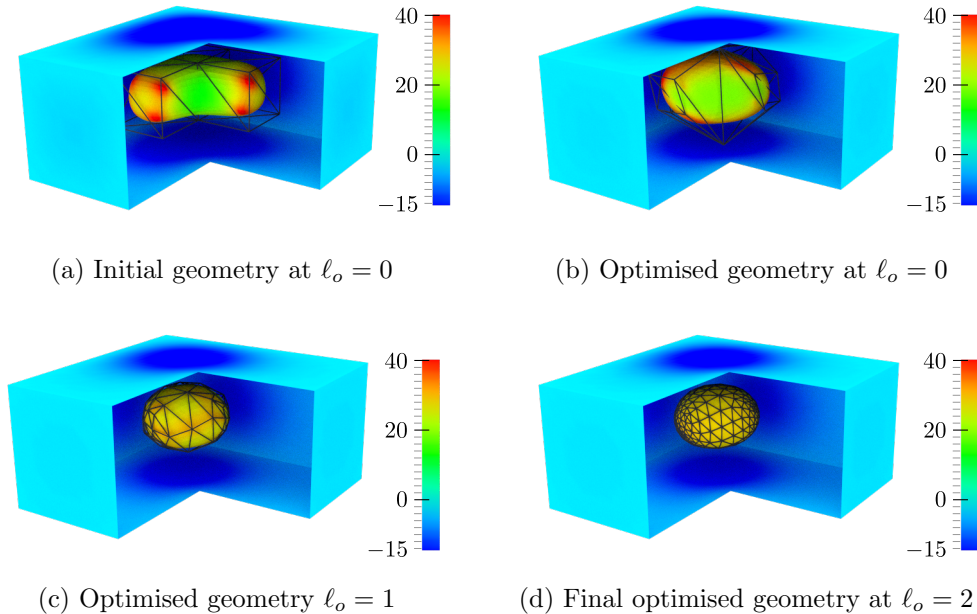


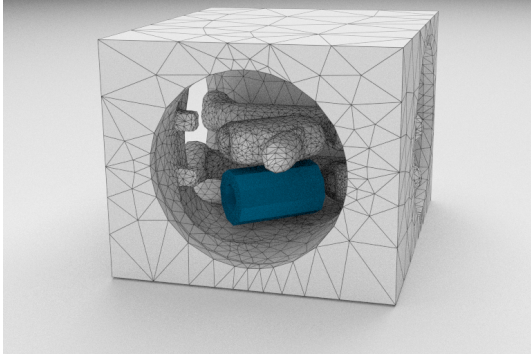
Figure 13: L-shaped domain. Initial and optimised geometries with isocontours of the normal flux. The meshes indicate the optimisation level ℓ_o and the isocontours belong to the fine computational mesh at level $\ell_c = 2$. The geometries shown in (b) and (c) represent intermediate results and (d) represents the final result.

6.3 Gas insulated switchgear

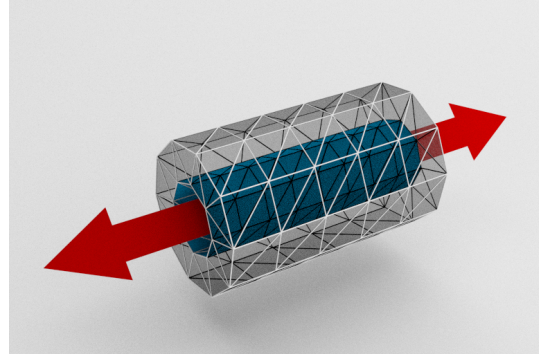
In this example we apply the proposed shape optimisation approach to the design of an electrode in a gas insulated switchgear component, see Figure 14. Such devices are widely used as disconnectors in high-voltage power transmission. The objective of shape optimisation is to reduce the propensity of the component for electric breakdown with the ultimate aim to enable more compact component geometries. This can be achieved by modifying the electrode geometries such that the maximum normal flux $J_{\max}(\Omega^{\ell_c}, u^{\ell_c})$ is minimised. We attempt to achieve this by minimising the cost functional $J(\Omega^{\ell_c}, u^{\ell_c})$.

In Figure 14 the gas insulated switchgear component is shown with the electrode in the form of a primitive cylinder. The cylinder represents the electrode geometry Γ_f to be optimised. The initial coarse mesh of the cylinder contains 264 elements. The creases on the cylinder are not tagged. Therefore, the geometry becomes smoother while it is refined by subdivision. As a design constraint, the inner surface of the cylinder is required to have a constant radius for a bolt passing through it. Geometric bounds on the positions of vertices lying on the inner surface are applied to prevent any radial movement that would violate this design requirement, see Figure 14b.

The once subdivision refined mesh with 1056 elements is chosen as the computational level,

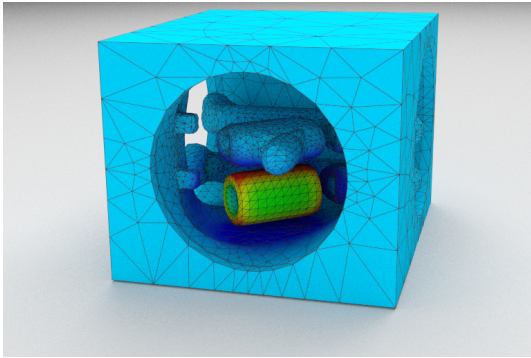


(a) The cylindrical electrode boundary Γ_f to be optimised is shown in dark blue and the other surfaces representing Γ_0 are shown in light grey.

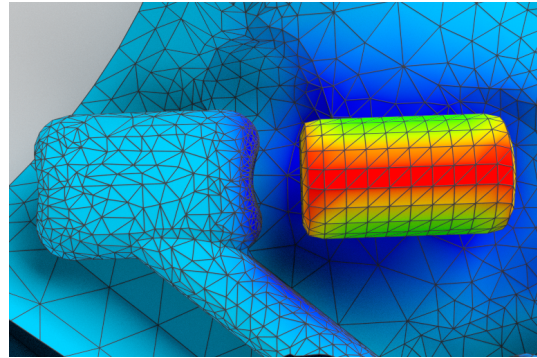


(b) Close-up of the cylindrical electrode. Vertices on the inner surface of the cylinder shown in dark blue are only allowed to move along the axis of the cylinder.

Figure 14: Gas insulated switchgear. Initial geometry and geometric optimisation constraints.



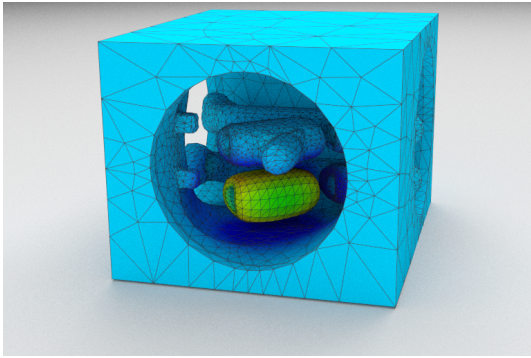
(a) Overall component.



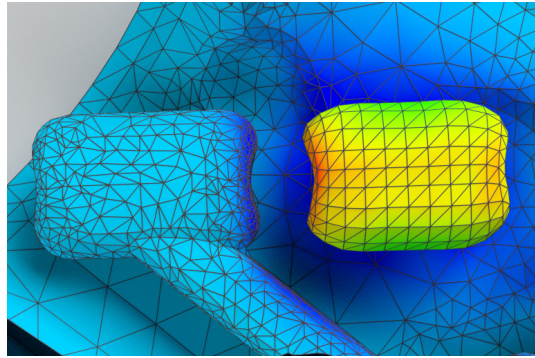
(b) Close-up of the electrode.

Figure 15: Gas insulated switchgear. Isocontours of the normal flux for the initial cylindrical electrode design.

i.e., $\ell_c = 1$. As can be seen in Figure 15, the ends of the cylinder become smoother because the usual (vs. extended) subdivision stencils are applied throughout the mesh. In this example, we consider the geometry at level $\ell_o = 0$ for optimisation. In the initial design, Figure 15, the maximum normal flux is $J_{\max}(\Omega^{\ell_c}, u^{\ell_c}) = 81.63$ before optimisation and reduces to $J_{\max}(\Omega^{\ell_c}, u^{\ell_c}) = 66.99$ in the optimised shape shown in Figure 16, corresponding to a reduction of 17.94%. However, the reduction in the cost function $J(\Omega^{\ell_c}, u^{\ell_c})$ is much higher with 38.24%. In Figure 17 we also show the component with the electrode geometry as currently manufactured by ABB. This electrode geometry has been obtained over the years by combining engineering intuition with simple calculations and testing. The similarities between the methodically shape optimised and the electrode geometry in production are

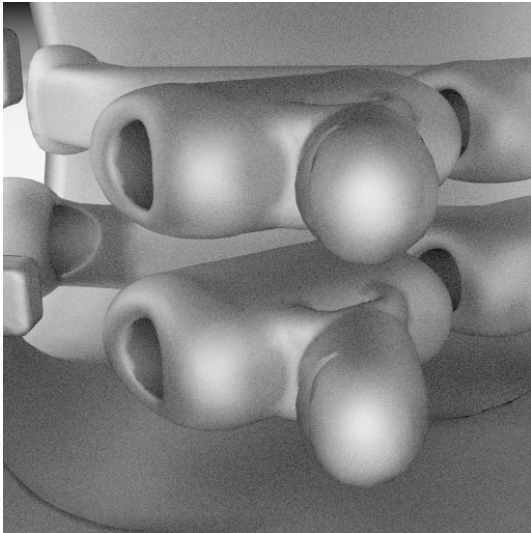


(a) Overall component.

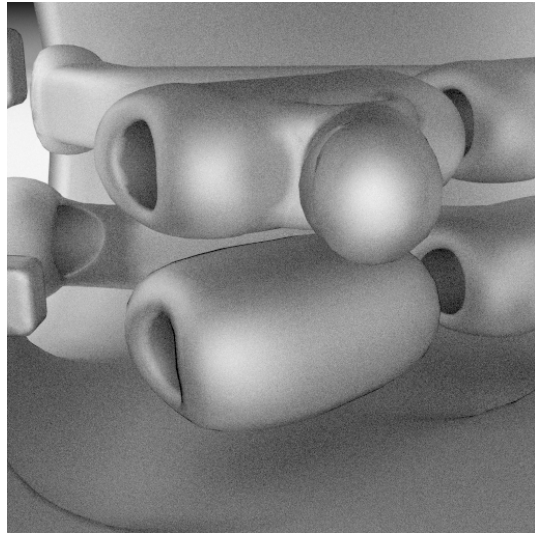


(b) Close-up of the electrode.

Figure 16: Gas insulated switchgear. Isocontours of the normal flux for the optimised electrode design.



(a) Manually optimised electrode.



(b) Limit surface of the multiresolution optimised electrode.

Figure 17: Gas insulated switchgear. Comparison of multiresolution shape optimised design with manually optimised electrode created by ABB engineers.

striking. Notice in particular the saddle shape at the two ends of the original cylinder. It helps to lower the large normal flux at the sharp crease at the boundary of the inner hole.

7 Summary and conclusions

We have introduced a novel multiresolution shape optimisation approach for electrostatic problems by combining fast boundary element methods with subdivision surfaces. For the considered electrostatic problems only a surface mesh is required for the boundary element discretisation. This is particularly appealing for three-dimensional problems because of the difficulties in generating and updating volume meshes. It was critical for the successful computation of the presented optimisation examples that there were no domain meshes.

For the description of the domain boundaries we used the subdivision surfaces. The inherent hierarchy of the subdivision surfaces allows to consider the same surface at different resolutions and to take advantage of multiresolution editing techniques. Starting from the coarsest control mesh increasingly finer meshes are used for geometry updating and a fine control mesh is used for the boundary element discretisation. As demonstrated with the computed examples, this effectively inhibits the non-physical geometry oscillations that may occur in shape optimisation. Moreover, any pathological element distortions on the computational mesh are practically avoided. As a result, there is no need to regenerate or smooth the surface mesh during the optimisation.

In the presented approach subdivision surfaces were used only for geometry parameterisation. It is, however, possible to use subdivision surfaces, or the underlying b-spline basis functions, for both analysis and geometry description. This has already been successfully demonstrated in isogeometric analysis of solids and shells with the finite element method, see, e.g., [38, 25]. Recently, the isogeometric analysis of solids with the boundary element method has also been explored [39]. In closing, we note that the introduced shape optimisation technique is easily extendible to other boundary value problems with known fundamental solutions, like elasticity, Stokes and Helmholtz. Moreover, due to the similarities between shape optimisation and inverse problems, see e.g. [40, 41], the present technique is also promising for inverse boundary problems.

Acknowledgement

We gratefully acknowledge the support provided through the FP7 Marie Curie IAPP project CASOPT (www.casopt.com). Jan Zapletal thanks for the support provided by the European Regional Development Fund in the IT4Innovations Centre of Excellence project (CZ.1.05/1.1.00/02.0070) and by the project SPOMECH - Creating a multidisciplinary R&D team for reliable solution of mechanical problems, reg. no. CZ.1.07/2.3.00/20.0070 within the Operational Programme ‘Education for competitiveness’ funded by the Structural Funds of the European Union and the state budget of the Czech Republic. Special thanks to Andreas Blaszyk from the ABB Corporate Research Center Switzerland for fruitful discussions and for providing the industrial applications.

References

- [1] E. Kuffel, W. Zaengl, J. Kuffel, High Voltage Engineering: Fundamentals, Butterworth-Heinemann, 2nd edition edition, 2000.
- [2] N. de Kock, M. Mendik, Z. Andjelic, A. Blasczyk, Application of the 3d boundary element method in the design of ehv gis components, *Electrical Insulation Magazine* 14 (1998).
- [3] D. Griffiths, Introduction to Electrodynamics, Pearson, 1998.
- [4] W. McLean, Strongly Elliptic Systems and Boundary Integral Equations, Cambridge University Press, Cambridge, 2000.
- [5] S. A. Sauter, C. Schwab, Boundary Element Methods, volume 39 of *Springer Series in Computational Mathematics*, Springer, Berlin, Heidelberg, 2011.
- [6] G. C. Hsiao, W. L. Wendland, Boundary integral equations, volume 164 of *Applied Mathematical Sciences*, Springer, Berlin, 2008.
- [7] O. Steinbach, Numerical approximation methods for elliptic boundary value problems. Finite and boundary elements, Springer, New York, 2008.
- [8] S. Rjasanow, O. Steinbach, The fast solution of boundary integral equations, *Mathematical and Analytical Techniques with Applications to Engineering*, Springer, New York, 2007.
- [9] M. C. Delfour, J.-P. Zolésio, Shapes and Geometries: Metrics, Analysis, Differential Calculus, and Optimization, SIAM, Philadelphia, second edition, 2011.
- [10] J. Haslinger, R. A. E. Mäkinen, Introduction to Shape Optimization: Theory, Approximation, and Computation, SIAM, 2003.
- [11] J. Sokolowski, J.-P. Zolésio, Introduction to Shape Optimization, Springer, Berlin, 1983.
- [12] G. Of, O. Steinbach, W. L. Wendland, The fast multipole method for the symmetric boundary integral formulation, *IMA Journal of Numerical Analysis* 26 (2006) 272–296.
- [13] L. Greengard, V. Rokhlin, A fast algorithm for particle simulations, *Journal of Computational Physics* 73 (1987) 325–348.
- [14] M. Bebendorf, S. Hardesty, Adaptive cross approximation of tensors arising in the discretization of boundary integral operator shape derivatives, *Engineering Analysis with Boundary Elements* 37 (2013) 60–67.

- [15] K. Eppler, H. Harbrecht, Fast wavelet BEM for 3d electromagnetic shaping, *Applied Numerical Mathematics* 54 (2005) 537–554.
- [16] K. Eppler, H. Harbrecht, Tracking Neumann data for stationary free boundary problems, *SIAM Journal on Control and Optimization* 48 (2009) 2901–2916.
- [17] V. Braibant, C. Fleury, Shape optimal design using b-splines, *Computer Methods in Applied Mechanics and Engineering* 44 (1984) 247–267.
- [18] R. T. Haftka, R. V. Grandhi, Structural shape optimization — a survey, *Computer Methods in Applied Mechanics and Engineering* 57 (1986) 91–106.
- [19] K.-U. Bletzinger, S. Kimmich, E. Ramm, Efficient modeling in shape optimal design, *Computing Systems in Engineering* 2 (1991) 483–495.
- [20] F. Cirak, M. Scott, E. Antonsson, M. Ortiz, P. Schröder, Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision, *Computer Aided Design* 34 (2002) 137–148.
- [21] W. Wall, M. Frenzel, C. Cyron, Isogeometric structural shape optimisation, *Computer Methods in Applied Mechanics and Engineering* 197 (2008) 2976–2988.
- [22] C. Loop, Smooth Subdivision Surfaces Based on Triangles, Master’s thesis, Department of Mathematics, University of Utah, 1987.
- [23] D. Zorin, P. Schröder, Subdivision for modeling and animation, *SIGGRAPH 2000 Course Notes*, 2000.
- [24] J. Stam, Evaluation of Loop subdivision surfaces, in: *SIGGRAPH 1999 Course Notes*, Los Angeles, CA.
- [25] F. Cirak, M. Ortiz, P. Schröder, Subdivision surfaces: A new paradigm for thin-shell finite-element analysis, *International Journal for Numerical Methods in Engineering* 47 (2000) 2039–2072.
- [26] D. Zorin, P. Schröder, W. Sweldens, Interactive multiresolution mesh editing, in: *SIGGRAPH 1996 Conference Proceedings*, pp. 257–264.
- [27] M. Lounsbery, T. D. DeRose, J. Warren, Multiresolution analysis for surfaces of arbitrary topological type, *ACM Transactions of Graphics* 16 (1997) 34–73.
- [28] J. Haslinger, K. Ito, T. Kozubek, K. Kunisch, G. Peichl, On the shape derivative for problems of Bernoulli type, *Interfaces and Free Boundaries* 11 (2009) 317–330.

- [29] M. Meyer, M. Desbrun, P. Schröder, A. H. Barr, Discrete differential-geometry operators for triangulated 2-manifolds, in: *Visualization and Mathematics III (2002)*, pp. 113–134.
- [30] P. Clément, Approximation by finite element functions using local regularization, *RAIRO Analyse Numérique* 9 (1975) 77–84.
- [31] F. Cirak, Q. Long, Subdivision shells with exact boundary control and non-manifold geometry, *International Journal for Numerical Methods in Engineering* 88 (2011) 897–923.
- [32] H. Biermann, A. Levin, D. Zorin, Piecewise smooth subdivision surfaces with normal control, in: *SIGGRAPH 2000 Conference Proceedings*, New Orleans, LA, pp. 113–120.
- [33] K. Bandara, T. Rüber, F. Cirak, Variational shape optimisation using immersed finite elements and multiresolution surfaces, To be submitted. (2013).
- [34] K. Svanberg, The method of moving asymptotes — a new method for structural optimization, *International Journal for Numerical Methods in Engineering* 24 (1987) 359–373.
- [35] K. Svanberg, A class of globally convergent optimization methods based on conservative convex separable approximations, *SIAM Journal on Optimization* 12 (2002) 555–573.
- [36] S. G. Johnson, The nlopt nonlinear-optimization package, <http://ab-initio.mit.edu/nlopt>.
- [37] M. Flucher, M. Rumpf, Bernoulli’s free-boundary problem, qualitative theory and numerical approximation, *Journal für die reine und angewandte Mathematik* 486 (1997) 165–204.
- [38] J. Cottrell, T. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley & Sons Ltd., Chichester, 2009.
- [39] M. A. Scott, R. N. Simpson, J. A. Evans, S. Lipton, S. P. A. Bordas, T. J. R. Hughes, T. W. Sederberg, Isogeometric boundary element analysis using unstructured T-splines, *Computer Methods in Applied Mechanics and Engineering* 254 (2013) 197–221.
- [40] F. Hettlich, W. Rundell, Iterative methods for the reconstruction of an inverse potential problem, *Inverse Problems* 12 (1996) 251–266.
- [41] N. Nemitz, M. Bonnet, Topological sensitivity and FMM-accelerated BEM applied to 3D acoustic inverse scattering, *Engineering Analysis with Boundary Elements* (2008) 957–970.

Erschienene Preprints ab Nummer 2013/1

- 2013/1 L. John, P. Pustějovská, O. Steinbach: On hemodynamic indicators related to aneurysm blood flow.
- 2013/2 G. Of, O. Steinbach (eds.): 9th Austrian Numerical Analysis Day. Book of Abstracts.
- 2013/3 D. Amann, A. Blaszczyk, G. Of, O. Steinbach: Simulation of floating potentials in industrial applications by boundary element methods.
- 2013/4 L. John, O. Steinbach: Schur complement preconditioners for the biharmonic Dirichlet boundary value problem.
- 2013/5 J. Hron, P. Pustejovska: Comparison of stabilized finite element methods for simulation of flow of diluted polymeric liquids.
- 2013/6 L. John, P. Pustejovska, O. Steinbach: On the influence of the wall shear stress vector form on hemodynamic indicators.
- 2013/7 M. Bulicek, P. Pustejovska: Existence analysis for a model describing flow of an incompressible chemically reacting non-Newtonian fluid.
- 2013/8 C. M. Augustin, G. A. Holzappel, O. Steinbach: Classical and all-floating FETI methods for the simulation of arterial tissues.
- 2013/9 A. Kimeswenger, O. Steinbach, G. Unger: Coupled finite and boundary element methods for fluid-solid interaction eigenvalue problems.
- 2013/10 U. Langer, M. Schanz, O. Steinbach, W. L. Wendland (eds.): 11. Workshop on Fast Boundary Element Methods in Industrial Applications. Book. of Abstracts.
- 2013/11 A. Kimeswenger, O. Steinbach: Boundary element methods for exterior boundary control problems