Multiresolution Shape Optimisation with Subdivision Surfaces

Fehmi Cirak and Kosala Bandara

Abstract We review our recent work on multiresolution shape optimisation and present its application to elastic solids, electrostatic field equations and thin-shells. In the spirit of isogeometric analysis the geometry of the domain is described with subdivision surfaces and different resolutions of the same surface are used for optimisation and analysis. The analysis is performed using a sufficiently fine control mesh with a fixed resolution. During shape optimisation the geometry is updated starting with the coarsest control mesh and then moving on to increasingly finer control meshes. The transfer of data between the geometry and analysis representations is accomplished with subdivision refinement and coarsening operators. Moreover, we discretise elastic solids with the immersed finite element method, electrostatic field equations with the boundary element method and thin-shells with the subdivision finite element technique. In all three discretisation techniques there is no need to generate and maintain an analysis-suitable volume discretisation.

1 Introduction

As widely discussed in isogeometric analysis literature, the geometry representations used in today's computer aided design (CAD) and computational analysis software are inherently incompatible [19]. This is particularly limiting in shape optimisation in which a geometry model is iteratively updated based on the results of a computational analysis [6, 8, 17]. It is tedious and often impossible to repeatedly map the results from an analysis mesh back to the CAD model. To remedy this, the shape optimisation of shells and solids by directly optimising the CAD geometry model using isogeometric analysis has been recently explored [12, 16, 21, 37].

In the present paper we review the multiresolution shape optimisation technique and present its application to linear elastic solids, electrostatic field equations and thin-shell structures [2–4]. We discretise elastostatic boundary value problems using immersed, or embedded, finite elements, see, e.g., [28, 29, 31, 32], which have clear

F. Cirak (✉) • K. Bandara
Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK
e-mail: fc286@cam.ac.uk; kkmb2@cam.ac.uk

B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*,
Lecture Notes in Computational Science and Engineering 107,
DOI 10.1007/978-3-319-23315-4_6

advantages when applied to structural shape optimisation [1, 18]. In order to cope with unbounded domains we discretise electrostatic problems with the boundary element method. Crucially, in immersed finite elements and boundary elements the geometry of the domain boundary can be updated without needing to generate or maintain domain meshes. Shell structures are modelled as a surface and are discretised with subdivision finite elements, which use the subdivision surfaces as basis functions [11, 12].

The domain boundaries and the shell mid-surface are represented with subdivision surfaces. Although historically subdivision and related techniques have originated in computer graphics, they recently became available in several CAD software packages, including Autodesk Fusion 360, PTC Creo and CATIA. As will be demonstrated in this paper, subdivision surfaces provide an elegant isogeometric analysis-suitable, bidirectional mapping between the geometry and analysis models. In subdivision a geometry is described using a control mesh and a limiting process of repeated refinement [27, 40]. The refinement rules are usually adapted from knot refinement rules for splines [9, 14, 22]. We consider the Catmull-Clark [9] scheme based on quadrilateral meshes and the Loop [25] scheme based on triangular meshes. The Catmull-Clark scheme is the generalisation of cubic tensor-product b-splines to unstructured meshes and the Loop scheme is the generalisation of quartic box-splines. Both subdivision schemes lead to smooth surfaces even in case of unstructured meshes with extraordinary vertices. The hierarchy of control meshes underlying subdivision surfaces lends itself naturally to multiresolution decomposition of geometries [26, 41]. The size of the geometric region influenced by each vertex depends on the resolution of the control mesh, editing coarser levels leads to large-scale changes while editing finer levels lead to small-scale changes.

The introduced multiresolution optimisation approach relies on subdivision curves/surfaces for the description of boundaries. The multiresolution paradigm allows us to describe the same geometry with control meshes of different resolution for analysis and optimisation purposes. For finite element and boundary element analysis a relatively fine control mesh is used in order to minimise the discretisation errors. In contrast, the degrees of freedom in optimisation (i.e., design variables) are chosen as the vertex coordinates of a coarser control mesh. Perhaps counterintuitively, the use of the same fine control mesh for optimisation and analysis leads to suboptimal optimisation results with possible non-physical geometry oscillations. This behaviour is also known from earlier work on structural optimisation in which the finite element nodes were used as optimisation design variables [8, 17]. This paper will demonstrate that the best optimisation results are achieved when starting with a coarse control mesh increasingly finer control meshes are optimised. During the optimisation iterations the refinement level of the control mesh is increased each time a minimum is reached.

2 Multiresolution Subdivision Surfaces

2.1 Subdivision Refinement of Spline Curves

The refinability property of cubic b-splines can be utilised to derive a corresponding subdivision scheme. To illustrate this, we consider the coarse knot sequence $\xi_i^0 = 0, 1, 2, 3, \ldots$ and the fine knot sequence $\xi_i^1 = 0, 0.5, 1, 1.5, 2, 2.5, 3, \ldots$. We denote the b-splines on the coarse knot sequence with $B_i^0(\xi)$ and the ones on the fine knot sequence with $B_i^1(\xi)$, see Fig. 1. According to the b-spline refinability equation, see, e.g., [7, 40], it is possible to represent the coarse b-splines (in the interior) as a linear combination of the fine b-splines

$$B_i^0(\xi) = \sum_{j=0}^{4} S_{ij} B_{2i+j-2}^1(\xi) \quad \text{with} \quad S_{ij} = \frac{1}{8}\binom{4}{i}, \tag{1}$$

where S_{ij} is the subdivision matrix with its entries given in terms of the usual binomial coefficients. In order to keep the derivations simple we exclude from our discussion the b-splines close to the boundaries. For subsequent derivations (1) is best expressed in matrix notation

$$\boldsymbol{B}^0 = \boldsymbol{S}\boldsymbol{B}^1 . \tag{2}$$

Next, we consider a spline curve defined in terms of the coarse b-splines and the corresponding control vertices, i.e.,

$$\boldsymbol{x}_h(\xi) = \boldsymbol{B}^0 \cdot \boldsymbol{x}^0 \tag{3}$$

with matrix $\boldsymbol{x}^0$ containing the coordinates of control vertices. The number of columns of $\boldsymbol{x}^0$ is equal to space dimension and the number of rows is equal to the number of all control vertices.

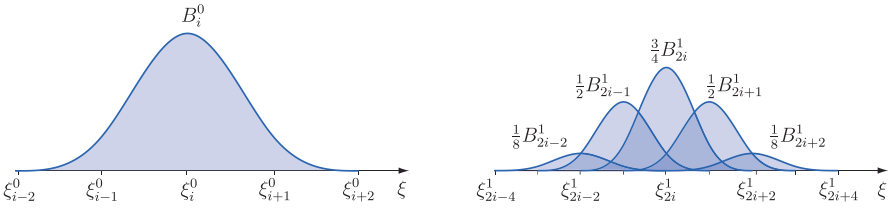

Fig. 1 Refinement relation for cubic b-splines

Introducing the refinement relation (2) into (3) the spline curve can be expressed with

$$\boldsymbol{x}_h(\xi) = \left(\boldsymbol{S}\boldsymbol{B}^1\right) \cdot \boldsymbol{x}^0 = \boldsymbol{B}^1 \cdot \left(\boldsymbol{S}^{\mathsf{T}}\boldsymbol{x}^0\right) . \tag{4}$$

This implies that the control vertex coordinates on the finer level can be computed with the subdivision relation

$$\boldsymbol{x}^1 = \boldsymbol{S}^{\mathsf{T}}\boldsymbol{x}^0 . \tag{5}$$

In subdivision schemes the described refinement approach is applied recursively, i.e.,

$$\boldsymbol{x}^{\ell+1} = \boldsymbol{S}^{\mathsf{T}}\boldsymbol{x}^{\ell} , \tag{6}$$

where $\boldsymbol{x}^{\ell+1}$ and $\boldsymbol{x}^{\ell}$ are two matrices containing the coordinates of all the vertices at levels $\ell+1$ and ℓ with $\ell \geq 0$. Although the dimensions of $\boldsymbol{S}$ increase with ℓ we denote all subdivision matrices with $\boldsymbol{S}$ since each row has the same non-zero components only shifted relative to adjacent rows [3]. The successive refinement of a given control polygon using (6) is illustrated in Fig. 2.

For computer implementation and generalisation to bivariate splines, it is instructive to think that the refinement of a control polygon according to (6) consists of a refinement and an averaging step. In the refinement step each segment of the polygon is subdivided into two segments, see Fig. 3a. Subsequently, the vertex

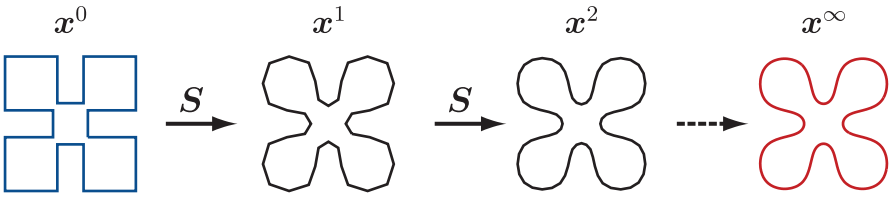

Fig. 2 Subdivision refinement of a given control polygon (shown *left*)

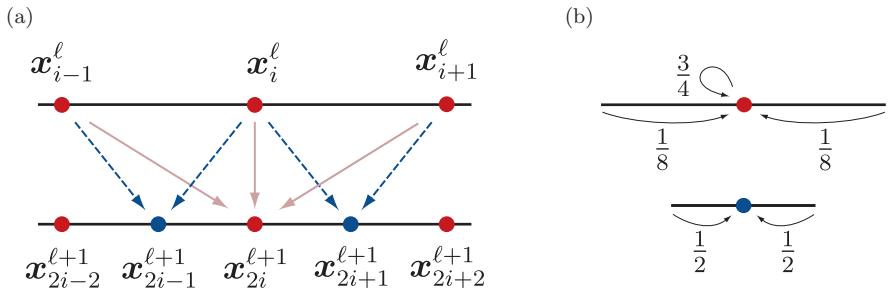

Fig. 3 Subdivision refinement for cubic spline curves. (**a**) Refinement by bisectioning and vertex renumbering. (**b**) Stencils for even (*top*) and odd (*bottom*) vertices

coordinates of the refined polygon are determined by averaging the coarse vertex coordinates with the two stencils shown in Fig. 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 where newly inserted vertices receive odd numbers. According to Fig. 3b, for a given coarse polygon of level ℓ with vertex coordinates $\boldsymbol{x}_i^\ell$ a refined polygon of level $\ell + 1$ with vertex coordinates $\boldsymbol{x}_i^{\ell+1}$ is computed with

$$\boldsymbol{x}_{2i}^{\ell+1} = \frac{1}{8}\boldsymbol{x}_{i-1}^{\ell} + \frac{3}{4}\boldsymbol{x}_i^{\ell} + \frac{1}{8}\boldsymbol{x}_{i+1}^{\ell}\,, \qquad \boldsymbol{x}_{2i+1}^{\ell+1} = \frac{1}{2}\boldsymbol{x}_i^{\ell} + \frac{1}{2}\boldsymbol{x}_{i+1}^{\ell}\,. \tag{7}$$

2.2 *Subdivision Refinement of Bivariate Surfaces*

Similar to tensor-product b-splines the tensor products of one-dimensional subdivision stencils yield the corresponding subdivision stencils for surfaces. The tensor product construction works only for quadrilateral meshes and in the refinement step each quadrilateral is subdivided into four quadrilaterals, see Fig. 4a. It is evident that the tensor-product stencils only apply to meshes in which each vertex within the domain is connected to four faces. The number of faces connected to a vertex is referred to as the valence of that vertex. The domain vertices with a valence other than four are known as extraordinary vertices or star-vertices.

For subdivision surfaces using quadrilateral meshes, the original stencils proposed by Catmull and Clark [9], shown in Fig. 5, are used. Note, the vertex stencil in Fig. 5c depends on the valence v of the vertex and reduces to a standard tensor product stencil for structured meshes with valence $v = 4$. There is mathematical theory which shows that the resulting surface is C^2 continuous almost everywhere except at the extraordinary vertices where it is only C^1 continuous [27]. In addition, there are also extended subdivision stencils for vertices on edges, creases and corners, see, e.g., [5, 10, 39]. In this context, a crease is a line on the surface across which the surface is only C^0 continuous. As an illustrative example, Fig. 6

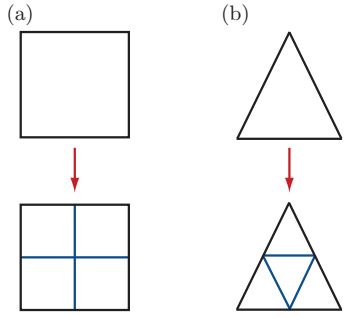

Fig. 4 Refinement by quadrisectioning. (**a**) Quadrilaterals. (**b**) Triangles

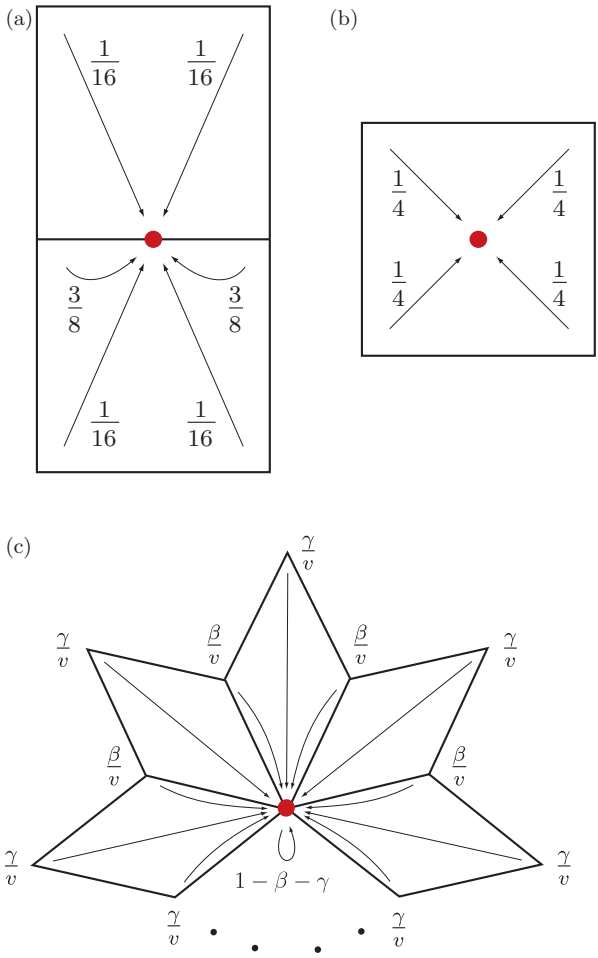

Fig. 5 Subdivision stencils for the Catmull-Clark scheme. Each of the stencils are used for computing the coordinates of vertices of the type indicated by the *red dot*. The weights for the vertex stencil are $\beta = \frac{3}{2v}$ and $\gamma = \frac{1}{4v}$, where v is the valence of the vertex. (**a**) Edge stencil. (**b**) Face stencil. (**c**) Vertex stencil

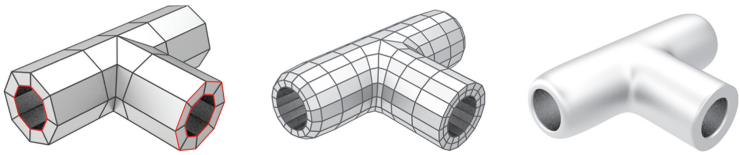

Fig. 6 Refinement of a given quadrilateral control mesh (shown *left*) with Catmull-Clark subdivision. In the vicinity of the edges in *red* modified subdivision stencils are applied in order preserve the sharp edge. The once subdivided mesh is shown in the *middle* and the limit surface on the *right*

shows the subdivision refinement of a control mesh for a T-junction geometry with extraordinary vertices and prescribed crease edges.

In case of triangular control meshes the subdivision scheme introduced by Loop [25] can be used. On three-directional triangular meshes (with each vertex having valence six) the Loop scheme yields quartic box-splines. 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. 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 shown in Fig. 4b. Subsequently, the vertex coordinates of the refined mesh are computed with the subdivision stencils given in Fig. 7 [38]. Figure 8 shows a mechanical connector geometry containing extraordinary vertices and sharp features described with the extended subdivision surfaces.

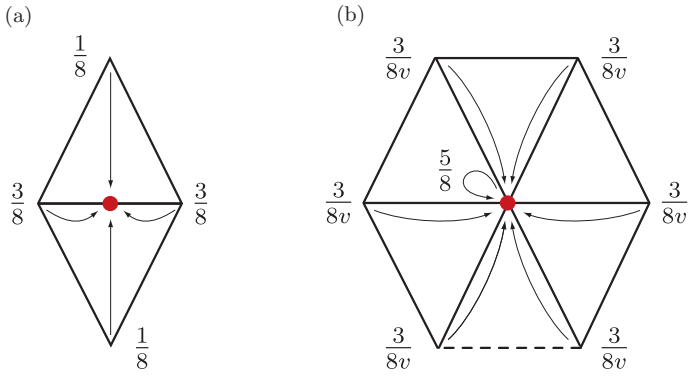

Fig. 7 Subdivision stencils for the Loop scheme. Each of the stencils are used for computing the coordinates of vertices of the type indicated by the *red dot*. (**a**) Edge stencil. (**b**) Face stencil

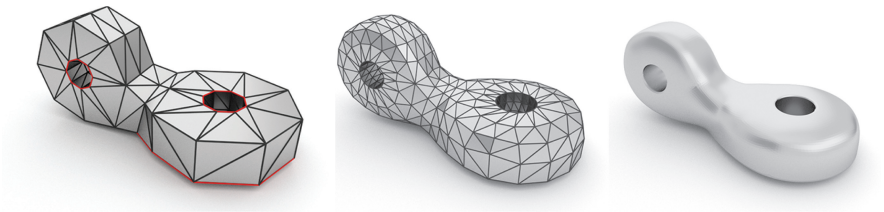

Fig. 8 Subdivision refinement of a given triangular control mesh (shown *left*) with Loop subdivision. In the vicinity of the edges in *red* modified subdivision stencils are applied in order preserve the sharp edge. The once subdivided mesh is shown in the *middle* and the limit surface on the *right*

2.3 Multiresolution Editing

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, 41]. The basic idea in multiresolution editing is to modify the coarse mesh vertex coordinates to apply large-scale changes (to the limit surface) and to modify the fine mesh vertex coordinates to add localised changes. By way of example, this is illustrated in Fig. 9 for the connector geometry previously introduced in Fig. 8. First the control mesh vertex coordinates $\boldsymbol{x}^0$ at level $\ell = 0$ are modified with $\boldsymbol{x}^0 + \boldsymbol{d}^0$, where $\boldsymbol{d}^0$ can be thought of as a user given perturbation vector. In the considered example, $\boldsymbol{d}^0$ 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., $\boldsymbol{x}^1 + \boldsymbol{d}^1 = \boldsymbol{S}\boldsymbol{x}^0 + \boldsymbol{d}^1$. This results in a more localised change on the limit surface. 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 simultaneously edit coarse and fine resolutions, see, e.g., [3, 41]. 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, 41]. To compute such a decomposition it is necessary to define, in addition to the subdivision refinement, a coarsening operation

$$\boldsymbol{x}^\ell = \boldsymbol{R}\boldsymbol{x}^{\ell+1}\,. \tag{8}$$

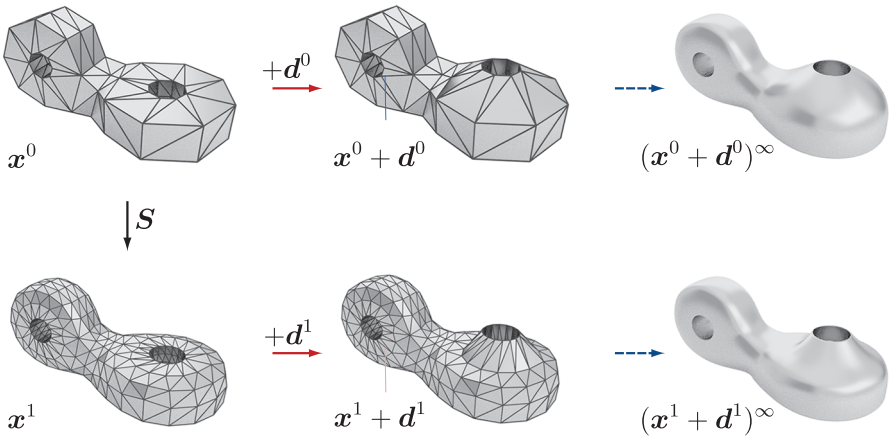

Fig. 9 Multiresolution editing of the connector geometry introduced in Fig. 8. The geometry is modified by moving the edge of one of the holes in the vertical direction. On the first row the modification is performed on level $\ell = 0$ and on the second row it is performed on level $\ell = 1$. Notice the effect of the modification level on the limit surface (*last column*)

The coarsening matrix $\boldsymbol{R}$ enables the computation of the coarse control mesh $\boldsymbol{x}^\ell$ corresponding to a given edited fine control mesh $\boldsymbol{x}^{\ell+1}$. In contrast, recall that the subdivision matrix $\boldsymbol{S}$ enables the computation of a refined mesh from a given coarse mesh, cf. (6). Different choices for the matrix $\boldsymbol{R}$ are possible. For instance, it can be determined with discrete least squares fitting

$$\boldsymbol{x}^\ell = \underset{\boldsymbol{y}^\ell}{\operatorname{argmin}} \|\boldsymbol{x}^{\ell+1} - \boldsymbol{S}\boldsymbol{y}^\ell\|^2 , \tag{9}$$

which leads to

$$\boldsymbol{S}^\mathsf{T}\boldsymbol{S}\boldsymbol{x}^\ell = \boldsymbol{S}^\mathsf{T}\boldsymbol{x}^{\ell+1} . \tag{10}$$

By comparison with (8) we observe that the coarsening matrix has to be $\boldsymbol{R} = (\boldsymbol{S}^\mathsf{T}\boldsymbol{S})^{-1}\boldsymbol{S}^\mathsf{T}$. For more details we refer to [3, 4].

Instead of using least squares fitting the coarsening matrix $\boldsymbol{R}$ can also be defined based on quasi-interpolation [23] or smoothing [41]. On the other hand coarsening by simply subsampling of the fine control mesh usually leads to artefacts in form of oscillations in the coarse control mesh. The proposed least squares fit approach is not very common in computer graphics because of the need for interactivity and fast processing times. Although the least squares matrix in (10) is sparse its solution cannot be found at interactive rates.

Subdivision surfaces do not by themselves provide the possibility of simultaneously editing coarse and fine control meshes. For instance, after a fine control mesh is edited it is not possible to further edit a coarser level in order to apply larger scale changes to the geometry. Simultaneous editing of different levels can be achieved with a wavelet-like multiresolution decomposition of the control meshes, as discussed in [3, 41].

3 Governing Equations of Shape Optimisation

In this section, we revisit the governing equations for shape optimisation of linear elastic solids, electrostatic field equations and thin-shells. The considered cost functions are the structural compliance for elastic solids and thin-shells and the normal flux over the domain boundaries for electrostatic problems. For computing the derivatives of the cost function with respect to the domain boundary perturbations we consider both the continuous and the discrete adjoint formulations. Specifically, for elastic and electrostatic field equations we use the continuous formulation and for thin-shells we use the discrete formulation. In the continuous formulation the differentiation is first performed analytically and subsequently the resulting boundary value problems are discretised with a method of choice. In contrast, in the discrete formulation first the governing equations are discretised and subsequently the resulting algebraic equations differentiated.

3.1 Continuous Shape Sensitivity Formulation

In continuous shape sensitivity analysis it is necessary to differentiate functionals that are defined on varying domains. To this end, borrowing terminology from continuum mechanics, we consider a reference configuration with domain Ω and boundary Γ and a perturbed configuration with domain Ω_t and boundary Γ_t, see Fig. 10. The corresponding deformation $\boldsymbol{\varphi}_t(\boldsymbol{x})$ which maps material points $\boldsymbol{x} \in \Omega$ onto $\boldsymbol{x}_t \in \Omega_t$ is assumed to be of the form

$$\boldsymbol{x}_t = \boldsymbol{\varphi}_t(\boldsymbol{x}) = \boldsymbol{x} + t\boldsymbol{V}\,, \tag{11}$$

where $\boldsymbol{V}$ is a prescribed constant vector field and t is a scalar parameter. It is convenient to interpret $\boldsymbol{V}$ as a velocity field and t as the time parameter.

In the sequel the derivatives of domain and boundary integrals with respect to boundary perturbations are needed. First, we consider the generic domain integral

$$I_1(\Omega_t) = \int_{\Omega_t} \psi(\boldsymbol{x}_t)\ \mathrm{d}\,\Omega_t \tag{12}$$

with a scalar integrand $\psi(\boldsymbol{x}_t)$. At the reference configuration $\boldsymbol{x}$, the derivative of the domain integral in the direction of $\boldsymbol{V}$ is defined with

$$\frac{\partial I_1}{\partial \Omega}\boldsymbol{V} = \frac{d}{dt} I_1(\boldsymbol{x} + t\boldsymbol{V})\Big|_{t=0} = \frac{d}{dt}\int_{\Omega_t} \psi(\boldsymbol{x}_t)\ \mathrm{d}\,\Omega_t\Big|_{t=0}\,. \tag{13}$$

By applying standard techniques from continuum mechanics and shape calculus, see [3, 13, 33], we obtain

$$\frac{\partial I_1}{\partial \Omega}\boldsymbol{V} = \int_{\Omega} (\nabla\psi(\boldsymbol{x})\cdot\boldsymbol{V} + \psi(\boldsymbol{x})\nabla\cdot\boldsymbol{V})\ \mathrm{d}\,\Omega = \int_{\Gamma} \psi(\boldsymbol{x})(\boldsymbol{V}\cdot\boldsymbol{n})\ \mathrm{d}\,\Gamma\,, \tag{14}$$

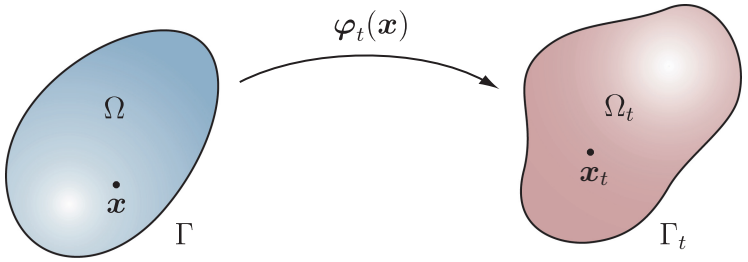

Fig. 10 Reference and the perturbed configurations (*left* and *right*, respectively)

where $\boldsymbol{n}$ is the unit normal to the boundary. As to be expected, this integral is zero when the perturbation direction $\boldsymbol{V}$ is orthogonal to the boundary normal $\boldsymbol{n}$. Perturbations tangential to the boundary do not lead to a change in I_1.

Next, we consider the generic boundary integral

$$I_2(\Gamma_t) = \int_{\Gamma_t} \psi(\boldsymbol{x}_t)\ \mathrm{d}\,\Gamma_t \tag{15}$$

with a scalar integrand $\psi(\boldsymbol{x}_t)$. The differentiation of this integral at the reference configuration $\boldsymbol{x}$ in the direction of $\boldsymbol{V}$ is more involved and can be found in standard texts on shape calculus, see [13, 33],

$$\frac{\partial I_2}{\partial \Gamma}\boldsymbol{V} = \int_{\Gamma} \left(\nabla\psi(\boldsymbol{x}) \cdot \boldsymbol{n} + H(\boldsymbol{x})\psi(\boldsymbol{x})\right)(\boldsymbol{V}\cdot\boldsymbol{n})\ \mathrm{d}\,\Gamma\,, \tag{16}$$

where $H(\boldsymbol{x})$ is the mean curvature on Γ.

3.1.1 Compliance Optimisation in Elasticity

As the first specific optimisation problem we consider linear elasticity. The equilibrium equation for a solid body with the domain Ω is given by

$$\nabla\cdot\boldsymbol{\sigma}(\boldsymbol{u}) + \boldsymbol{f} = \boldsymbol{0} \qquad \text{in } \Omega\,, \tag{17a}$$

$$\boldsymbol{u} = \boldsymbol{0} \qquad \text{on } \Gamma_D\,, \tag{17b}$$

$$\boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n} = \bar{\boldsymbol{t}} \qquad \text{on } \Gamma_N\,, \tag{17c}$$

where $\boldsymbol{\sigma}$ is the stress tensor, $\boldsymbol{u}$ is the displacement vector, $\boldsymbol{f}$ is the external load vector and $\bar{\boldsymbol{t}}$ is the prescribed traction on the Neumann boundary Γ_N with the outward normal $\boldsymbol{n}$. On Γ_D, for simplicity, only homogenous Dirichlet boundary conditions are assumed.

We assume a homogenous linear elastic material model

$$\boldsymbol{\sigma}(\boldsymbol{u}) = \boldsymbol{C} : \boldsymbol{\epsilon}(\boldsymbol{u}) \tag{18}$$

with the linear elastic strain tensor

$$\boldsymbol{\epsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla\boldsymbol{u} + \nabla^{\mathsf{T}}\boldsymbol{u})\,. \tag{19}$$

The aim of shape optimisation is to find a shape Ω that minimises a cost functional $J(\Omega, \boldsymbol{u})$ such that

$$\min_{\Omega} J(\Omega, \boldsymbol{u}) \tag{20}$$

where the displacement vector $\boldsymbol{u}$ has to satisfy (17). In practice there are usually additional constraints, e.g., pertaining to the volume of the domain Ω, which are neglected here for brevity. In case of structural compliance, the cost functional $J(\Omega,\boldsymbol{u})$ reads

$$J(\Omega,\boldsymbol{u}) = \int_{\Omega} \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\epsilon}(\boldsymbol{u}) \,\mathrm{d}\,\Omega = \int_{\Omega} \boldsymbol{f}\cdot\boldsymbol{u} \,\mathrm{d}\,\Omega + \int_{\Gamma_N} \bar{\boldsymbol{t}}\cdot\boldsymbol{u} \,\mathrm{d}\,\Gamma\,. \tag{21}$$

In adjoint shape sensitivity analysis the minimisation problem (20) with the equilibrium equation (17) as a constraint is expressed with the functional

$$\begin{aligned} L(\Omega,\boldsymbol{u},\boldsymbol{\lambda}) = J(\Omega,\boldsymbol{u}) &+ \int_{\Omega} \nabla\boldsymbol{\lambda} : \boldsymbol{\sigma}(\boldsymbol{u}) \,\mathrm{d}\,\Omega - \int_{\Omega} \boldsymbol{\lambda}\cdot\boldsymbol{f} \,\mathrm{d}\,\Omega \\ &- \int_{\Gamma_D} \boldsymbol{u}\cdot(\boldsymbol{C} : \nabla\boldsymbol{\lambda})\boldsymbol{n} + \boldsymbol{\lambda}\cdot\boldsymbol{\sigma}(\boldsymbol{u})\boldsymbol{n} \,\mathrm{d}\,\Gamma - \int_{\Gamma_N} \boldsymbol{\lambda}\cdot\bar{\boldsymbol{t}} \,\mathrm{d}\,\Gamma\,, \end{aligned} \tag{22}$$

where $\boldsymbol{\lambda}$ is a vector-valued Lagrange parameter field. The stationarity condition for the functional $L(\Omega,\boldsymbol{u},\boldsymbol{\lambda})$, i.e.,

$$\delta L(\Omega,\boldsymbol{u},\boldsymbol{\lambda}) = 0 \tag{23}$$

provides the complete set of equations for shape optimisation. The variation of $L(\Omega,\boldsymbol{u},\boldsymbol{\lambda})$ with respect to the Lagrange parameter $\boldsymbol{\lambda}$ yields the original boundary value problem (17). The variation of $L(\Omega,\boldsymbol{u},\boldsymbol{\lambda})$ with respect to the displacement vector $\boldsymbol{u}$ yields the adjoint boundary value problem

$$\nabla\cdot\boldsymbol{\sigma}(\boldsymbol{\lambda}) - \boldsymbol{f} = \boldsymbol{0} \qquad \text{in } \Omega\,, \tag{24a}$$

$$\boldsymbol{\lambda} = \boldsymbol{0} \qquad \text{on } \Gamma_D\,, \tag{24b}$$

$$\boldsymbol{\sigma}(\boldsymbol{\lambda})\boldsymbol{n} = -\bar{\boldsymbol{t}} \qquad \text{on } \Gamma_N\,. \tag{24c}$$

By comparing the adjoint problem with the original boundary value problem (17) we deduce that $\boldsymbol{\lambda} = -\boldsymbol{u}$. This holds only when the structural compliance (21) is the cost functional.

Next, we consider the variation of $L(\Omega,\boldsymbol{u},\boldsymbol{\lambda})$ with respect to the domain Ω. In deriving it we make use of the domain deformation $\boldsymbol{x}_t = \boldsymbol{\varphi}_t(\boldsymbol{x}) = \boldsymbol{x} + t\boldsymbol{V}$ introduced in (11). In shape optimisation of linear elasticity problems usually only the parts of the Neumann boundary with zero tractions are allowed to move, i.e.,

$$\begin{aligned} &\boldsymbol{V} = \boldsymbol{0} \quad \text{on } \Gamma_D\,, \\ &\boldsymbol{V} = \boldsymbol{0} \quad \text{on } \Gamma_N \text{ with } \boldsymbol{\sigma}\boldsymbol{n} = \bar{\boldsymbol{t}}\,, \\ &\boldsymbol{V} \neq \boldsymbol{0} \quad \text{on } \Gamma_N \text{ with } \boldsymbol{\sigma}\boldsymbol{n} = \boldsymbol{0}\,. \end{aligned} \tag{25}$$

The variation of the Lagrangian (22) with respect to the domain perturbations yields the shape gradient. For structural compliance (21) as the cost functional (i.e., $\boldsymbol{\lambda} = -\boldsymbol{u}$) and with the result (14) at hand we obtain for the shape gradient

$$\frac{\partial L}{\partial \Omega}\boldsymbol{V} = \int_{\Gamma_N} (2\boldsymbol{u}\cdot\boldsymbol{f} - \nabla\boldsymbol{u} : \boldsymbol{\sigma}(\boldsymbol{u}))\,(\boldsymbol{V}\cdot\boldsymbol{n})\ \mathrm{d}\,\Gamma = \int_{\Gamma_N} g(\boldsymbol{u})(\boldsymbol{V}\cdot\boldsymbol{n})\ \mathrm{d}\,\Gamma\,, \tag{26}$$

where $g(\boldsymbol{u})$ is the shape kernel function. It is worth emphasising that without restricting $\boldsymbol{V}$ as stated in (25) the shape derivate would contain several more terms.

In iterative shape optimisation the shape kernel function $g(\boldsymbol{u})$ is used as gradient information. In order to achieve a decrease in the cost functional the boundary perturbation is chosen proportional to

$$\boldsymbol{V} = -g(\boldsymbol{u})\boldsymbol{n} \tag{27}$$

such that

$$\frac{\partial L}{\partial \Omega}\boldsymbol{V} = -\int_{\Gamma_N} g(\boldsymbol{u})^2\ \mathrm{d}\,\Gamma\,. \tag{28}$$

During the shape optimisation the boundary value problem (17) has to be repeatedly solved on a continuously changing domain. In a conventional finite element setting this would require frequent mesh smoothing or updating. Therefore, immersed, or embedded, finite element approaches that do not require remeshing have clear advantages in shape optimisation. In the present work, we use an immersed finite element technique previously developed in the context of incompressible fluid-structure interaction [28–30]. For more details see [3].

3.1.2 Boundary Flux Optimisation in Electrostatics

The motivation for our second optimisation problem comes from the design of high-voltage electrical devices. Avoidance of electrical breakdown is one of the key design considerations for such devices. In a first approximation, electrical breakdown can be avoided by reducing the electrical field strength on critical components. The aim of shape optimisation is to modify the geometry of components so that the maximum electric field strength on surfaces, i.e., the normal flux, is reduced.

The electrostatic field equation in absence of space charges is given by a Dirichlet boundary value problem for the Laplace equation

$$\begin{aligned} -\Delta u &= 0 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \Gamma_{D_0}, \\ u &= \overline{u} \quad \text{on } \Gamma_{D_f}, \end{aligned} \tag{29}$$

where u is the scalar electric potential or voltage, Ω is a multiply connected domain with a Dirichlet boundary consisting of a (to be optimised) free part Γ_{D_f} and a fixed part Γ_{D_0}. It is assumed that the potentials on Γ_{D_f} and Γ_{D_0} are constant. The geometry of the free Dirichlet boundary Γ_{D_f} with the prescribed potential $\overline{u}$ is to be determined with shape optimisation. Notice that the electric potential u is denoted non-bold because it is a scalar field quantity.

In order to pointwise minimise the maximum of the normal flux on the free part of the boundary Γ_{D_f} we chose to minimise the cost functional

$$J(\Omega, u) = \frac{1}{2} \int_{\Gamma_{D_f}} \left(\frac{\partial u}{\partial \boldsymbol{n}} - Q \right)^2 \mathrm{d}\,\Gamma\,, \tag{30}$$

where $Q \geq 0$ is a user prescribed constant expected value.

The adjoint sensitivity analysis, as discussed in Sect. 3.1.1 for elasticity, is used for obtaining the adjoint boundary value problem and the shape gradient. The adjoint boundary value problem corresponding to the cost functional (30) reads

$$\begin{aligned} -\Delta\lambda &= 0 && \text{in } \Omega, \\ \lambda &= 0 && \text{on } \Gamma_{D_0}, \\ \lambda &= \frac{\partial u}{\partial \boldsymbol{n}} - Q && \text{on } \Gamma_{D_f}\,. \end{aligned} \tag{31}$$

Importantly, the adjoint solution λ is different from the solution u of the original problem so that in applications both problems (29) and (31) have to be solved.

Moreover, the shape gradient corresponding to the cost functional (30) reads

$$\frac{\partial L}{\partial \Omega} \boldsymbol{V} = \int_{\Gamma_{D_f}} \left(-\frac{\partial \lambda}{\partial \boldsymbol{n}} \frac{\partial u}{\partial \boldsymbol{n}} - \frac{H}{2} \right) (\boldsymbol{V} \cdot \boldsymbol{n}) \, \mathrm{d}\,\Gamma = \int_{\Gamma_{D_f}} g(\lambda, u)(\boldsymbol{V} \cdot \boldsymbol{n}) \, \mathrm{d}\,\Gamma\,, \tag{32}$$

where the shape kernel function $g(\lambda, u)$ now depends on the electric potential u and adjoint solution λ. In deriving the shape gradient we made use of (16) and H is the mean curvature of the boundary.

As discussed for elasticity, during the iterative shape optimisation the shape kernel function $g(\boldsymbol{u})$ is used as gradient information. In order to achieve a decrease in the cost functional the boundary perturbation is chosen proportional to $\boldsymbol{V} = -g(\lambda, \boldsymbol{u})\boldsymbol{n}$.

For solving the original and the adjoint boundary value problems (29) and (31), respectively, we use the boundary element method, see e.g. [36], which does not require a domain discretisation and is ideal for solving problems with unbounded domains that occur in electrostatic field analysis. For more details see [3].

3.2 Discrete Shape Sensitivity Formulation

3.2.1 Compliance Optimisation of Thin-Shells

We use the Kirchhoff-Love energy functional for modelling the mechanical response of thin shells. Since the related equations are classically formulated in curvilinear coordinates, it is usually easier to compute the shape sensitivities using a discrete approach. To this end, we first discretise the governing equations with finite elements and subsequently differentiate the cost function and the discrete equilibrium equations with respect to control vertex coordinates.

The Kirchhoff-Love energy functional depends on mid-surface displacements and their first and second derivatives. Therefore, the basis functions have to be smooth and their second order derivatives square-integrable. As originally introduced by Cirak et al. [10, 11] Kirchhoff-Love shells can be elegantly discretised with subdivision basis functions. In the resulting discrete equilibrium equations the control vertex displacements are the only degrees of freedom.

To begin with, we consider a shell with the undeformed mid-surface Ω, the position vector $\boldsymbol{x}(\xi_1, \xi_2)$ and the displacement vector $\boldsymbol{u}(\xi_1, \xi_2)$. It is assumed that the mid-surface is parameterised in terms of the curvilinear coordinates (ξ_1, ξ_2). The two corresponding (covariant) surface basis vectors follow from

$$\boldsymbol{a}_\alpha = \boldsymbol{x}_{,\alpha} \,. \tag{33}$$

From here onwards Greek indices take the values 1 and 2 and a comma denotes partial differentiation with respect to curvilinear coordinates.

According to [11] the linearised membrane and bending strains $\alpha_{\alpha\beta}$ and $\beta_{\alpha\beta}$, respectively, are given by

$$\alpha_{\alpha\beta} = \frac{1}{2}(\boldsymbol{a}_\alpha \cdot \boldsymbol{u}_{,\beta} + \boldsymbol{u}_{,\alpha} \cdot \boldsymbol{a}_\beta) \,, \tag{34}$$

$$\begin{aligned} \beta_{\alpha\beta} = &-\boldsymbol{u}_{,\alpha\beta} \cdot \boldsymbol{a}_3 + \frac{1}{\sqrt{a}} \left[\boldsymbol{u}_{,1} \cdot (\boldsymbol{a}_{\alpha,\beta} \times \boldsymbol{a}_2) + \boldsymbol{u}_{,2} \cdot (\boldsymbol{a}_1 \times \boldsymbol{a}_{\alpha,\beta})\right] \\ &+ \frac{\boldsymbol{a}_3 \cdot \boldsymbol{a}_{\alpha,\beta}}{\sqrt{a}} \left[\boldsymbol{u}_{,1} \cdot (\boldsymbol{a}_2 \times \boldsymbol{a}_3) + \boldsymbol{u}_{,2} \cdot (\boldsymbol{a}_3 \times \boldsymbol{a}_1)\right] . \end{aligned} \tag{35}$$

With the above strain expressions at hand, the potential energy of a linear-elastic Kirchhoff-Love shell reads

$$\Pi(\boldsymbol{u}) = \int_\Omega \left(W^m(\alpha_{\alpha\beta}) + W^b(\beta_{\alpha\beta})\right) \mu \, \mathrm{d}\, \Omega + \Pi^{ext}(\boldsymbol{u}) \,, \tag{36}$$

where Ω is the shell mid-surface, W^m is the membrane energy density, W^b is the bending energy density, μ is the jacobian associated with the integration over the thickness and $\Pi^{ext}(\boldsymbol{u})$ is the potential of the external forces. For a linear elastic

material the two energy densities are defined with

$$W^m(\alpha_{\alpha\beta}) = \frac{1}{2}\frac{Et}{1-\nu^2} H^{\alpha\beta\gamma\delta}\alpha_{\alpha\beta}\alpha_{\gamma\delta}\,, \tag{37a}$$

$$W^b(\beta_{\alpha\beta}) = \frac{1}{2}\frac{Et^3}{12(1-\nu^2)} H^{\alpha\beta\gamma\delta}\beta_{\alpha\beta}\beta_{\gamma\delta}\,, \tag{37b}$$

where t is the shell thickness, E is the Young's modulus, ν is the Poisson's ratio and $H^{\alpha\beta\gamma\delta}$ is an auxiliary fourth order tensor

$$H^{\alpha\beta\gamma\delta} = \nu\, a^{\alpha\beta}a^{\gamma\delta} + \frac{1}{2}(1-\nu)\,(a^{\alpha\gamma}a^{\beta\delta} + a^{\alpha\delta}a^{\beta\gamma})\,. \tag{38}$$

The contravariant metric tensor components $a^{\alpha\beta} = \boldsymbol{a}^\alpha \cdot \boldsymbol{a}^\beta$ are computed from the contravariant basis vectors defined according to $\boldsymbol{a}^\alpha \cdot \boldsymbol{a}_\beta = \delta^\alpha_\beta$, where δ^α_β is the Kronecker delta.

Next, the mid-surface position and displacement vector are discretised with subdivision basis functions

$$\boldsymbol{x}_h(\xi_1,\xi_2) = \sum_i N_i(\xi_1,\xi_2)\boldsymbol{x}_i\,, \qquad \boldsymbol{u}_h(\xi_1,\xi_2) = \sum_i N_i(\xi_1,\xi_2)\boldsymbol{u}_i\,, \tag{39}$$

where $N_i(\xi_1,\xi_2)$ is the basis function, $\boldsymbol{x}_i$ is the position vector and $\boldsymbol{u}_i$ is the displacement of a control vertex with the index i. Recall that on structured meshes the subdivision basis functions are equivalent to quartic box-splines in case of the Loop scheme and equivalent to tensor-product cubic b-splines in case of the Catmull-Clark scheme. As shown by Stam [34, 35] in the vicinity of extraordinary vertices the subdivision surfaces can be evaluated in the same way as regular patches after only few steps of refinement. For implementation details see [10, 11].

In order to obtain the discrete equilibrium equations, first the interpolation (39) is introduced into the energy functional (36) and subsequently the integrals are evaluated with Gauss quadrature. Computing the stationary points of the discretised energy functional yields the discrete equilibrium equations

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{f}, \tag{40}$$

where $\boldsymbol{K}$ is the stiffness matrix, $\boldsymbol{f}$ is the array of vertex forces and, with a slight abuse of notation, $\boldsymbol{u}$ is the array of vertex displacements.

In discrete shape sensitivity formulation we aim to minimise a discretised cost function

$$\min_{\boldsymbol{x}} J(\boldsymbol{x},\boldsymbol{u}) \tag{41}$$

which depends on the array of vertex coordinates $\boldsymbol{x}$ and displacements $\boldsymbol{u}$. It is clear that the displacements have to satisfy (40). The compliance cost function corresponding to the discrete equilibrium equation reads

$$J(\boldsymbol{x},\boldsymbol{u}) = \boldsymbol{f}^{\mathsf{T}}\boldsymbol{u} = \boldsymbol{u}^{\mathsf{T}}\boldsymbol{K}\boldsymbol{u}\,. \tag{42}$$

In order to compute the derivatives of the cost function with respect to the vertex coordinates we consider the adjoint formulation

$$L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda}) = J(\boldsymbol{x},\boldsymbol{u}) + \boldsymbol{\lambda}^{\mathsf{T}}[\boldsymbol{f} - \boldsymbol{K}\boldsymbol{u}], \tag{43}$$

where $\boldsymbol{\lambda}$ is a Lagrange parameter vector. The stationarity condition for $L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda})$ with respect to the vertex displacements leads to the adjoint problem

$$\boldsymbol{K}\boldsymbol{\lambda} = \frac{\partial J(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\,. \tag{44}$$

Here, we made use of the symmetry of the stiffness matrix $\boldsymbol{K}$. The stationarity condition for $L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda})$ with respect to the vertex coordinates leads to the discrete shape gradients

$$\frac{\partial L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda})}{\partial \boldsymbol{x}} = \frac{\partial J(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}} + \boldsymbol{\lambda}^{\mathsf{T}}\left[\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} - \frac{\partial \boldsymbol{K}}{\partial \boldsymbol{x}}\boldsymbol{u}\right]. \tag{45}$$

By introducing the cost function (42) and the adjoint solution with $\boldsymbol{\lambda} = \boldsymbol{u}$ we obtain

$$\frac{\partial L(\boldsymbol{x},\boldsymbol{u},\boldsymbol{\lambda})}{\partial \boldsymbol{x}} = 2\boldsymbol{u}^{\mathsf{T}}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} - \boldsymbol{u}^{\mathsf{T}}\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{x}}\boldsymbol{u}\,. \tag{46}$$

The derivatives of the stiffness matrix with respect to the vertex coordinates can be determined by consecutively differentiating the discretised Kirchhoff-Love energy functional (36) first with respect to vertex displacements and then with respect to the vertex positions. In gradient-based optimisation, in order achieve a decrease in the cost function the perturbation of the control vertex positions is chosen proportional to

$$\boldsymbol{V} = -2\boldsymbol{u}^{\mathsf{T}}\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} + \boldsymbol{u}^{\mathsf{T}}\frac{\partial \boldsymbol{K}}{\partial \boldsymbol{x}}\boldsymbol{u}\,. \tag{47}$$

Note that, with a slight abuse of notation, the perturbation vectors for the continuous and discrete shape sensitivity formulations are denoted with the same symbol $\boldsymbol{V}$. For more details we refer to [2].

4 Multiresolution Shape Optimisation

The introduced subdivision multiresolution editing technique enables to use two different resolutions of the same geometry for optimisation and analysis. A simplified two-level version of the proposed optimisation algorithm is shown Fig. 11. The optimisation and analysis meshes correspond to different refinement levels in a multiresolution hierarchy. The optimisation level is $\ell_o = 0$ and the analysis level is $\ell_c = n$. In our actual implementation the optimisation level is successively increased until $\ell_o = \ell_c$ and the analysis level ℓ_c is fixed. Crucially, in the spirit of isogeometric analysis the control meshes for analysis and optimisation represent the same geometry.

In shape optimisation it is usually necessary to use a coarse control mesh for geometry updating and a relatively fine control mesh for analysis. As is known, unwanted geometry oscillations may appear when the analysis and geometry representations have similar resolutions [8]. These geometry oscillations are usually a numerical artefact or a result of the ill-posedness of the considered optimisation problem. Moreover, in practical applications it might be desirable to optimise only a very coarse representation out of aesthetic or manufacturability reasons.

A more detailed description of the employed multiresolution technique is given in Algorithm 1. In the presented computations, different from Algorithm 1 we use instead of the steepest descent update on line 9 the method of moving asymptotes (MMA) as implemented in the NLopt library [20]. In addition, in practical computations there are usually constraints, such as bounds on vertex positions or area/volume constraints, which are not mentioned in Algorithm 1. For more details we refer to [3, 4].

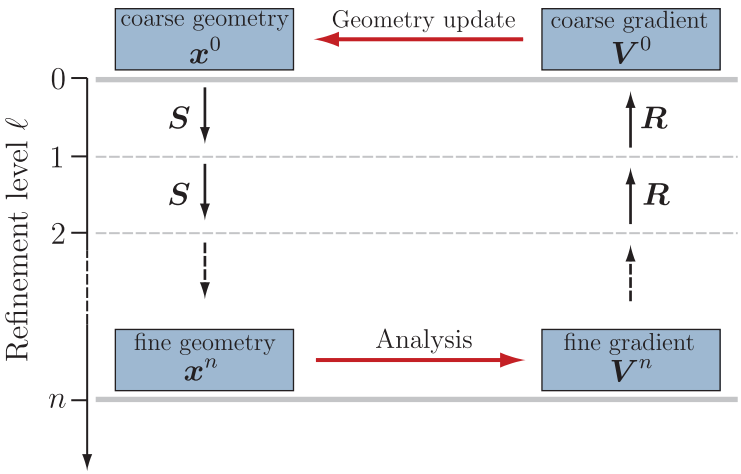

Fig. 11 Flowchart of the simplified multiresolution optimisation algorithm

Algorithm 1 Multiresolution shape optimisation

// choose computational level ℓ_c
// read coarse input control mesh $\boldsymbol{x}^0$
// Initialise optimisation level
$\ell_o = 0$
// Initialise cost functional
$J = \infty$
// iterate over optimisation levels
while $\ell_o \leq \ell_c$ **do**
  // update vertex coordinates $\boldsymbol{x}^{\ell_o}$ *while the cost functional decreases*
  repeat
    // subdivide optimisation level ℓ_o *up to analysis level* ℓ_c
    for $\ell \leftarrow \ell_o$ to ℓ_c **do**
      $\boldsymbol{x}^{\ell} \leftarrow \boldsymbol{S}\boldsymbol{x}^{\ell}$
    // compute cost functional $J = J(\boldsymbol{x}^{\ell_c}, \boldsymbol{u}(\boldsymbol{x}^{\ell_c}))$ *and the descent direction* $\boldsymbol{V}^{\ell_c}$
    // project the descent direction $\boldsymbol{V}^{\ell_c}$ *to the optimisation level*
    for $\ell \leftarrow \ell_c$ to ℓ_o **do**
      $\boldsymbol{V}^{\ell} \leftarrow \boldsymbol{R}\boldsymbol{V}^{\ell}$
    // update vertex coordinates of the optimisation level
    $\boldsymbol{x}^{\ell_o} \leftarrow (\boldsymbol{x}^{\ell_o} + \alpha \boldsymbol{V}^{\ell_o})$ with $\alpha > 0$
  until $J < J_{\text{previous}}$
  // increment optimisation level
  $\ell_o \leftarrow (\ell_o + 1)$

5 Examples

5.1 Immersed Finite Element Discretised Elasticity Problems

5.1.1 Simply Supported Plate with a Hole

This example highlights the advantages of multiresolution optimisation over classical approaches that use only one or two representation levels. The problem consists of a square plate with an edge length $L = 2$ and a circular hole with diameter $D = 1$, see Fig. 12. The plate is loaded with a line load of length 1. The Young's modulus and Poisson's ratio of the plate are $E = 100$ and $\nu = 0.4$, respectively. During optimisation the shape of the hole is to be modified so that the structural compliance of the plate is minimised.

Initially, at level $\ell = 0$ the hole is represented with a cubic spline with 8 control points. The immersed finite element grid has 100×100 cells of uniform size. Three cases referred to as C1, C2 and C3 with different geometry and analysis resolutions are studied:

- In C1 only one level with $\ell_o = \ell_c = 0$ is used for analysis and optimisation.
- In C2 a four times subdivided control mesh at refinement level $\ell_o = \ell_c = 4$ is used for analysis and optimisation.

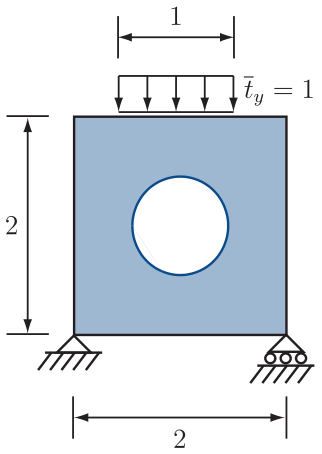

Fig. 12 Simply supported plate with a hole. Problem description

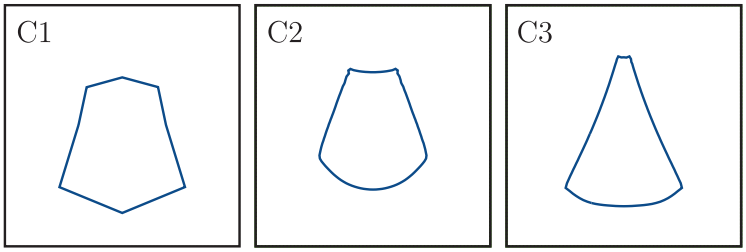

Fig. 13 Simply supported plate with a hole. Optimised hole shapes for cases C1, C2 and C3

– In C3 the optimisation level starts with $\ell_o = 0$ and increases until $\ell_o = \ell_c = 4$ is reached. Throughout the computations the analysis level is fixed to $\ell_c = 4$.

In case C1 the control mesh that is visible by the immersed finite element grid contains 8 elements and in cases C2 and C3 it contains 128 elements. It is clear that in case C1 the hole geometry is poorly resolved on the immersed finite element grid.

In Fig. 13 the optimised final hole shapes for the three cases are shown. In particular, the difference in optimal shapes for cases C2 and C3 which use the same analysis level $\ell_c = 4$ is striking. The case C1 is different from the other two cases because of the mentioned inadequately coarse analysis control mesh with $\ell_c = 0$. During optimisation the optimisation level ℓ_o is successively increased only for case C3. The optimisation level is always incremented when a minimum is reached, cf. Algorithm 1. For the three cases the reduction of the relative cost function over the number of iterations is shown in Fig. 14. The case C2 with fixed fine resolution achieves the smallest cost reduction while the case C3 with multiresolution achieves the largest cost reduction. The strong dependence of the optimisation results on geometry parameterisation is well known in structural optimisation and is often

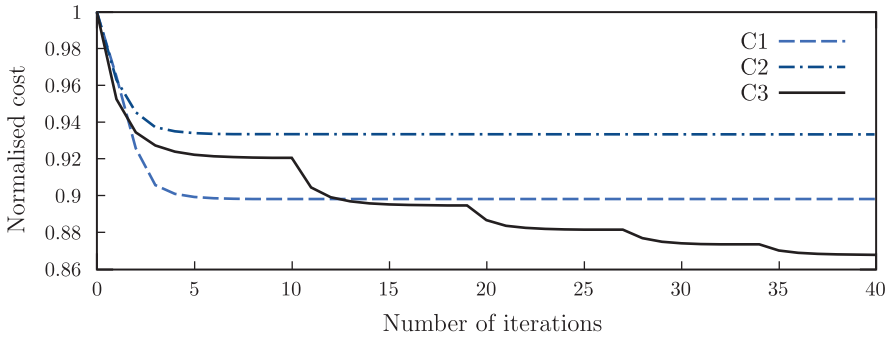

Fig. 14 Simply supported plate with a hole. Reduction of the normalised cost over the number of optimisation iterations. The initial cost for case C1 is 0.073 and for cases C2 and C3 is 0.065

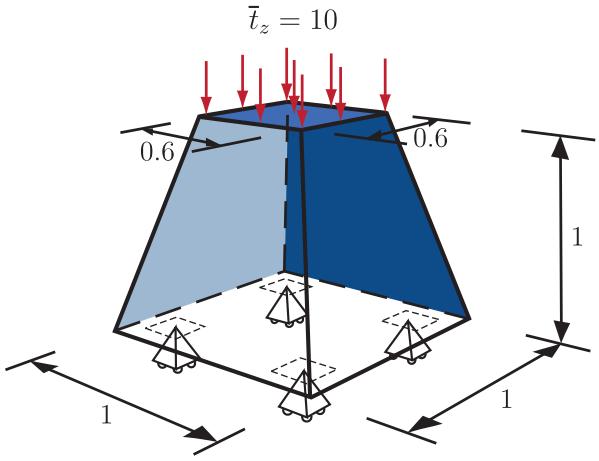

Fig. 15 Three-dimensional stool. Problem description. Roller supports are applied to all finite element nodes inside the regions of size 0.2×0.2 marked by *dashed squares*

associated with the non-convexity of the considered optimisation problem. We conjecture that by initially using a coarse control mesh for optimisation the possible number of local minima is significantly reduced which reduces the possibility of landing in a non-optimal local minimum. It appears that in case C2 the optimisation problem is caught in a local minimum which is significantly higher than the global minimum.

5.1.2 Three-Dimensional Stool

In this example we present the combined topology and shape optimisation of a three-dimensional solid, see Fig. 15. Our algorithm for combined topology and shape

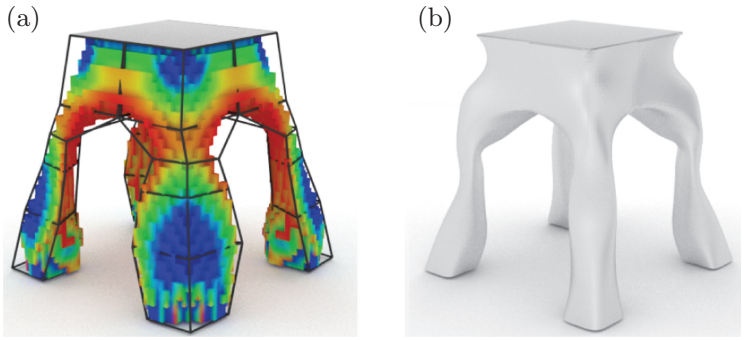

Fig. 16 Three-dimensional stool. First topology and subsequent shape optimisation step. (**a**) Topology optimised. (**b**) Shape optimised

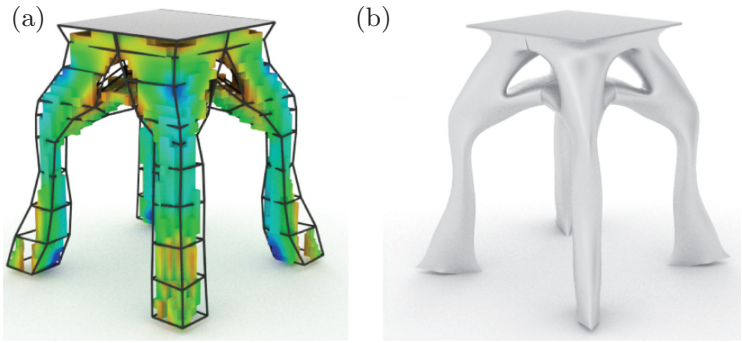

Fig. 17 Three-dimensional stool. Second topology and subsequent shape optimisation step. (**a**) Topology optimised. (**b**) Shape optimised

optimisation is described in [3]. The initial domain is a truncated pyramid and is at its top loaded with a uniform distributed load $\bar{t}_z = 10$. At its bottom it is supported by four distributed roller supports each of size 0.2×0.2. The Young's modulus and Poisson's ratio are chosen with $E = 100$ and $\nu = 0.4$, respectively. In the optimisation study only one quarter of the domain is considered and appropriate bounds and geometry tags are applied at the two planes of symmetry. The corresponding immersed finite element grid is of size $0.7 \times 0.7 \times 1$ and consists of $30 \times 30 \times 30$ cells.

The sequence of the performed topology and shape optimisation steps are shown in Figs. 16 and 17. In total two topology and two shape optimisation steps are performed. During each topology optimisation step we remove in one go a relatively large amount of material by deleting computational cells with the topology derivative below a given threshold. Figure 16a shows the result of the first topology optimisation step and the semi-manually generated coarse control mesh for representing the new topology. In the following shape optimisation step, see Fig. 16b, the generated control mesh serves as the optimisation level $\ell_o = 0$ and

the computation level is chosen with $\ell_c = 2$. During the shape optimisation the volume of the domain is constraint to remain constant. The result of the second topology optimisation step and the generated control mesh are shown in Fig. 17a. This is followed by the final shape optimisation step shown Fig. 17b.

5.2 Boundary Element Discretised Electrostatic Problems

5.2.1 Box in a Sphere

We optimise the shape of a box placed inside a sphere, see Fig. 18, with the expected normal flux density Q in (30) 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 [15]. The box, representing the to be optimised boundary Γ_{D_f}, is of size $0.16 \times 0.2 \times 0.24$ and the outer sphere, representing the fixed boundary Γ_{D_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 Sect. 2.2, see also [5, 10]. With the extended subdivision stencils the limit surface corresponding to the coarse box mesh is a box of the same geometry. 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 18a 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., $\ell_o = 0$. After consecutively selecting $\ell_o = 1$ and $\ell_o = 2$ and optimising we obtain the final optimised geometry shown in Fig. 18b. The 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 %. As to be

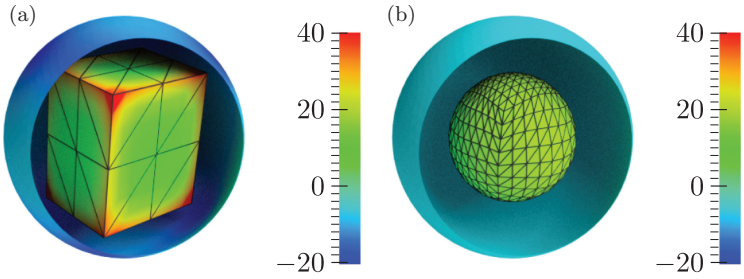

Fig. 18 Box in a sphere. Initial and optimised geometries with isocontours of the normal flux. The meshes indicate the optimisation level ℓ_o. The isocontours belong to the fine computational mesh at level $\ell_c = 2$. (**a**) Initial geometry at $\ell_o = 0$. (**b**) Final optimised geometry at $\ell_o = 2$

expected, the optimisation leads to a geometry with nearly uniform distribution of normal flux as seen in Fig. 18b. Since the extended subdivision scheme was used in this case, the marked creases were not smoothed out by the subdivision itself, but rather by the shape optimisation procedure.

5.2.2 Gas Insulated Switchgear

In this example we apply the proposed shape optimisation approach to the design of an electrode in a gas insulated switchgear, see Fig. 19a. Such devices are widely used as circuit-breakers in high-voltage power transmission. The objective of shape optimisation is to reduce the propensity for electrical breakdown with the ultimate aim of enabling more compact device geometries. This can be achieved by modifying the electrode geometries such that the cost functional $J(\Omega^{\ell_c}, u^{\ell_c})$ in (30) is minimised.

In Fig. 19a the gas insulated switchgear is shown with the electrode in the form of a primitive cylinder. The cylinder represents the electrode geometry 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, see Fig. 19b. As a design constraint, the inner surface of the cylinder is required to have a constant radius for a bolt passing through it. Geometric constraints on the positions of vertices lying on the inner surface are applied to prevent any movement that would violate this design requirement.

In this example, a once subdivided electrode mesh with 1056 elements is chosen as the computational level ($\ell_c = 1$) and the geometry at level $\ell_o = 0$ is used for optimisation. In the initial design, Fig. 19b, the maximum normal flux is 81.63 before optimisation and reduces to 66.99 in the optimised shape shown in Fig. 20, 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 Fig. 20b two electrode

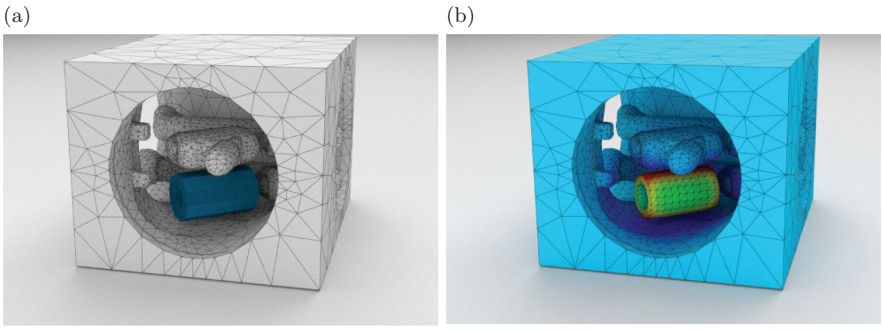

Fig. 19 Gas insulated switchgear with coarse and refined electrode meshes. The considered electrode is shown in *dark blue* in (**a**). The isocontours in (**b**) represent the normal flux

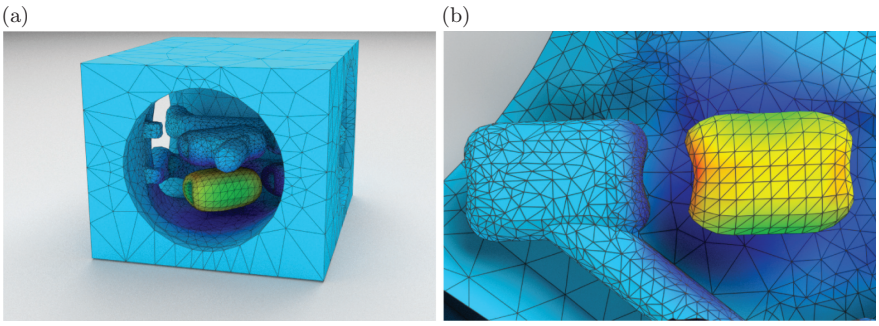

Fig. 20 Gas insulated switchgear with the optimised electrode geometry. The isocontours represent the normal flux. (**a**) Isometric view. (**b**) Top view

geometries are visible. The one in blue is an electrode geometry that has been obtained over the years by combining engineering intuition with simple calculations and testing. The similarities between the systematically shape optimised and the electrode geometry in production are striking. Notice in particular the saddle shapes at the two ends of the both electrodes, which help to reduce the large normal fluxes that are present at the sharp edges of the inner hole.

5.3 Subdivision Finite Element Discretised Thin-Shell Problems

5.3.1 Inverted Catenary Arch

In this verification example we optimise a thin strip pinned at both ends and subjected to a vertical distributed load. The optimal shape for such a strip is (in the limit of zero thickness) a catenary because for shells it is energetically more favourable to carry loads through membrane action instead of bending action. The catenary is the shape assumed by a loose string hung freely from two fixed points [24], its equation reads

$$y = \alpha \cosh\left(\frac{x+\beta}{\alpha}\right) + \gamma\,, \tag{48}$$

where the y-axis is parallel to the applied load vector and the three constants α, β and γ are determined based on the location of the supports and the string length l. Two cases are studied, one where the supports are at equal height and the other where one support is higher than the other. In both cases the horizontal distance between the supports is 1. The vertical offset between the supports in the second case is 0.2. Initially, the strip is a narrow flat plate connecting the supports with length equal to

the distance between the supports. Width and thickness are constant with 0.05 and 0.02, respectively. The magnitude of the vertical uniformly distributed load is 1000, the Young's modulus and Poisson's ratio are $E = 2 \times 10^8$ and $\nu = 0.3$, respectively. The length l of the optimised strip is chosen with 1.1 and 1.3.

The Catmull-Clark subdivision scheme is used for geometry representation and for discretising the thin-shell equations. The initial coarse mesh used for optimisation contains only 3 elements along the length and 1 element across the width of the strip. This increases to 48 in the twice subdivided fine mesh used for analysis. During optimisation the mesh resolution is increased starting $\ell_o = 0$ up to $\ell_o = 2$. Only the y coordinates of the control points are optimised. The comparison of the optimisation results with the corresponding catenary curve for different curve lengths and support positions is shown in Fig. 21. The reduction of the objective function is more than 99.9 % for all cases and the results show good visual agreement with catenary curves. The slight deviations from the catenary are possibly due to the finite width of the strip, which leads during optimisation to some curvature generation across the width of the shell (not visible in Fig. 21).

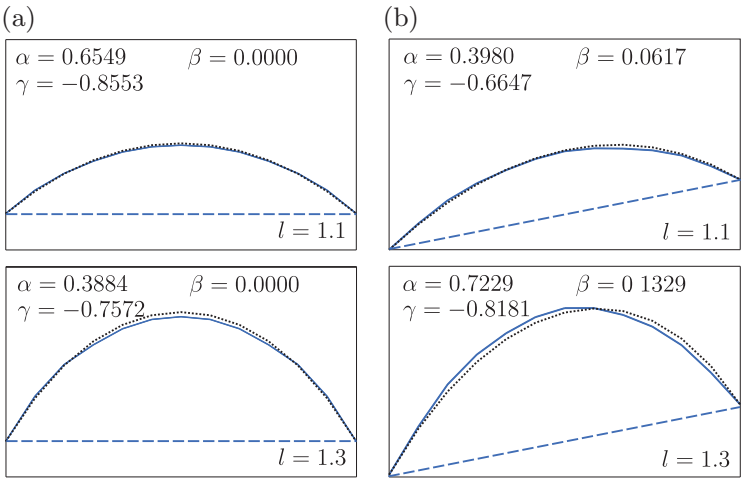

Fig. 21 Inverted catenary arch. The *dashed* and *solid blue lines* show the centre lines of the strip before and after optimisation, respectively. The *dotted black line* is the catenary curve corresponding to length l. The α, β and γ values are the parameters in (48). In the chosen coordinate system the left support has the coordinates $(-0.5, 0.0)$. (**a**) Supports at same height. (**b**) Supports at different heights

5.3.2 Architectural Roof

We consider the optimisation of an initial roof design shown in Fig. 22 with approximate dimensions of $2.31 \times 6.27 \times 0.75$. This design was created by an architect using the Autodesk Maya software. A vertical uniformly distributed load of 1000 is chosen as the design load and the shell thickness is $t = 0.02$. The Young's modulus and the Poisson's ratio are $E = 1 \times 10^{10}$ and $\nu = 0.2$, respectively.

During the shape optimisation only the vertical coordinates of control points are chosen as design variables. Moreover, an area constraint is applied in order to restrict the size of the optimised shape. An important architectural feature of the roof is the ridge profile, which is preserved by applying special extended subdivision stencils at the corner vertices and crease edges, see Figs. 22 and 23.

The coarse mesh at optimisation level $\ell_o = 0$ contains 26 nodes which is twice subdivided to obtain the computation mesh at level $\ell_c = 2$ with 272 nodes. The second optimisation stage is done at level $\ell_o = 1$ but no optimisation is done at level $\ell_o = 2$ as this results in oscillatory surface features. The optimised roof design is shown in Fig. 24. The initial value of the compliance cost function is 31.36 and is reduced by 79.13 % through optimisation.

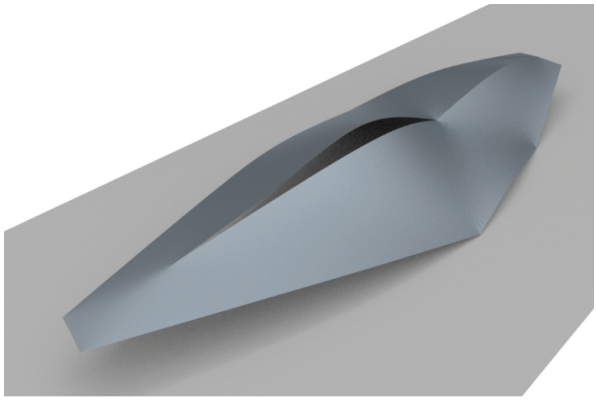

Fig. 22 Limit surface of the initial architectural roof design. The model has three supports, a central opening and a roof ridge. The corresponding coarse control mesh and selected vertex and edge tags are shown in Fig. 23

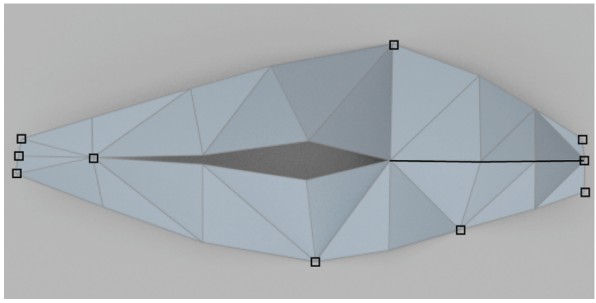

Fig. 23 Top view of the initial control mesh of the architectural roof. The nodes indicated by *black squares* are tagged as corner and the edges highlighted by the *black solid line* as crease

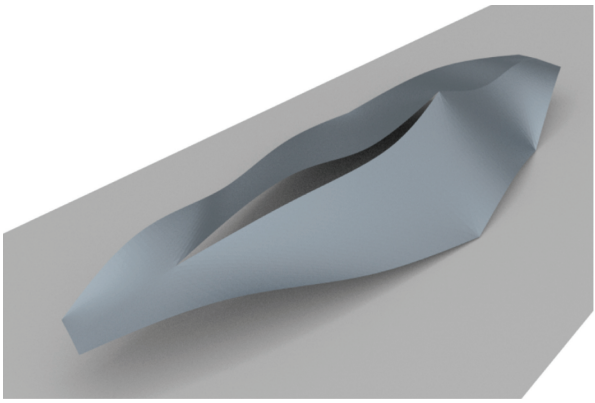

Fig. 24 Limit surface of the optimised architectural roof design

6 Summary and Conclusions

We reviewed the multiresolution optimisation technique based on subdivision surfaces and presented its application to elasticity, electrostatics and thin-shell problems. For discretising the governing partial differential equations we employed three different discretisation techniques, namely immersed finite elements for elasticity, boundary elements for electrostatics and subdivision finite elements for thin-shells. The inherent hierarchy of the subdivision surfaces allows us 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 always a fine mesh is used for analysis. As demonstrated with the computed examples, this effectively inhibits the appearance of 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 boundary mesh during the optimisation. Finally, it is evident that the multiresolution editing semantics is particularly appealing for isogeometric analysis because it enables the decoupling of geometry and analysis representations.

Acknowledgements The partial support of the EPSRC through grant # EP/G008531/1 and EC through the Marie Curie Actions (IAPP) program CASOPT project are gratefully acknowledged.

References

1. G. Allaire, F. Jouve, A.M. Toader, Structural optimization using sensitivity analysis and a level-set method. J. Comput. Phys. **194**, 363–393 (2004)
2. K. Bandara, F. Cirak, Multiresolution shape optimisation of thin-shell structures with subdivision finite elements. Technical report, University of Cambridge (2015)

3. K. Bandara, T. Rüberg, F. Cirak, Shape optimisation with multiresolution subdivision surfaces and immersed finite elements. arXiv:1510.02719 (2015)
4. K. Bandara, F. Cirak, G. Of, O. Steinbach, J. Zapletal, Boundary element based multiresolution shape optimisation in electrostatics. J. Comput. Phys. **297**, 584–598 (2015)
5. H. Biermann, A. Levin, D. Zorin, Piecewise smooth subdivision surfaces with normal control, in *SIGGRAPH 2000 Conference Proceedings*, New Orleans, 2000, pp. 113–120
6. K.U. Bletzinger, S. Kimmich, E. Ramm, Efficient modeling in shape optimal design. Comput. Syst. Eng. **2**, 483–495 (1991)
7. P.B. Bornemann, F. Cirak, A subdivision-based implementation of the hierarchical b-spline finite element method. Comput. Methods Appl. Mech. Eng. **253**, 584–598 (2013)
8. V. Braibant, C. Fleury, Shape optimal design using b-splines. Comput. Methods Appl. Mech. Eng. **44**, 247–267 (1984)
9. E. Catmull, J. Clark, Recursively generated b-spline surfaces on arbitrary topological meshes. Comput.-Aided Des. **10**, 350–355 (1978)
10. F. Cirak, Q. Long, Subdivision shells with exact boundary control and non-manifold geometry. Int. J. Numer. Methods Eng. **88**, 897–923 (2011)
11. F. Cirak, M. Ortiz, P. Schröder, Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. Int. J. Numer. Methods Eng. **47**, 2039–2072 (2000)
12. F. Cirak, M. Scott, E. Antonsson, M. Ortiz, P. Schröder, Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. Comput.-Aided Des. **34**, 137–148 (2002)
13. M. Delfour, J.P. Zolésio, *Shapes and Geometries: Analysis, Differential Calculus, and Optimization*. Advances in Design and Control (SIAM, Philadelphia, 2001)
14. D. Doo, M. Sabin, Behavior of recursive division surfaces near extraordinary points. Comput.-Aided Des. **10**, 356–360 (1978)
15. M. Flucher, M. Rumpf, Bernoulli's free-boundary problem, qualitative theory and numerical approximation. Journal für die reine und angewandte Mathematik **486**, 165–204 (1997)
16. D. Fußeder, B. Simeon, A.V. Vuong, Fundamental aspects of shape optimization in the context of isogeometric analysis. Comput. Methods Appl. Mech. Eng. **286**, 313–331 (2015)
17. R.T. Haftka, R.V. Grandhi, Structural shape optimization – a survey. Comput. Methods Appl. Mech. Eng. **57**, 91–106 (1986)
18. J. Haslinger, R.A.E. Mäkinen, *Introduction to Shape Optimization: Theory, Approximation, and Computation* (SIAM, Philadelphia, 2003)
19. T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**, 4135–4195 (2005)
20. S.G. Johnson, The NLopt nonlinear-optimization package. http://ab-initio.mit.edu/nlopt
21. J. Kiendl, R. Schmidt, R. Wüchner, K.U. Bletzinger, Isogeometric shape optimization of shells using semi-analytical sensitivity analysis and sensitivity weighting. Comput. Methods Appl. Mech. Eng. **274**, 148–167 (2014)
22. J.M. Lane, R.F. Riesenfeld, A theoretical development for the computer generation and display of piecewise polynomial surfaces. IEEE Trans. Pattern Anal. Mach. Intell. **PAMI-2**, 35–46 (1980)
23. N. Litke, A. Levin, P. Schröder, Fitting subdivision surfaces, in *IEEE Visualization 2001*, San Diego, 2001, pp. 319–324
24. E.H. Lockwood, *A Book of Curves* (Cambridge University Press, Cambridge, 1961)
25. C. Loop, Smooth subdivision surfaces based on triangles. Master's thesis, Department of Mathematics, University of Utah (1987)
26. M. Lounsbery, T.D. DeRose, J. Warren, Multiresolution analysis for surfaces of arbitrary topological type. ACM Trans. Graph. **16**, 34–73 (1997)
27. J. Peters, U. Reif, *Subdivision Surfaces*. Springer Series in Geometry and Computing (Springer, Berlin/Heidelberg, 2008)
28. T. Rüberg, F. Cirak, An immersed finite element method with integral equation correction. Int. J. Numer. Methods Eng. **86**, 93–114 (2011)

29. T. Rüberg, F. Cirak, Subdivision-stabilised immersed b-spline finite elements for moving boundary flows. Comput. Methods Appl. Mech. Eng. **209–212**, 266–283 (2012)
30. T. Rüberg, F. Cirak, A fixed-grid b-spline finite element technique for fluid–structure interaction. Int. J. Numer. Methods Fluids **74**(9), 623–660 (2014)
31. R. Sanches, P. Bornemann, F. Cirak, Immersed b-spline (i-spline) finite element method for geometrically complex domains. Comput. Methods Appl. Mech. Eng. **200**, 1432–1445 (2011)
32. D. Schillinger, L. Dede, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, T.J.R. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. Comput. Methods Appl. Mech. Eng. **249–253**, 116–150 (2012)
33. J. Sokolowski, J.P. Zolesio, *Introduction to Shape Optimization: Shape Sensitivity Analysis* (Springer, Berlin/Heidelberg, 1992)
34. J. Stam, Exact evaluation of Catmull-Clark subdivision surfaces at arbitrary parameter values, in *SIGGRAPH 1998 Conference Proceedings*, Orlando, 1998, pp. 395–404
35. J. Stam, Evaluation of Loop subdivision surfaces, in *SIGGRAPH 1999 Course Notes*, Los Angeles (1999)
36. O. Steinbach, *Numerical Approximation Methods for Elliptic Boundary Value Problems. Finite and Boundary Elements* (Springer, New York, 2008)
37. W. Wall, M. Frenzel, C. Cyron, Isogeometric structural shape optimizisation. Comput. Methods Appl. Mech. Eng. **197**, 2976–2988 (2008)
38. J. Warren, H. Weimer, *Subdivision Methods for Geometric Design: A Constructive Approach* (Morgan Kaufmann, San Francisco, 2001)
39. L. Ying, D. Zorin, Nonmanifold subdivision, in *IEEE Visualization 2001*, San Diego, 2001, pp. 325–332
40. D. Zorin, P. Schröder, Subdivision for modeling and animation, in *SIGGRAPH 2000 Course Notes*, New Orleans (2000)
41. D. Zorin, P. Schröder, W. Sweldens, Interactive multiresolution mesh editing, in *Proceedings of the 24th annual conference on computer graphics and interactive techniques (SIGGRAPH)*, Los Angeles, (ACM Press/Addison-Wesley, New York, 1996), pp. 259–268

http://www.springer.com/978-3-319-23314-7

Isogeometric Analysis and Applications 2014
Jüttler, B.; Simeon, B. (Eds.)
2015, XIV, 290 p. 127 illus., 15 illus. in color., Hardcover
ISBN: 978-3-319-23314-7