Shape optimization directly from CAD: an isogeometric boundary element approach using T-splines

H. Lian[a,b,c], P. Kerfriden[c], S.P.A. Bordas[d,c,e,f,*]

[a] *College of Mining Engineering, Taiyuan University of Technology, Taiyuan, Shanxi, China*
[b] *Key Laboratory of In-situ Property-Improving Mining of Ministry of Education, Taiyuan, Shanxi, China*
[c] *School of Engineering, Cardiff University, The Parade, CF24 3AA, Cardiff, UK*
[d] *Faculté des Sciences, de La Technologie, Luxembourg University, 6, rue Richard Coudenhove-Kalergi, L-1359 Luxembourg*
[e] *Research Unit in Engineering Science Campus Kirchberg, G 007, Luxembourg*
[f] *University of Western Australia, 35 Stirling Hwy, Crawley, WA 6009, Australia*

Abstract

We develop a T-spline isogeometric boundary element method (IGABEM) [1, 2, 3] to shape sensitivity analysis and gradient-based shape optimization in three dimensional linear elasticity. Contrary to finite element based isogeometric analysis (IGA) approaches, no parametrization of the volume is required. Hence, the iterative optimization algorithm can be implemented directly from CAD without any mesh generation nor postprocessing step for returning the resulting structure to CAD designers. T-splines also guarantee a water-tight geometry without the manual geometrical-repair work as with non-uniform rational B-splines (NURBS). We demonstrate the worth of the method by analysing problems with and without analytical solutions, including engineering examples involving complex shapes. Additionally, we provide all the derivations of the required sensitivities and the details pertaining to the geometries examined in the benchmarking, to provide helpful reference problems for 3D shape optimization.

Keywords: shape optimization, isogeometric boundary element methods, T-splines, shape sensitivities, 3D design, CAD

*Corresponding author
Email address: stephane.bordas@uni.lu (S.P.A. Bordas)

1. Introduction

Shape optimization is a critical step in engineering design to obtain the optimal shape of a component under given objectives and constraints. In this article, we refer to shape optimization as a process only involving the variation of the component boundaries, which distinguishes itself from topology optimization [4, 5, 6] in that the latter requires nucleation of holes and in which the construction of the design space is closely related to material parameters. The wide application of automatic shape optimization in industry is still elusive. One reason for this is that traditional numerical methods, such as the finite element method (FEM) [7] and the boundary element method (BEM) [8, 9, 10, 11, 12], are based on an approximate geometric representation distinct from the computer-aided design (CAD) model, so that a mesh generation/regeneration procedure, which occupies around 80% of the total problem solving time for linear problems at each iterative step, requires significant human intervention, thereby hindering the automation of the process. To avoid cumbersome meshing procedures, the meshfree (meshless) methods [13, 14, 15, 16, 17] use a set of nodes associated with a domain in the construction of the approximation. The application of the meshfree methods in shape optimization can be found in [18, 19, 20]. Nonetheless, it is not easy to choose appropriate node locations and maintain geometric accuracy in the meshfree methods. The extended finite element methods (XFEM) alleviate meshing burden [21, 22, 23, 24, 25, 26] by separating the FEM mesh and the geometry representation. Its application in shape optimization can be read in [27, 28, 29]. Due to the implicit representation of the geometry, the capture of the geometry boundary for domain integration is a challenging task, especially in three dimensional problems.

A promising direction in shape optimization is isogeometric analysis (IGA) proposed in [30] (see the recent review and computer implementation aspects in [31]), which was proposed to integrate the geometry and analysis representations. This is achieved by using the data provided by CAD models *directly* rather than converting it through a preprocessing routine into a form suitable

for analysis (*e.g.* a mesh). The resulting benefit is that the meshing procedure is bypassed and the exact geometry can also be preserved. IGA has been applied into numerous areas successfully, such as the finite deformation [32], plates and shells [33, 34], structural vibration analysis [35], contact mechanics [36, 37], fluid-structure interaction [38], electromagnetics [39], *etc.* The advantage of IGA in shape optimization [40, 41, 42, 43, 44, 45] is specially manifest because shape optimization requires a continuous communication between the CAD and analysis. However, several shortcomings of IGA are still present, in the original form proposed in [30]: 1) Local h-adaptivity is difficult. 2) Solids generally cannot be represented by a single non-uniform rational B-spline (NURBS) patch but continuity between patches is not naturally provided by NURBS functions, thereby creating spurious jumps along the interfaces. [46, 47] provide an approach to overcome this difficulty through Nitsche's coupling. 3) CAD provides only boundary information but numerical methods require in general the parameterization of the interior of the domain. The advances in this direction can be read in the work of [48, 49, 50, 51]. However, there is still a lack of an efficient method for general geometries.

The first two difficulties can be alleviated by the advance in CAD techniques. For example, T-splines were proposed by Sederberg *et al.* [52], and also applied in IGA by Bazilevs *et al.* [53]. Adaptive analysis by local h-refinement with T-splines was given by Dorfel *et al.* [54], and a large deformation frictionless contact problem was addressed by [55]. Other alternatives include PHT-splines [56, 57, 58], hierarchical splines [59, 60] and Locally Refined splines [61]. The main advantage of T-splines over other alternatives is that it is more flexible to subdivide the geometry and control the order of the geometry continuity.

The third difficulty associated to the domain parametrization can be naturally overcome by the isogeometric boundary element methods (IGABEM) [62, 1, 63, 64, 2, 3, 65, 66, 67]. The idea of IGABEM is to employ the basis/blending functions in CAD to discretize the boundary integral equation (BIE). The equation system is obtained by using a collocation scheme instead of the Galerkin approach, similar to the isogeometric collocation method [68, 69, 70], for the

purpose of efficiency. IGABEM can achieve a close integration of CAD and analysis. Recently, T-splines were incorporated into IGABEM for 3D linear elastostatic analysis [2] and acoustic analysis [3] on complex geometries. Particularly, [2, 3] adopted Bézier extraction techniques, which can lead to a familiar element assembly procedure as in traditional BEM.

The advantage of the tight integration of analysis and CAD renders the IGABEM immediate advantages in shape optimization. In [71], the shape optimization in three dimensional linear elasticity was conducted by the IGABEM with NURBS. [72] incorporated T-splines to IGABEM for shape-hull optimization in hydrodynamic problems combined with a gradient-less optimization algorithm. An IGABEM scheme with subdivision surfaces was also proposed in [73] for the shape optimization in electrostatics. The present paper employs IGABEM with T-splines to structural shape sensitivity analysis and optimization in linear elasticity. To the authors' best knowledge, it is also the first time that T-spline based IGABEM is used for gradient-based shape optimization, which has more solid mathematical foundation compared with various heuristic or gradient-less optimization algorithms.

In the aspect of the IGABEM implementation, a main difficulty is addressing strongly singular integral and jump terms, as explained in [1]. To resolve this difficulty, a regularised form of boundary integral equation [74, 75, 76] was discretized by CAD basis functions or blending functions [2, 3], without needing to evaluate strongly singular integrals and jump terms. By noticing that the fundamental solutions are the same order as their shape derivatives, the regularised form can be still available for sensitivity analysis. Hence, our work takes the shape differentiation on the regularized boundary integral equation, simplifying the implementation greatly.

The paper is organized as follows. T-splines and the Bézier extraction are reviewed in Section 2. Section 3 presents the formulation of a regularized IGABEM with T-splines. Shape sensitivity analysis with IGABEM is introduced in Section 4. Section 5 illustrates the IGABEM shape optimization, followed by numerical examples in Section 6. Section 7 contains conclusions and future

work.

2. T-splines and Bézier extraction

2.1. T-mesh

A T-spline control mesh, or called T-mesh, is distinct from a NURBS control mesh in that it has T-junctions, which are similar to the concept of "hanging nodes" and oct/quad-tree meshes in the FEM. See Fig. 1. If a T-mesh is simply a rectangular grid with no T-junctions, T-splines reduce to B-splines.

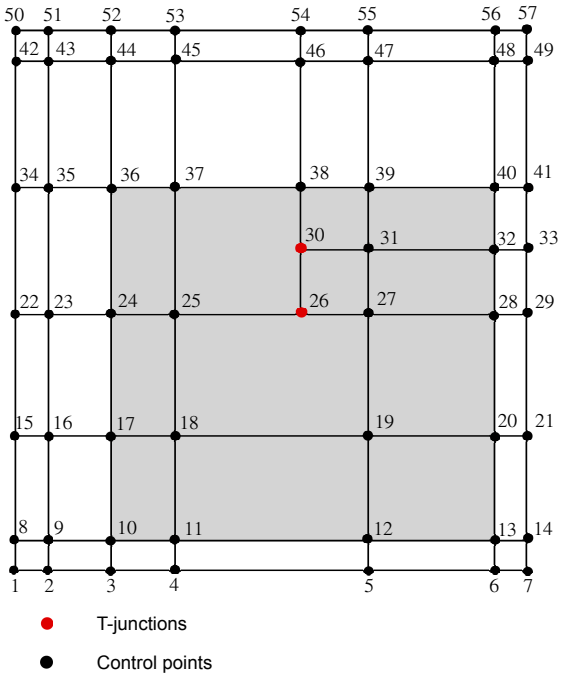


Figure 1: T-mesh and T-junctions

Instead of sharing a global knot vector like NURBS, each control point in a T-mesh is associated with a local knot vector in each dimension

$$\Xi_A = [\xi_1, \xi_2, \cdots, \xi_{p+1}], \tag{1}$$

where A is the global index of the control point in the T-mesh, ξ the knot in local knot vector, and p the order of T-splines. The local knot vector in multiple

directions can be collected as

$$\boldsymbol{\Xi}_A = \{\Xi_A^d\}_{d=1}^{n_d}, \tag{2}$$

where d denotes the direction index, and n_d the dimension of the geometry.

Now we define a knot interval vector as

$$\Delta\Xi_A = [\Delta\xi_1, \Delta\xi_2, \cdots, \Delta\xi_{p+1}], \tag{3}$$

and its vector form in multiple directions is

$$\boldsymbol{\Delta\Xi}_A = \{\Delta\Xi_A^d\}_{d=1}^{n_d}. \tag{4}$$

In practice, the local knot vector is obtained from a local knot interval vector which can be deducted from a predefined knot interval configuration on the T-mesh. See [77] for details. The local knot vector leads to a local parametric space, on which T-spline blending functions on the dth dimension can be formulated using an iterative formula.

For $p = 0$,

$$N_A^d(\xi|\xi_1, \xi_2) = \begin{cases} 1 & \text{if } \xi_1 \leqslant \xi < \xi_2, \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

and for $p > 0$,

$$\begin{aligned} N_A^d(\xi|\xi_1, \xi_2, \cdots, \xi_{p+2}) = &\frac{\xi - \xi_1}{\xi_{p+1} - \xi_1} N_A^d(\xi|\xi_1, \xi_2, \cdots, \xi_{p+1}) \\ &+ \frac{\xi_{p+2} - \xi}{\xi_{p+2} - \xi_2} N_A^d(\xi|\xi_2, \xi_3, \cdots, \xi_{p+2}). \end{aligned} \tag{6}$$

The multivariate T-spline blending function can be obtained using the product of the blending functions in each dimension

$$N_A(\boldsymbol{\xi}|\boldsymbol{\Xi}_A) \equiv \prod_{d=1}^{n_d} N_A^d(\xi_A^d|\Xi_A^d). \tag{7}$$

To obtain a rational form for T-spline blending functions, a weight can be assigned to each control point and a rational normalization should be used, in the same way as when extending B-splines to NURBS.

T-splines inherit most of the merits of NURBS, but T-spline blending functions cannot always guarantee linear independence, which is an indispensable

requirement for subsequent numerical analysis [78]. Analysis-suitable T-splines [79], a large subset of T-splines, satisfy this requirement. For T-meshes without extraordinary points, an analysis-suitable T-spline is defined to be one whose T-mesh has no intersecting extension. The interested readers are referred to [2] for a detailed explanation. In the present work, all the three-dimensional models utilize analysis-suitable T-splines.

Although T-splines have intrinsic element structures, the set of the blending functions supported by each element are different. To further integrate IGA with existing FEM codes, the Bézier extraction technique was introduced, first for NURBS in [80] and then for T-splines in [77]. The idea of Bézier extraction is that localized NURBS or T-spline blending functions can be represented by a linear combination of Bernstein polynomials. Bézier extraction provides an element data structure suitable for analysis. That is, similar to Lagrangian polynomial elements in traditional FEM, Bernstein bases do not change from element to element.

The mechanism underlying Bézier extraction is to replicate the existing knots using the knot insertion algorithm until their multiplicity is equal to the order p, thus subdividing the geometry into Bézier elements. The form of Bézier extraction is

$$\mathbf{N}^e(\tilde{\xi}) = \mathbf{C}^e\mathbf{B}(\tilde{\xi}), \tag{8}$$

where $\mathbf{N}$ is a vector of T-spline blending functions which are supported by element e, $\mathbf{C}$ is called Bézier extraction operator, and $\mathbf{B}(\tilde{\xi})$ is Bernstein basis. It is noteworthy that the Bézier extraction operator is only determined by the knot vector, independent of the positions of control points, which is a significant feature for the application in shape optimization, meaning that the same Bézier extraction operator can be kept through all iterative procedures.

3. Isogeometric boundary element methods

3.1. Isogeometric boundary element methods based on regularized boundary integral equations (BIE)

The displacement boundary integral equation (DBIE) is given by

$$C_{ij}(\mathbf{s})\, u_j(\mathbf{s}) + \int\!\!\!\!\!\!-_S T_{ij}(\mathbf{s},\ \mathbf{x}) u_j(\mathbf{x})\, \mathrm{d}S(\mathbf{x}) = \int_S U_{ij}(\mathbf{s},\ \mathbf{x}) t_j(\mathbf{x})\, \mathrm{d}S(\mathbf{x}), \tag{9}$$

where U_{ij} and T_{ij} are fundamental solutions, and in three dimensional elasticity are given by

$$U_{ij}(\mathbf{s},\ \mathbf{x}) = \frac{1}{16\pi\mu(1-\nu)r}[(3-4\nu)\delta_{ij} + r_{,i}r_{,j}], \tag{10}$$

$$T_{ij}(\mathbf{s},\ \mathbf{x}) = \frac{-1}{8\pi(1-\nu)r^2}\left\{\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij} + 3r_{,i}r_{,j}] + (1-2\nu)(n_i r_{,j} - n_j r_{,i})\right\}, \tag{11}$$

where $\mathbf{x}$ is the field point on the boundary, $\mathbf{s}$ the source point, and $r = r(\mathbf{s}, \mathbf{x}) = ||\mathbf{x} - \mathbf{s}||$ the distance between the source point and field point (Fig. 2).

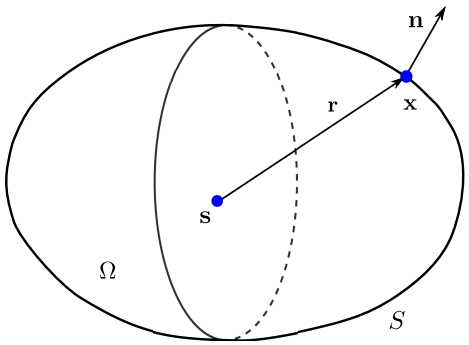


Figure 2: The distance between the source point and field point

The above BIE is a singular form where the first integral in the equation is an improper integral in the sense of Cauchy Principal Value [81]. The singular form was widely used in traditional BEM with isoparametric elements because the singular integral can be bypassed using rigid body motion method. However, rigid body motion technique in the formulation of IGABEM is not available, so

the explicit evaluation of jump terms and strongly singular integrals are necessary with the singular form. Now we employ a regularized boundary integral equation proposed by Liu *et al.* [74, 75, 76], which is written as

$$\int_S T_{ij}\left(\mathbf{s},\ \mathbf{x}\right)\left[u_j\left(\mathbf{x}\right)-u_j\left(\mathbf{s}\right)\right]\mathrm{d}S\left(\mathbf{x}\right)=\int_S U_{ij}\left(\mathbf{s},\ \mathbf{x}\right)t_j\left(\mathbf{x}\right)\mathrm{d}S\left(\mathbf{x}\right). \tag{12}$$

The regularized form above cancels the strong singularity because in three dimensional problems

$$T_{ij}(\mathbf{s},\mathbf{x})\left[u_j(\mathbf{x})-u_j(\mathbf{s})\right]\sim\ \ \mathcal{O}(\tfrac{1}{r^2})\mathcal{O}(r)=\mathcal{O}(\tfrac{1}{r}). \tag{13}$$

Consequently, the regularized form only contains weakly singular integrals, which can be evaluated easily using polar integration.

For the geometries constructed by T-splines, the Cartesian coordinate of a point on the surface can be expressed using T-spline blending functions

$$\mathbf{x}^e(\tilde{\boldsymbol{\xi}})=\sum_{a=1}^{n_a}R_a^e(\tilde{\boldsymbol{\xi}})\mathbf{P}_a^e\qquad\text{in }\tilde{S}, \tag{14}$$

where R is the blending function, $\tilde{S}$ denotes the parent element, e the parent element index, n_a the number of blending functions supported by the element, a the local index of the blending function in element e, $\mathbf{P}$ the control points, and $\tilde{\boldsymbol{\xi}}$ the intrinsic coordinates of the field points in the parent element.

The displacement and traction fields around the boundary are also discretized using T-spline blending functions,

$$u_j^e(\tilde{\boldsymbol{\xi}})=\sum_{a=1}^{n_a}R_a^e(\tilde{\boldsymbol{\xi}})\tilde{u}_j^{ea}, \tag{15}$$

$$t_j^e(\tilde{\boldsymbol{\xi}})=\sum_{a=1}^{n_a}R_a^e(\tilde{\boldsymbol{\xi}})\tilde{t}_j^{ea}, \tag{16}$$

where $\tilde{u}_j^{ea}$ and $\tilde{t}_j^{ea}$ are the unknowns related to control points, $\tilde{\boldsymbol{\xi}}$ the intrinsic coordinates in the parent element.

Using the discretization scheme as Eqs. (14, 15, 16), Eq. (12) is expressed by

$$\sum_{e=1}^{n_e}\int_{\tilde{S}}T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\left[\sum_{a=1}^{n_a}R_a^e(\tilde{\boldsymbol{\xi}})\tilde{u}_j^{ea}-\sum_{a_0=1}^{n_{a_0}}R_{a_0}^{e_0}(\tilde{\boldsymbol{\zeta}}_c)\tilde{u}_j^{e_0a_0}\right]J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})$$

$$= \sum_{e=1}^{n_e} \int_{\tilde{S}} U_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) \sum_{a=1}^{n_a} R_a^e(\tilde{\boldsymbol{\xi}}) \tilde{t}_j^{ea} J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \tag{17}$$

where n_e is the number of elements, J_e is the related Jacobian, c the global index of the collocation point, e_0 the element in which the collocation point is located, a_0 the local index of the basis functions in element e_0, and $\tilde{\boldsymbol{\zeta}}_c$ the intrinsic coordinates of the collocation point in parent elements. Because the integrals are evaluated numerically using Gauss-Legendre quadrature rule which is a summation form, we can rewrite Eq. (17) by splitting its left-hand side as

$$\begin{aligned} &\sum_{e=1}^{n_e} \sum_{a=1}^{n_a} \left\{ \int_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_a^e(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) \right\} \tilde{u}_j^{ea} \\ &- \sum_{a_0=1}^{n_{a_0}} \left\{ \int_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_{a_0}^{e_0}(\tilde{\tilde{\boldsymbol{\zeta}}}_c) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) \right\} \tilde{u}_j^{e_0 a_0} \\ &= \sum_{e=1}^{n_e} \sum_{a=1}^{n_a} \left\{ \int_{\tilde{S}} U_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_a^e(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}})) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) \right\} \tilde{t}_j^{ea}. \end{aligned} \tag{18}$$

With the following definitions,

$$\bar{H}_{ij}^{cea} = \int_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_a^e(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}, \tag{19}$$

$$\hat{H}_{ij}^{ce_0a_0} = \int_{\tilde{S}} T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_{e_0 a_0}(\tilde{\boldsymbol{\zeta}}_c) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}, \tag{20}$$

$$G_{ij}^{cea} = \int_{\tilde{S}} U_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_a^e(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}, \tag{21}$$

Eq. (18) can be written as

$$\sum_{a_0=1}^{n_{a_0}} \hat{H}_{ij}^{ce_0a_0} \tilde{u}_j^{e_0a_0} + \sum_{e=1}^{n_e} \sum_{a=1}^{n_a} \bar{H}_{ij}^{cea} \tilde{u}_j^{ea} = \sum_{e=1}^{n_e} \sum_{a=1}^{n_a} G_{ij}^{cea} \tilde{t}_j^{ea}. \tag{22}$$

Using the mapping from the local node index to the global index A

$$(e, a) \mapsto A, \qquad (e_0, a_0) \mapsto c, \tag{23}$$

Eq. (22) can be assembled into the following equation,

$$H_{ij}^{cA} \tilde{u}_j^A = G_{ij}^{cA} \tilde{t}_j^A, \tag{24}$$

or in a matrix form,

$$\mathbf{Hu} = \mathbf{Gt}. \tag{25}$$

Matrix $\mathbf{H}$ collects the entries of H_{ij}^{cA} and $\mathbf{G}$ of G_{ij}^{cA}. $\mathbf{u}$ and $\mathbf{t}$ contain the nodal parameters of displacements and tractions. Both $\mathbf{u}$ and $\mathbf{t}$ include unknowns and the values given by boundary conditions. By swapping the unknowns of both sides of $\mathbf{u}$ and $\mathbf{t}$, we get

$$\mathbf{Az} = \mathbf{By}. \tag{26}$$

Vector $\mathbf{z}$ contains all displacement and traction unknowns, $\mathbf{y}$ contains all the nodal parameters given by the boundary conditions, $\mathbf{A}$ is a coefficient matrix which is a usually non-symmetric and densely populated, and $\mathbf{B}$ is a matrix which contains the coefficients corresponding to the prescribed boundary conditions. The product of $\mathbf{B}$ and $\mathbf{y}$ yields column vector $\mathbf{f}$ on the right-hand side, so the above equation becomes

$$\mathbf{Az} = \mathbf{f}. \tag{27}$$

3.2. Imposition of boundary conditions

The blending functions in IGABEM lack the *Kronecker delta* property, so the nodal parameters do not possess a clear physical interpretation. Hence, the boundary conditions cannot be substituted directly into nodal parameters. To impose boundary conditions, two approaches can be used as follows.

3.2.1. Semi-discrete method

An approach is not to discretize the part of the boundary where displacements or tractions are prescribed. We rearrange Eq. (12) by separating the integrals into two parts

$$\begin{aligned}
&\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\left\{\int_{\tilde{S}_t} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_a^e(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\tilde{u}_j^{ea}\\
&-\sum_{a_0=1}^{n_{a_0}}\left\{\int_{\tilde{S}_t} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_{a_0}^{e_0}(\tilde{\boldsymbol{\zeta}}_c)J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\tilde{u}_j^{e_0a_0}\\
&+\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\left\{\int_{\tilde{S}_u} U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_a^e(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}}))\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\tilde{t}_j^{ea}\\
=&-\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\int_{\tilde{S}_u} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\bar{u}_j^{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})
\end{aligned}$$

$$+\sum_{a_0=1}^{n_{a_0}}\left\{\int_{\tilde{S}_u} T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\bar{u}_j^{e_0a_0}(\tilde{\boldsymbol{\zeta}}_c)$$
$$-\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\int_{\tilde{S}_t} U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\bar{t}_j^{ea}(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \tag{28}$$

where $\tilde{S}_u$ denotes the parent element on the boundary portion prescribed with displacement boundary conditions, and $\tilde{S}_t$ with traction boundary conditions. Now $\tilde{u}_j$ and $\tilde{t}_j$ on the left-hand side is without boundary conditions and the right-hand side has no unknowns. The left-hand side goes into a matrix $\mathbf{A}$ which contains the coefficients associated with unknowns, and the right-hand side forms a column vector $\mathbf{f}$,

$$\mathbf{Az}=\mathbf{f}, \tag{29}$$

where $\mathbf{z}$ includes all the unknown displacements and tractions. An advantage of this method is that the boundary conditions can be imposed exactly. But the method has difficulties in dealing with mixed boundary conditions and is not convenient for postprocessing due to the loss of a discretized representation on some portions of the boundary.

3.2.2. L_2 projection method

A scheme based on the L_2 projection method consists in enforcing boundary conditions in an "average" sense, *i.e.*

$$\int_{S_u}\mathbf{R}^{\mathrm{T}}\mathbf{u}\mathrm{d}S=\int_{S_u}\mathbf{R}^{\mathrm{T}}\bar{\mathbf{u}}\mathrm{d}S \qquad \text{on } S_u, \tag{30}$$
$$\int_{S_t}\mathbf{R}^{\mathrm{T}}\mathbf{t}\mathrm{d}S=\int_{S_t}\mathbf{R}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}S \qquad \text{on } S_t, \tag{31}$$

where the basis function matrix $\mathbf{R}$ is used as weighting function. Substituting the approximation for displacement and traction (Eqs. (15, 16)) into the above equations leads to

$$\int_{S_u}\mathbf{R}^{\mathrm{T}}\mathbf{R}\tilde{\mathbf{u}}\mathrm{d}S=\int_{S_u}\mathbf{R}^{\mathrm{T}}\bar{\mathbf{u}}\mathrm{d}S, \qquad \text{on } S_u \tag{32}$$
$$\int_{S_t}\mathbf{R}^{\mathrm{T}}\mathbf{R}\tilde{\mathbf{t}}\mathrm{d}S=\int_{S_t}\mathbf{R}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}S \qquad \text{on } S_t. \tag{33}$$

Hence, the solution vector $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{t}}$ can be obtained by solving the following matrix equation

$$\mathbf{A}_1\tilde{\mathbf{u}} = \mathbf{z}_1 \qquad \text{on } S_u, \tag{34}$$

$$\mathbf{A}_2\tilde{\mathbf{t}} = \mathbf{z}_2 \qquad \text{on } S_t, \tag{35}$$

where

$$\mathbf{A}_1 = \int_{S_u} \mathbf{R}^{\mathrm{T}}\mathbf{R}\mathrm{d}S \qquad \text{on } S_u, \tag{36}$$

$$\mathbf{A}_2 = \int_{S_t} \mathbf{R}^{\mathrm{T}}\mathbf{R}\mathrm{d}S \qquad \text{on } S_t, \tag{37}$$

and

$$\mathbf{z}_1 = \int_{S_u} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{u}}\mathrm{d}S \qquad \text{on } S_u, \tag{38}$$

$$\mathbf{z}_2 = \int_{S_t} \mathbf{R}^{\mathrm{T}}\bar{\mathbf{t}}\mathrm{d}S \qquad \text{on } S_t. \tag{39}$$

After obtaining $\tilde{\mathbf{u}}$ and $\tilde{\mathbf{t}}$, we can substitute them into the governing equations for analysis. Hence, the approach can be viewed as a separate preprocessing step.

4. Shape sensitivity analysis with IGABEM

Shape sensitivity analysis refers to the evaluation of the derivatives of a quantity of interest (objective function) with respect to shape design variables. Shape sensitivity analysis is a critical step for gradient-based shape optimization. Implicit methods have been widely used in BEM for sensitivity analysis. The present work also employs implicit differentiation within IGABEM, but has two ingredients differing from previously published approaches:

- The sensitivities of the displacement and traction fields are discretized by T-splines.
- The differentiation form of BIE is a regularized form.

4.1. Implicit differentiation method

Now we take shape derivatives of both sides of the regularized BIE (Eq. (12)) with respect to the design variables

$$\begin{aligned}
&\int_S \left\{ \dot{T}_{ij}\left(\mathbf{s},\mathbf{x}\right)\left[u_j\left(\mathbf{x}\right)-u_j\left(\mathbf{s}\right)\right]+T_{ij}\left(\mathbf{s},\mathbf{x}\right)\left[\dot{u}_j\left(\mathbf{x}\right)-\dot{u}_j\left(\mathbf{s}\right)\right\} \mathrm{d}S\left(\mathbf{x}\right) \\
&+\int_S T_{ij}\left(\mathbf{s},\mathbf{x}\right)\left[u_j\left(\mathbf{x}\right)-u_j\left(\mathbf{s}\right)\right]\left[\mathrm{d}\dot{S}(\mathbf{x})\right] \\
=&\int_S \left[\dot{U}_{ij}\left(\mathbf{s},\mathbf{x}\right)t_j\left(\mathbf{x}\right)+U_{ij}\left(\mathbf{s},\mathbf{x}\right)\dot{t}_j\left(\mathbf{x}\right)\right] \mathrm{d}S\left(\mathbf{x}\right) \\
&+\int_S U_{ij}\left(\mathbf{s},\mathbf{x}\right)t_j\left(\mathbf{x}\right)\left[\mathrm{d}\dot{S}(\mathbf{x})\right]. \qquad (40)
\end{aligned}$$

We remark that $\dot{T}_{ij}$ and $\dot{U}_{ij}$ share the same singularity order with T_{ij} and U_{ij} respectively. Hence, the equation is still without strong singularity. The expressions of $\dot{T}_{ij}$ and $\dot{U}_{ij}$ can be seen in Appendix A. The design variables are normally chosen to be the positions of the control points. This is a natural choice because in CAD the geometries are determined by control points directly, and the CAD designers also modify the geometries through changing control point positions. In addition, a control point in T-splines only influences the local geometry close to it, so it is convenient to find the subset of control points influencing the design area. We set the intrinsic coordinates in parent elements as the material coordinates, which are independent of the design variables. Thus the shape derivatives of the field points are

$$\dot{\mathbf{x}}_e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_a^e(\tilde{\boldsymbol{\xi}})\dot{\mathbf{P}}_a^e. \qquad (41)$$

where R denotes T-spline blending functions of CAD in constructing geometric models and $\dot{\mathbf{P}}$ the sensitivities of control points.

We discretize the displacement field and traction field around the boundary using T-spline blending functions,

$$u_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_a^e(\tilde{\boldsymbol{\xi}})\tilde{u}_j^{ea}, \qquad (42)$$

$$t_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_a^e(\tilde{\boldsymbol{\xi}})\tilde{t}_j^{ea}. \qquad (43)$$

The shape derivatives of the boundary displacement and traction field also need to be discretized using T-spline blending functions as

$$\dot{u}_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_a^e(\tilde{\boldsymbol{\xi}})\dot{\tilde{u}}_j^{ea}, \tag{44}$$

$$\dot{t}_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_a^e(\tilde{\boldsymbol{\xi}})\dot{\tilde{t}}_j^{ea}. \tag{45}$$

By noticing

$$[\mathrm{d}S_e(\mathbf{x})] = J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \tag{46}$$

$$[\mathrm{d}\dot{S_e}(\mathbf{x})] = \dot{J}_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \tag{47}$$

we substitute Eqs. (42-45) to Eq. (40) and gain

$$\begin{aligned}
&\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\int_{\tilde{S}}\left[\dot{T}_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_a^e(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}}) + T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_a^e(\tilde{\boldsymbol{\xi}})\dot{J}_e(\tilde{\boldsymbol{\xi}})\right]\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\tilde{u}_j^{ea}\\
&-\sum_{e=1}^{n_e}\sum_{a_0=1}^{n_{a_0}}\int_{\tilde{S}}\left[\dot{T}_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}}) + T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})\dot{J}_e(\tilde{\boldsymbol{\xi}})\right]\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})R_{a_0}^{e_0}(\tilde{\boldsymbol{\zeta}}_c)\tilde{u}_j^{e_0a_0}\\
&+\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\int_{\tilde{S}}T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_a^e(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\dot{\tilde{u}}_j^{ea}\\
&-\sum_{e=1}^{n_e}\sum_{a_0=1}^{n_{a_0}}\int_{\tilde{S}}T_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})R_{a_0}^{e_0}(\tilde{\boldsymbol{\zeta}}_c)\dot{\tilde{u}}_j^{e_0a_0}\\
=&\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\left\{\int_{\tilde{S}}\left[\dot{U}_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_a^e(\tilde{\boldsymbol{\xi}})J_e(\tilde{\boldsymbol{\xi}}) + U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\tilde{\boldsymbol{\xi}})R_a^e(\tilde{\boldsymbol{\xi}})\dot{J}_e(\tilde{\boldsymbol{\xi}})\right]\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}\tilde{t}_j^{ea}\\
&+\sum_{e=1}^{n_e}\sum_{a=1}^{n_a}\left\{\int_{\tilde{S}}U_{ij}(\tilde{\boldsymbol{\zeta}}_c,\boldsymbol{\xi})J_e(\tilde{\boldsymbol{\xi}})\mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}})\right\}R_a^e(\tilde{\boldsymbol{\zeta}})\dot{\tilde{t}}_j^{ea}.
\end{aligned} \tag{48}$$

The above equation can be assembled to a matrix form in the same way as structural analysis, yielding the following form

$$\dot{\mathbf{H}}\mathbf{u} + \mathbf{H}\dot{\mathbf{u}} = \dot{\mathbf{G}}\mathbf{t} + \mathbf{G}\dot{\mathbf{t}}, \tag{49}$$

where $\mathbf{u}$ and $\mathbf{t}$ are vectors containing the displacement and traction nodal parameters, and $\mathbf{H}$ and $\mathbf{G}$ are the corresponding coefficient matrices. These values can be obtained from the IGABEM structural analysis result. $\dot{\mathbf{H}}$ and $\dot{\mathbf{G}}$ are the

coefficient matrices of the unknown field sensitivities $\dot{\mathbf{u}}$ and $\dot{\mathbf{t}}$. Matrix $\dot{\mathbf{H}}$ is assembled from $\dot{\hat{\mathbf{H}}}$ and $\dot{\tilde{\mathbf{H}}}$, whose entries are given as follows:

$$\dot{\hat{H}}_{ij}^{ce_0a_0} = -\int_{\tilde{S}} [\dot{T}_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) + T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) \dot{J}_e(\tilde{\boldsymbol{\xi}})] \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) R_{a_0}^{e_0}(\tilde{\boldsymbol{\zeta}}_c), \tag{50}$$

$$\dot{\tilde{H}}_{ij}^{cea} = \int_{\tilde{S}} [\dot{T}_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_a^e(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) + T_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_a^e(\tilde{\boldsymbol{\xi}}) \dot{J}_e(\tilde{\boldsymbol{\xi}})] \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \tag{51}$$

The entries in matrix $\dot{\mathbf{G}}$ are

$$\dot{G}_{ij}^{cea} = \int_{\tilde{S}} \left[\dot{U}_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_a^e(\tilde{\boldsymbol{\xi}}) J_e(\tilde{\boldsymbol{\xi}}) + \int_{\tilde{S}} U_{ij}(\tilde{\boldsymbol{\zeta}}_c, \tilde{\boldsymbol{\xi}}) R_a^e(\tilde{\boldsymbol{\xi}}) \dot{J}_e(\tilde{\boldsymbol{\xi}}) \right] \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \tag{52}$$

The boundary conditions for the sensitivity analysis can be found from the material differentiation of the boundary conditions prescribed for structural analysis,

$$\dot{u}_j(\mathbf{x}) = \dot{\bar{u}}_j(\mathbf{x}) \quad \text{on } S_u, \tag{53}$$

$$\dot{t}_j(\mathbf{x}) = \dot{\bar{t}}_j(\mathbf{x}) \quad \text{on } S_t, \tag{54}$$

where $\dot{\bar{u}}_j$ and $\dot{\bar{t}}_j$ are the displacement and traction sensitivity boundary conditions, respectively.

By swapping the unknowns in Eq. (49), a final matrix form is obtained as

$$\begin{bmatrix} \mathbf{H}, & \mathbf{G} \end{bmatrix} \begin{pmatrix} \dot{\mathbf{u}} \\ \dot{\mathbf{t}} \end{pmatrix} = \begin{bmatrix} \dot{\mathbf{H}}, & \dot{\mathbf{G}} \end{bmatrix} \begin{pmatrix} \mathbf{u} \\ \mathbf{t} \end{pmatrix} \tag{55}$$

, or collected as

$$\mathbf{A}\dot{\mathbf{z}} = \dot{\mathbf{A}}\mathbf{z}, \tag{56}$$

where the matrix $\mathbf{A}$ and column vector $\mathbf{z}$ are identical to that in IGABEM structural analysis. The matrix equation can be solved by imposing sensitivity boundary conditions.

4.2. Displacement and stress shape sensitivity recovery

After getting the displacement and traction sensitivities of the control points by solving Eq. (56), we can evaluate the displacement or stress sensitivities in the domain if necessary (Appendix B.1). The displacement and stress sensitivities of the interior point are obtained through the shape differentiation of

Somigliana's identities. To recover the stress sensitivities on the surface, an efficient approach is to use Hooke's law and Cauchy's formula from the displacement, displacement gradient and traction

$$u_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_a^e(\tilde{\boldsymbol{\xi}}) u_j^{ea}, \tag{57}$$

$$\frac{\partial u_j^e(\tilde{\boldsymbol{\xi}})}{\partial \tilde{\boldsymbol{\xi}}} = \sum_{a=1}^{n_a} \frac{\partial R_a^e(\tilde{\boldsymbol{\xi}})}{\partial \tilde{\boldsymbol{\xi}}} u_j^{ea}, \tag{58}$$

$$t_j^e(\tilde{\boldsymbol{\xi}}) = \sum_{a=1}^{n_a} R_a^e(\tilde{\boldsymbol{\xi}}) t_j^{ea}. \tag{59}$$

.

The detailed formulation can be seen in Appendix B.2.

5. Shape optimization using IGABEM

5.1. Shape optimization formulations

Shape optimization can be conducted through a gradient-less or gradient-based method. The gradient-less shape optimization does not require the evaluation of the shape derivatives, but can be prohibitively time-consuming for realistic problems and is not supported by a mathematical theory. So, gradient-based methods are normally preferred and thus are also used in the present work. Gradient-based shape optimization has a well-grounded mathematical foundation rooted in optimal-control theory. A shape optimization problem can be formulated as minimizing an objective function

$$f : \mathbb{R}^n \to \mathbb{R}, \tag{60}$$

$$f : \mathbf{t} \mapsto f(\mathbf{t}), \tag{61}$$

subject to the constraints

$$g_i(\mathbf{t}) \leqslant 0 \qquad \text{for } i = 1, \ldots, m, \tag{62}$$

$$t_i^l \leqslant t_i \leqslant t_i^u, \tag{63}$$

where $\mathbf{t}$ is a vector of parameters which controls geometrical configurations, also called design variables. f is the objective function, g_i the constraint functions, i the constraint function index, m the number of constraints. Eq. (63) gives side constraints to limit the search region for the optimum, where t_i^l and t_i^u are lower and upper bounds of the design variables, respectively. The side constraints can guarantee that the result is not a meaningless geometry, for example, precluding the splitting or self intersection of the control mesh. If the design variables are chosen to control the surface, as in our work, the side constraints can preclude the possibility of the surface splitting. If the design variables also control or influence the mesh inside the domain, as in the FEM, the side constraints must also guarantee the connectivity of the domain mesh.

To find the parameters associated with the minimum value of the objective function, numerical optimization algorithms employ the gradient of the objective and constraint functions within an iterative algorithm:

$$\left(f_k, \mathbf{g}_i^k, \frac{\mathrm{d}}{\mathrm{d}\mathbf{t}} f_k, \frac{\mathrm{d}}{\mathrm{d}\mathbf{t}} \mathbf{g}_i^k\right) \rightarrow \left(f_{k+1}, \mathbf{g}_i^{k+1}\right), \tag{64}$$

where k denotes the kth iterative step, $\frac{\mathrm{d}}{\mathrm{d}\mathbf{t}} f_k$ and $\frac{\mathrm{d}}{\mathrm{d}\mathbf{t}} \mathbf{g}_k$ are called shape derivatives in shape optimization problem. A numerical shape optimization procedure is shown in Fig. 3.

5.2. Shape derivatives of some quantities

The displacement and stress shape sensitivities can be obtained from the procedure shown in Section 5. However, a bit more effort is needed to calculate the sensitivities of some other commonly used quantities. To be consistent with our CAD and analysis model, all of the domain integrals involved should be transformed into boundary integral forms.

- The shape derivatives of volume V. The volume and area can be transformed into boundary integral readily by using the divergence theorem

$$V = \int_\Omega \mathrm{d}\Omega = \frac{1}{3}\int_\Omega \nabla \cdot \mathbf{x} \mathrm{d}\Omega = \frac{1}{3}\int_S \mathbf{x} \cdot \mathbf{n} \mathrm{d}S = \sum_{e=1}^{n_e} \frac{1}{3}\int_{\tilde{S}} \mathbf{x} \cdot \mathbf{n} J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \tag{65}$$

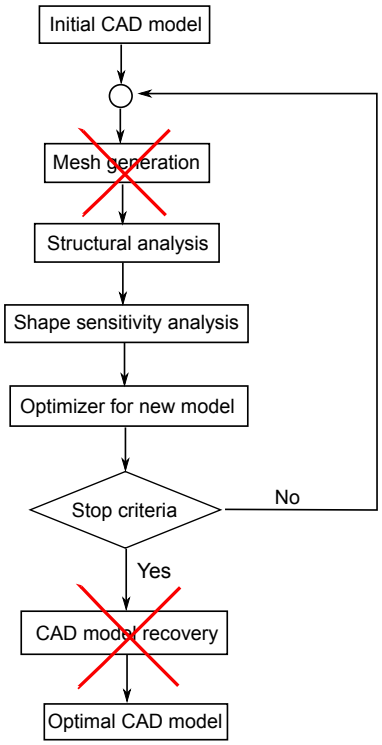


Figure 3: IGABEM shape optimization flowchart

so that the shape derivatives are

$$\begin{aligned}\dot{V} &= \sum_{e=1}^{n_e} \frac{1}{3} \int_{\tilde{S}} \left[\mathbf{x} \cdot \mathbf{n} J_e(\tilde{\boldsymbol{\xi}}) \right] \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) \\ &= \sum_{e=1}^{n_e} \frac{1}{3} \int_{\tilde{S}} \left[\dot{\mathbf{x}} \cdot \mathbf{n} J_e(\tilde{\boldsymbol{\xi}}) + \mathbf{x} \cdot \dot{\mathbf{n}} J_e(\tilde{\boldsymbol{\xi}}) + \mathbf{x} \cdot \mathbf{n} \dot{J}_e(\tilde{\boldsymbol{\xi}}) \right] \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}), \end{aligned} \tag{66}$$

- The shape derivatives of the conserved energy E.

$$E = \int_{\Omega} \sigma_{ij} \epsilon_{ij} \mathrm{d}\Omega = \int_{S} t_k u_k \mathrm{d}S. \tag{67}$$

Its shape derivative $\dot{E}$ is given by

$$\begin{aligned}\left[\int_S t_i(\mathbf{x}) \, u_i(\mathbf{x}) \, \mathrm{d}S(\mathbf{x}) \right]^{\cdot} &= \sum_{e=1}^{n_e} \int_{\tilde{S}} [\dot{t}_i(\tilde{\boldsymbol{\xi}}) u_i(\tilde{\boldsymbol{\xi}}) + t_i(\tilde{\boldsymbol{\xi}}) \dot{u}_i(\tilde{\boldsymbol{\xi}})] J_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}) \\ &\quad + \sum_{e=1}^{n_e} \int_{\tilde{S}} t_i(\tilde{\boldsymbol{\xi}}) u_i(\tilde{\boldsymbol{\xi}}) \dot{J}_e(\tilde{\boldsymbol{\xi}}) \mathrm{d}\tilde{S}(\tilde{\boldsymbol{\xi}}). \end{aligned} \tag{68}$$

5.3. The description of the optimization algorithm

1. Construct a CAD model using the T-splines. In our work, we used the Autodesk® T-splines plug-in for Rhino®.
2. Choose a subset of the control points as the design points (variables). To guarantee a reasonable geometry, the move limits and the constraints of the control points should be prescribed by the users. The shape derivatives of the geometry points on the geometry are determined by that of the control points.

$$\dot{\mathbf{x}}_e = \sum_{a=1}^{n_a} R_a^e \dot{\mathbf{P}}_{ea}, \tag{69}$$

which are evaluated for the following three types of control points.

- Design control points. For the control points which are set to be the design variables, the associated shape derivatives are unity with respect to the movement of itself and zero with respect to other design points.
- Fixed control points. Some control points are fixed in the optimization procedure, such as the ones corresponding to zero displacement boundary conditions, and their shape sensitivities are always zero.
- Linked control points. To keep the geometry vary "reasonably" or in a good topology, some control points which are not the design variables also need to move according to some rules. This concept is parallel to the domain mesh update in FEM optimization, but we only concern the control points on the boundary. The shape derivatives can be derived from the design boundary control mesh topology. The relationships between design points and linked points can be exerted by the designer provided it leads to a reasonable geometry. In our work, the movement of the control points are linked to that of design control points in the same control grid segments with prescribed fixed ratio. The use of Laplacian smoothing method is a more practical approach and will be investigated in the future work.

3. Read the CAD file using the solver. The optimization solver will automatically update the position of the control points, thereby updating the geometry. It is worth repeating that the Bézier extraction operator can be retained through all iterative procedures because it is independent on the control points.
4. The optimization procedure stopped when the convergence solution is achieved. The resulting geometry can be read directly by CAD software.

6. Numerical examples

In this section we will investigate the performance of IGABEM for shape sensitivity analysis through some numerical examples with or without closed-form solutions. All the geometries are modelled using T-splines exported from the Rhino® T-spline plugin [82]. The optimization solver uses the method of moving asymptotes (MMA) [83]. The Gauss-Legendre quadrature is used and the quadrature order is 12. To study the accuracy of numerical results $(\cdot)_h$ against analytical solutions $(\cdot)$, we define the following errors:

$$e_{L_2}(\cdot)_h = \frac{\|(\cdot)_h - (\cdot)\|_{L_2}}{\|(\cdot)\|_{L_2}}, \tag{70}$$

and

$$e_{L_\infty}(\cdot)_h = \frac{\|(\cdot)_h - (\cdot)\|_{L_2}}{\|(\cdot)\|_{L_\infty}}, \tag{71}$$

with

$$\|(\cdot)\|_{L_2} = \sqrt{\int_S (\cdot)\cdot(\cdot)\mathrm{d}S}, \tag{72}$$

and

$$\|(\cdot)\|_{L_\infty} = \max_{1\leqslant i\leqslant n} |(\cdot)_i|. \tag{73}$$

6.1. A shape sensitivity analysis example (spherical cavity)

Fig. 4a shows a problem of a traction free spherical cavity in an infinite domain subject to a tension $S = 10^5$ at infinity. The problem is analyzed by extracting a finite cube domain around the cavity (Fig. 4b) and by exerting the

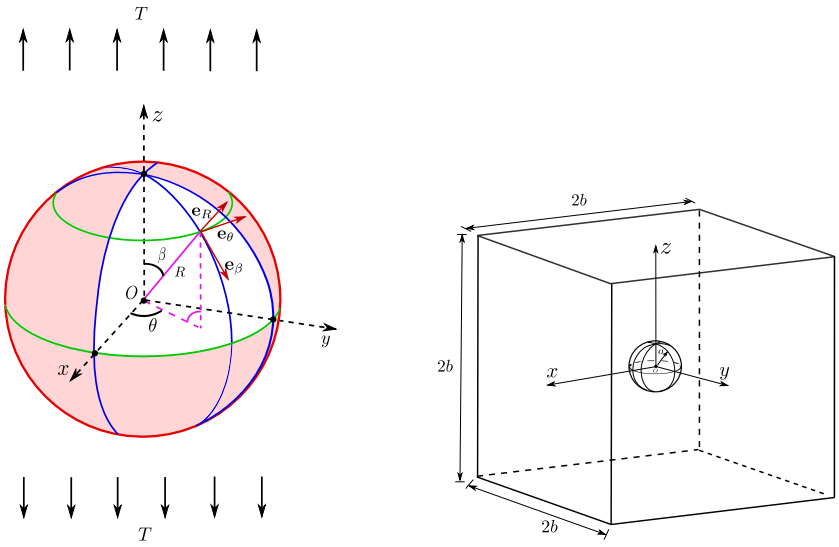
analytical displacement solutions around the cube surface as boundary conditions. The radius of the cavity is $a = 0.5$, and the length of the cube is $2b = 10$. The Young's modulus is $E = 10^5$, and Poisson's ratio $\nu = 0.3$. The analytical solution of the displacement and the shape sensitivities can be seen in Appendix C. We take the cavity radius a as the design variable.

The boundary conditions for structural and shape sensitivity analysis are enforced using the L_2 projection method. The analysis model has 224 Bézier elements on the surface, and 560 control points.

The comparison between the analytical and the numerical displacement sensitivities on the surface is shown in Fig. 5, and the displacement sensitivity errors are shown in Fig. 6. An excellent agreement can be seen. To investigate the shape sensitivities at the interior points, we take an inner spherical surface in the domain with the radius R =2.5. The points in the domain are assumed to be regularly distributed along the radial line, *i.e.*

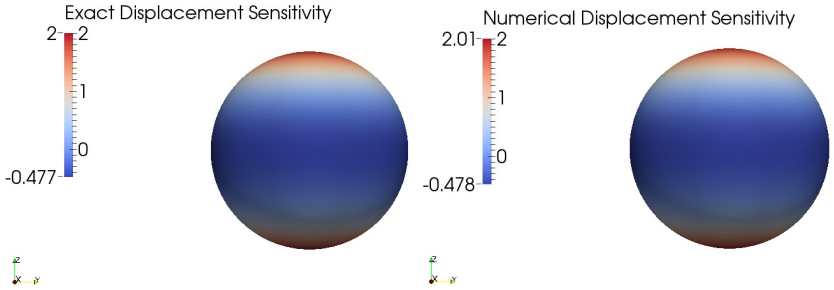
$$\dot{R} = \begin{cases} \frac{\left|\frac{L}{\cos\theta\sin\beta}\right|-R}{\left|\frac{L}{\cos\theta\sin\beta}\right|-a} & \text{if } |x| \geqslant |y| \text{ and } \frac{L}{|L/\cos\theta|} \geqslant \left|\frac{z}{R\sin\beta}\right|, \\ \\ \frac{\left|\frac{L}{\cos\beta}\right|-R}{\left|\frac{L}{\cos\beta}\right|-a} & \text{if } |x| \geqslant |y| \text{ and } \frac{L}{|L/\cos\theta|} < \left|\frac{z}{R\sin\beta}\right|, \\ \\ \frac{\left|\frac{L}{\sin\theta\sin\beta}\right|-R}{\left|\frac{L}{\sin\theta\sin\beta}\right|-a} & \text{if } |x| < |y| \text{ and } \frac{L}{|L/\sin\theta|} \geqslant \left|\frac{z}{R\sin\beta}\right|, \\ \\ \frac{\left|\frac{L}{\cos\beta}\right|-R}{\left|\frac{L}{\cos\beta}\right|-a} & \text{if } |x| < |y| \text{ and } \frac{L}{|L/\sin\theta|} < \left|\frac{z}{R\sin\beta}\right|. \end{cases} \tag{74}$$

The good agreements with analytical displacement and stress sensitivities are shown in Figs. 7 and 9, respectively. The related errors are shown in Figs. 8 and 10. The convergence study can be found in Fig. 11. It is noted that the facets of the figures arise from the visualization algorithm, not from the CAD model itself.



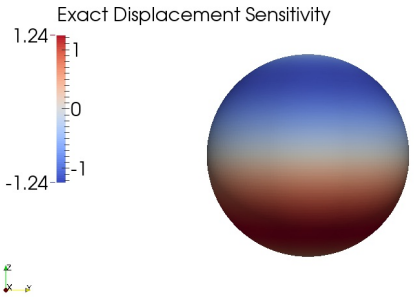
(a) (b)

Figure 4: (a) The definition of the spherical cavity problem, and (2) the analysis model of the spherical cavity problem

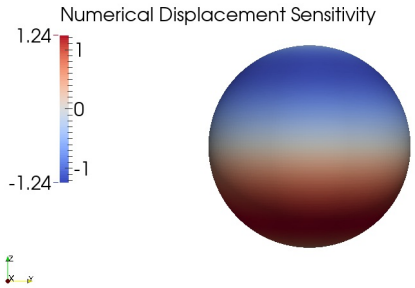


(a) Exact $\dot{u}_R$

(b) Numerical $\dot{u}_R$

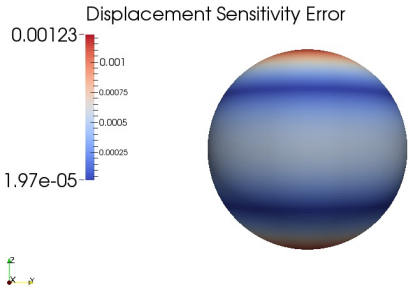


(c) Exact $\dot{u}_\beta$

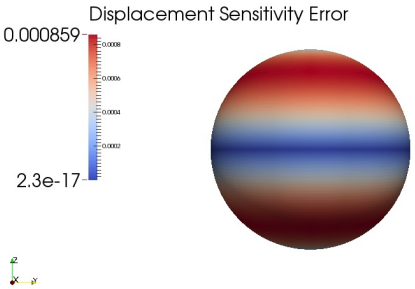


(d) Numerical $\dot{u}_\beta$

Figure 5: Displacement sensitivities on the cavity surface

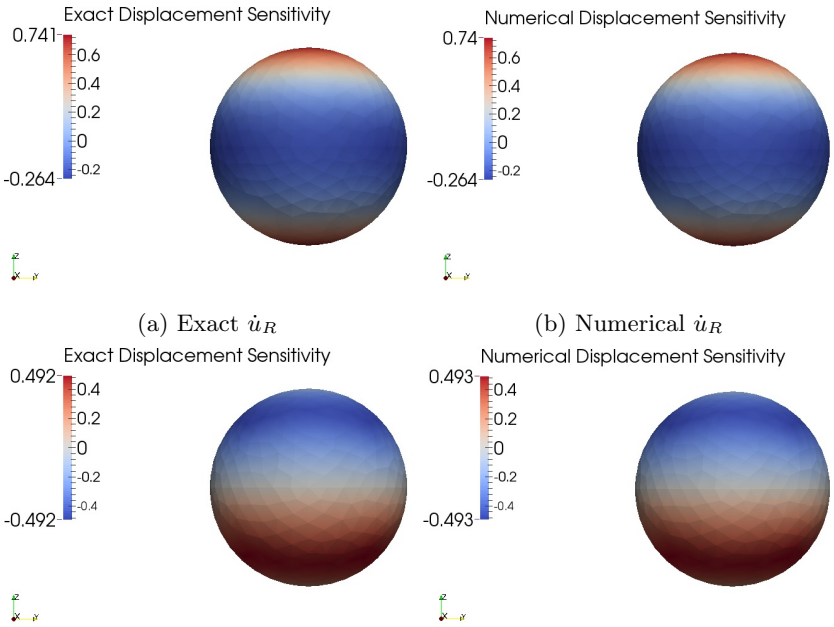


(a) $e_{L_\infty}(\dot{u}_R)$



(b) $e_{L_\infty}(\dot{u}_\beta)$

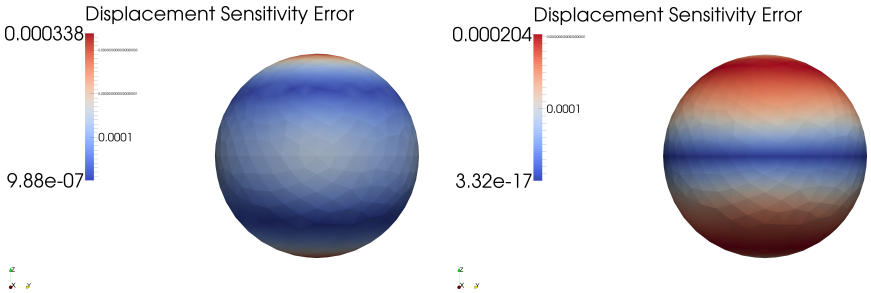
Figure 6: Displacement sensitivity errors on the cavity surface



(a) Exact $\dot{u}_R$ (b) Numerical $\dot{u}_R$

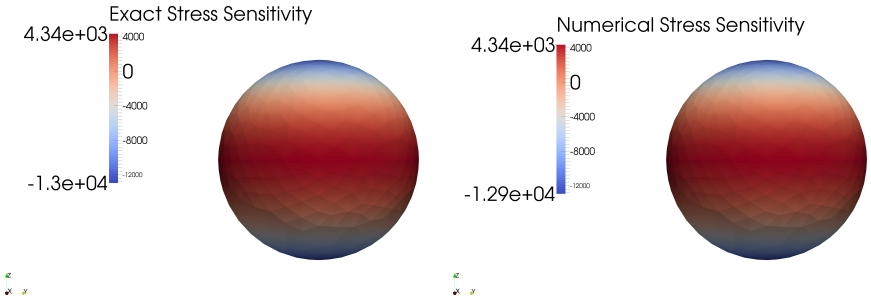
(c) Exact $\dot{u}_\beta$ (d) Numerical $\dot{u}_\beta$

Figure 7: Displacement sensitivities at interior points of the cavity ($R = 2.5$)



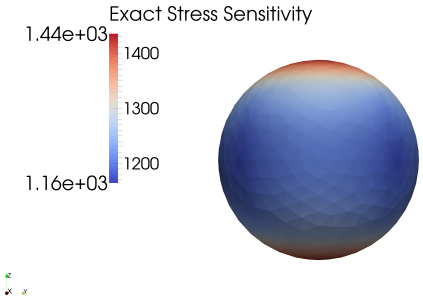
(a) $e_{L_\infty}(\dot{u}_R)$ (b) $e_{L_\infty}(\dot{u}_\beta)$

Figure 8: Displacement sensitivity errors at interior points of the cavity ($R = 2.5$)

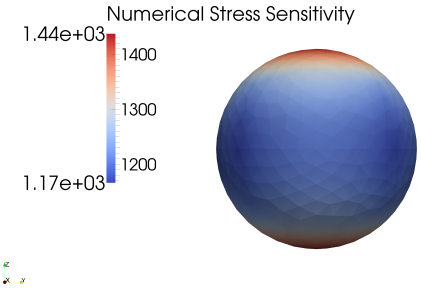


(a) Exact $\dot{\sigma}_{RR}$

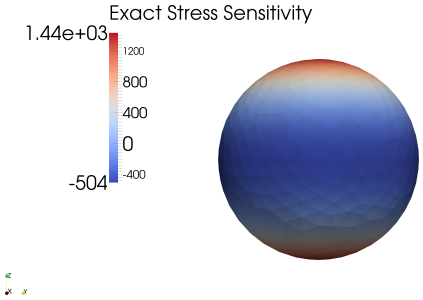
(b) Numerical $\dot{\sigma}_{RR}$



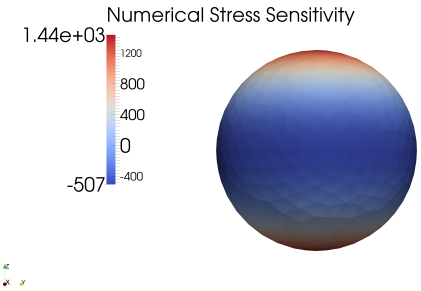
(c) Exact $\dot{\sigma}_{\beta\beta}$



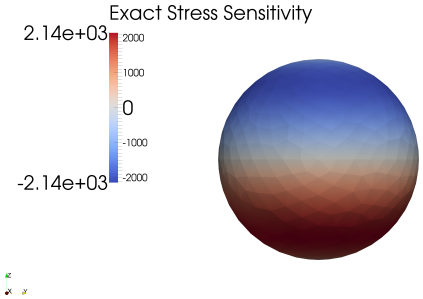
(d) Numerical $\dot{\sigma}_{\beta\beta}$



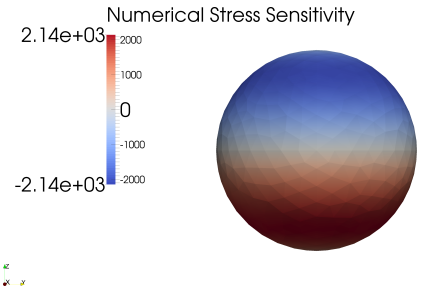
(e) Exact $\dot{\sigma}_{\theta\theta}$



(f) Numerical $\dot{\sigma}_{\theta\theta}$

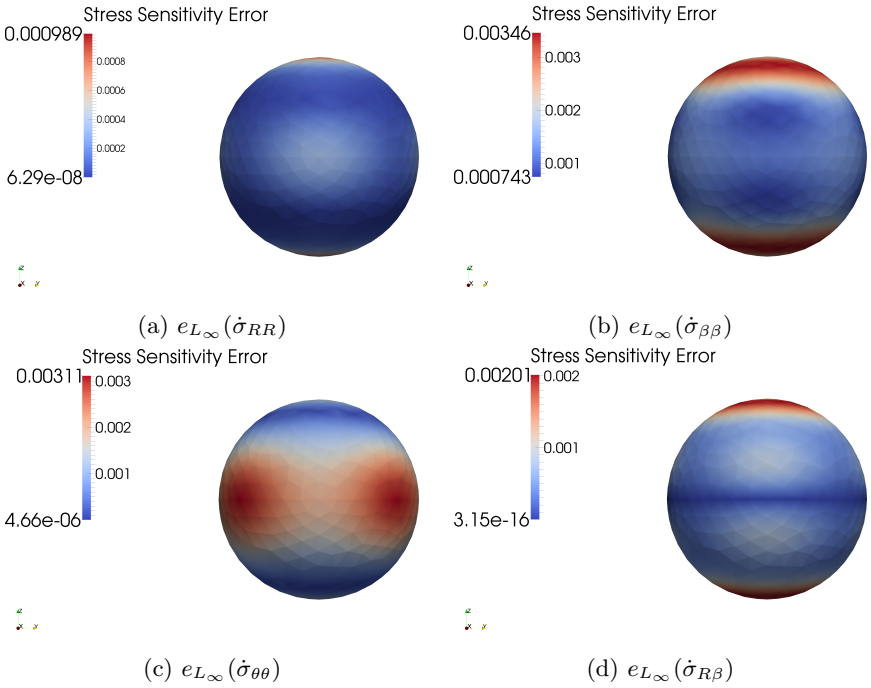


(g) Exact $\dot{\sigma}_{R\beta}$



(h) Numerical $\dot{\sigma}_{R\beta}$

Figure 9: Stress sensitivities at interior points of the cavity ($R = 2.5$)



(a) $e_{L_\infty}(\dot{\sigma}_{RR})$ (b) $e_{L_\infty}(\dot{\sigma}_{\beta\beta})$

(c) $e_{L_\infty}(\dot{\sigma}_{\theta\theta})$ (d) $e_{L_\infty}(\dot{\sigma}_{R\beta})$

Figure 10: Stress sensitivity errors at interior points of the cavity ($R = 2.5$)

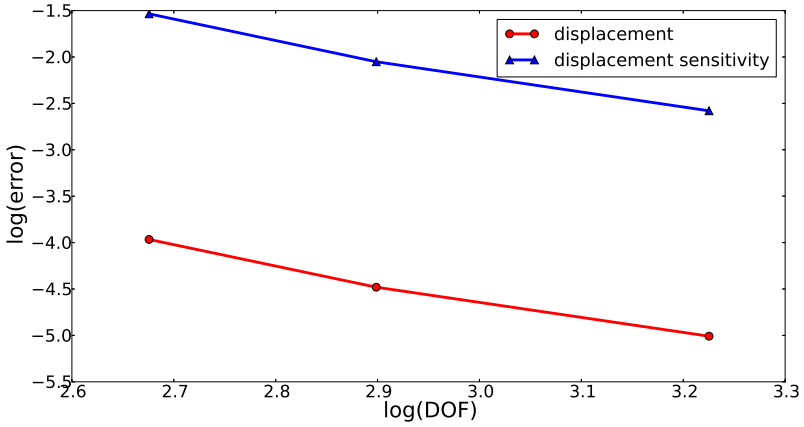


Figure 11: $e_{L_2}(\mathbf{u}_h)$ and $e_{L_2}(\dot{\mathbf{u}}_h)$ against DOF for the cavity problem

6.2. Shape optimization examples

6.2.1. Cantilever beam

Consider a three dimensional cantilever beam, fixed at the left side and subject to a traction on the bottom. The traction is in the z-direction and linearly distributed along the length as $\bar{t} = -100y$. The material parameters are Young's modulus $E = 10^5$, and Poisson's ratio $\nu = 0.3$. The length of the beam is $L = 60$, and the width and height are $h = 20$ (Fig. 12). A cubic T-spline model of the cantilever beam is exported from Rhino® T-spline plugin [82] (Fig. 13a) with 336 Bézier elements and 125 control points (Fig. 13b).

The objective is to minimize the displacement of the beam's end. The design variables are the control points on the top fibre/surface except that on the left side. The vertical positions of the bottom fibre/surface are fixed during optimization, with that of the remaining control points in the same column varying linearly along y-direction as linked control points, as Fig. (14). The volume constraint is $V \leqslant 26400$. The side constraint is $10 \leqslant z \leqslant 30$. After the iterative process (Fig. 16), an optimized geometry and vertical displacement distribution is produced (Fig. 15b), against to that of the initial design (Fig. 15a). The final positions of the control points can be seen in Tab. 1. It is noted that the slight oscillation in the iterative process is due to the violation of the constraints.

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	10	30	20	30
t_2	10	30	20	27.3548
t_3	10	30	20	19.7754
t_4	10	30	20	13.2369

Table 1: Design variables in the 3D beam optimization procedure

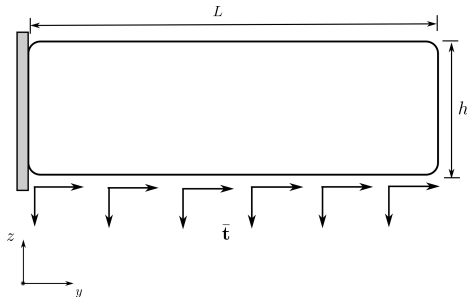
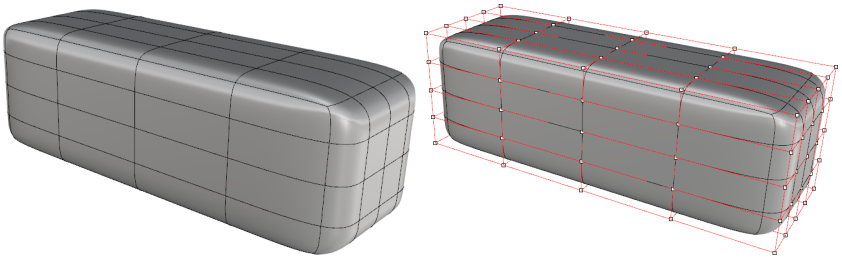


Figure 12: The definition of the 3D beam problem



(a) (b)

Figure 13: (a) The geometry of the 3D beam problem, and (b) the control points of the 3D beam

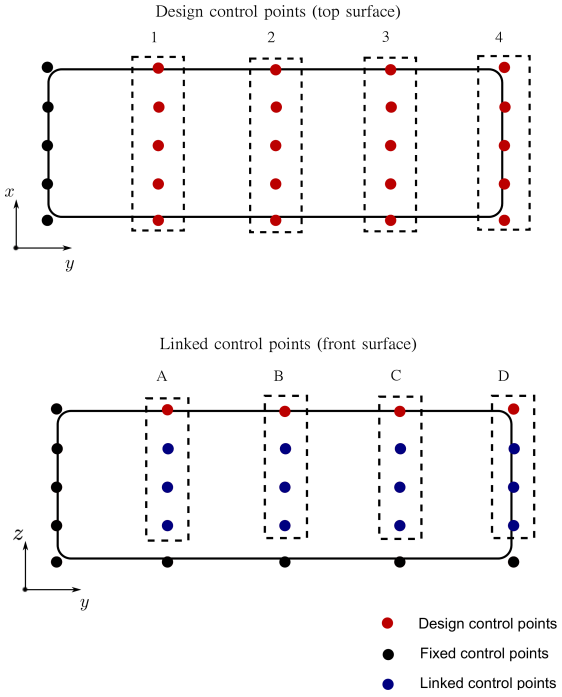
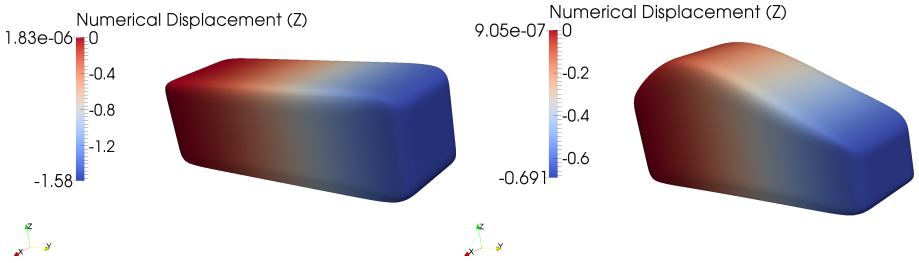


Figure 14: The design and linked control points of the 3D beam



(a) (b)

Figure 15: (a) The initial geometry of the 3D beam, and (b) the optimized geometry of the 3D beam

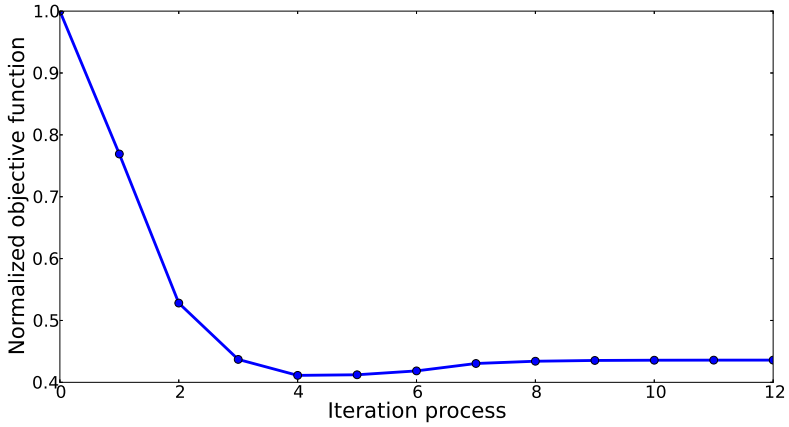
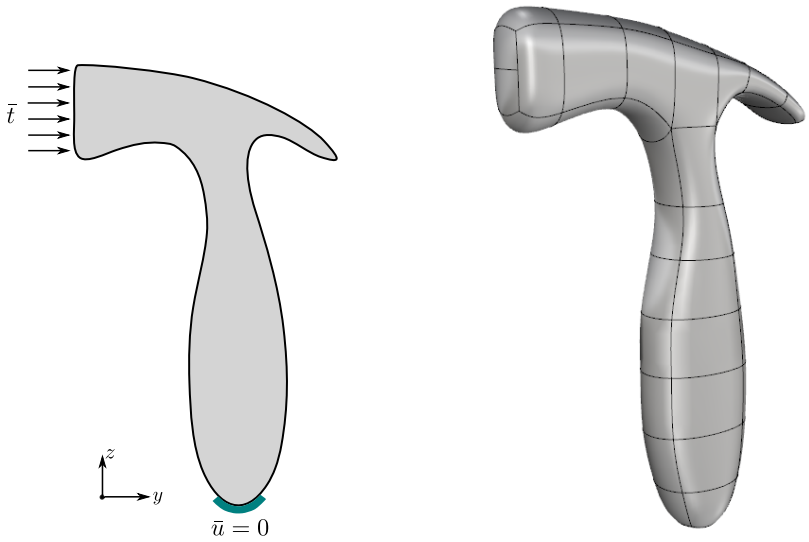


Figure 16: The iterative process of the 3D beam optimization

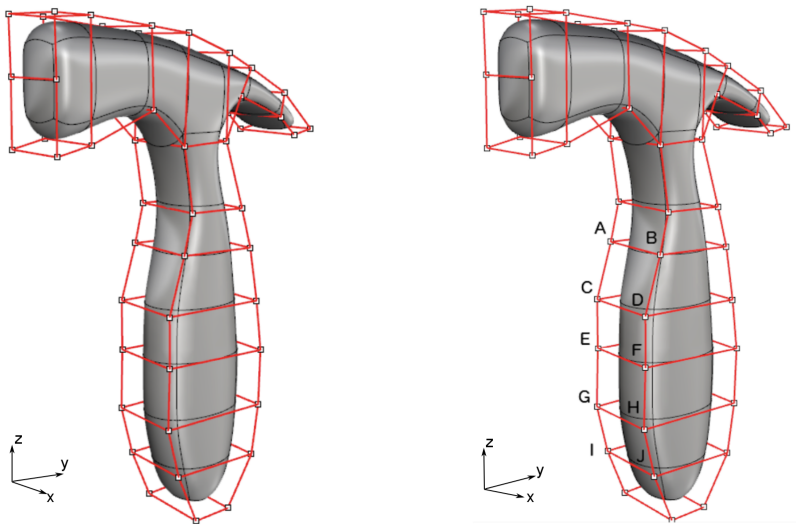
6.2.2. Hammer

The objective is to minimize the conserved energy of a hammer with a volume constraint. The T-spline model of the hammer and the related control points are shown in Figs. 17b and 18a. The hammer is fixed at the bottom, and subject to a uniform traction $\bar{t} = 10^2$ in y-direction on the front (Fig. 17a). The Young's modulus is $E = 10^5$, Poisson's ratio ν =0.3. The design control points are shown in Fig. 18b and the components in y-direction of the control points A(B), C(D), E(F), G(H), I(J) are set as design variables. The initial values of the design variables are $[2.45, 1.25, 1.33, 1.28, 2.30]$ and the side constraints are $0 \leqslant y \leqslant 4$ for all the control points. The initial volume is $V = 1257.63$ and the volume constraint is $V \leqslant 1307.94$. Fig. 20 illustrates the convergence of the iterative process, leading to an optimized geometry as shown in Fig. 19b, compared to the initial geometry in Fig. 19a. The final values of the design variables can be found in Tab. 2. Through the whole optimization procedure, the structural and shape sensitivity analysis can communicate with the the CAD model, and no meshing/remeshing is needed. The final optimized model can be returned directly to the CAD designer without any postprocessing or smoothing procedure.



(a) (b)

Figure 17: (a) Hammer problem definitions, and (b) hammer T-spline model

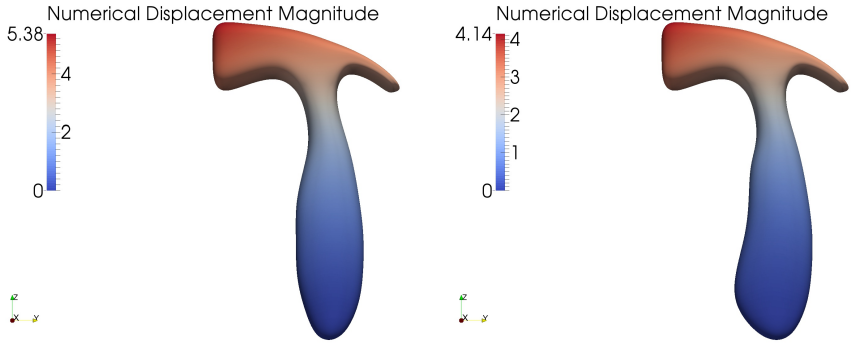


(a) (b)

Figure 18: (a) Hammer control points, and (b) hammer design points

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	0	4	2.45	1.8977
t_2	0	4	1.25	1.8353
t_3	0	4	1.33	1.4129
t_4	0	4	1.28	0
t_5	0	4	2.30	0

Table 2: Design variables in the hammer optimization procedure



(a) (b)

Figure 19: (a) The initial shape of the hammer, and (b) the optimized shape of the hammer

To further test the robustness of the present methodology, we take two other hammer geometries with different initial values for the design parameters. One is with the initial parameters [0.2, 0.2, 0.2, 0.2] and the initial shape is shown in Fig. 21a. From Fig. 22 we can see that the optimization process initially violates the volume constraints, thus leading to an increase of strain energy at the first steps. After locating a feasible region, the strain energy decreases and converges. The other set of parameters is [3.8, 3.8, 3.8, 3.8] with the initial shape as shown in Fig. 21b and the iterative process in Fig. 23. Both experiments converge to the same result as the initial one, reaching the same value of conserved energy 5916 and design parameters [1.9, 1.8, 1.4, 0.0, 0.0].

Next we choose a T-shape component as the initial geometry of the shape optimization, as shown in Fig. 24. The objective function, boundary conditions

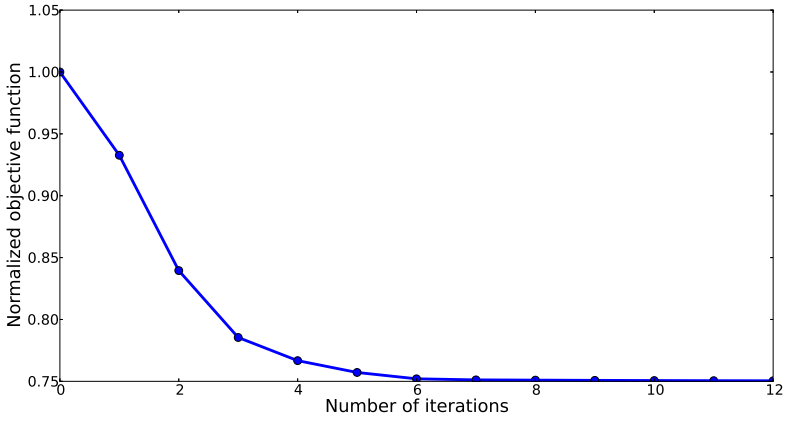
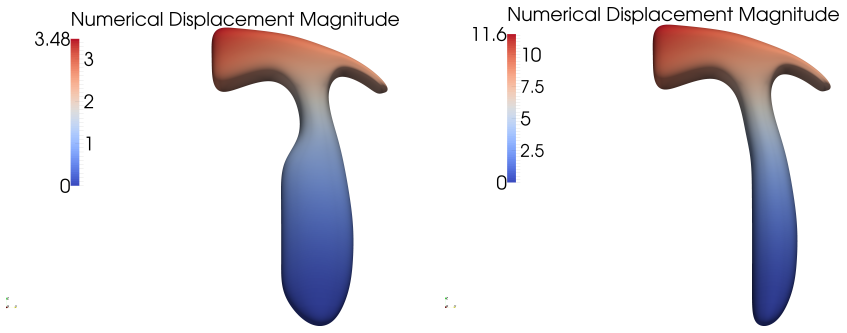


Figure 20: The iterative procedure of the hammer optimization

and material parameters are the same as above. The volume constraint is $V \leqslant 1564$. As shown in Fig 25, the design control points are divided to seven groups, and the design variables are listed as follows,

- t_1, the y-coordinates of the control points A(1, 2), B(1, 2), and C(1, 2).
- t_2, the y-coordinates of the control points D(1, 2) and E(1, 2).
- t_3, the y-coordinates of the control points F(1, 2), G(1, 2), and H(1, 2).
- t_4, the y-coordinates of the control points I(1, 2) and J(1, 2).
- t_5, the z-coordinates of the control points K(1, 2), L(1, 2), and M(1, 2).
- t_6, the z-coordinates of the control points N(1, 2), O(1, 2), and P(1, 2).
- t_7, the z-coordinates of Q(1, 2).

The initial values of the design variables and the side constraints can be seen in Tab. 3. Fig. 27 illustrates the convergence of the iterative process and Tab. 3 shows the side constraints and the final values of the design variables. It can be observed that, starting from a T-shape geometry, (Fig. 26a), the optimization procedure leads to a hammer-shape geometry (Fig. 26b).



(a) (b)

Figure 21: (a) The initial shape of the hammer in the second test, and (2) the initial shape of the hammer in the third test

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	-2	3.5	1.65	1.05
t_2	-2	3.5	1.4	-1.12
t_3	5.5	9	7.5	7.53
t_4	5.5	9	7.7	9
t_5	10.5	16	13.5	10.5
t_6	3	7.5	5.5	7.5
t_7	2	8	5.4	8

Table 3: Design variables in the T-shape component optimization procedure

6.2.3. Chair

To test the present method on more realistic geometries, consider a chair problem with a watertight geometry constructed by T-splines as given by Fig. 28. The original geometry file is sourced from [84] and contains 922 Bézier elements. The chair is fixed on the bottom and subject to a uniformly distributed traction with the magnitude of 50 along the opposite z-direction on the face. The Young's modulus is $E = 10^5$, and Poisson's ratio is $\nu = 0.3$. The optimization objective is to minimize the displacement magnitude of the center on the chair face.

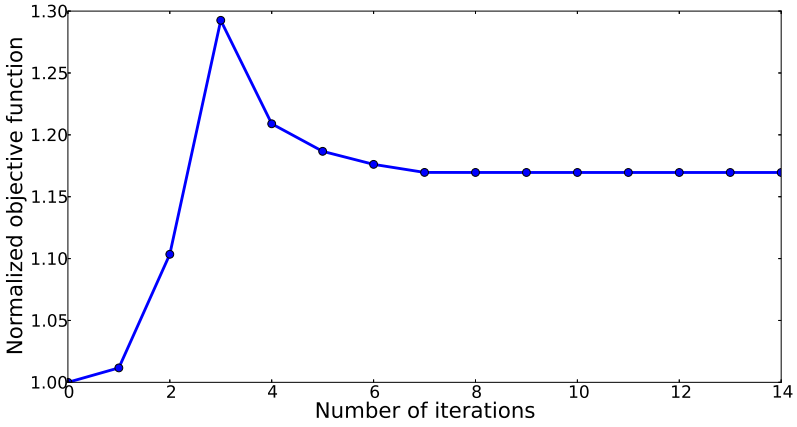


Figure 22: The iterative procedure of the hammer optimization (the second test)

In Fig. 29, A, B, and C denote three sets of control points. Each set has five control points which share the same values of y-coordinates. The y-coordinates of the control point set are set as design variables and their initial values and the corresponding side constraints can be found in Tab. 4. The initial value of the volume V is 1353.45 and the constraint is $V \leqslant 1385$. The initial geometry and the displacement fields are shown in Fig. 30. The optimized solution reduces the objective function and the final geometry is shown in Fig. 31. Fig. 32 shows the close-up image of the deformed section of the chair in shape optimization procedure. The change of the design variables can be found in Tab. 4 and the change of all control points can be seen in the tables of Appendix D. The iterative procedure is illustrated by Fig. 33. The example shows the ability of the present method of optimizing problems with complicated geometries. Throughout the optimization procedure no mesh generation is needed and the optimized geometry remains a CAD model.

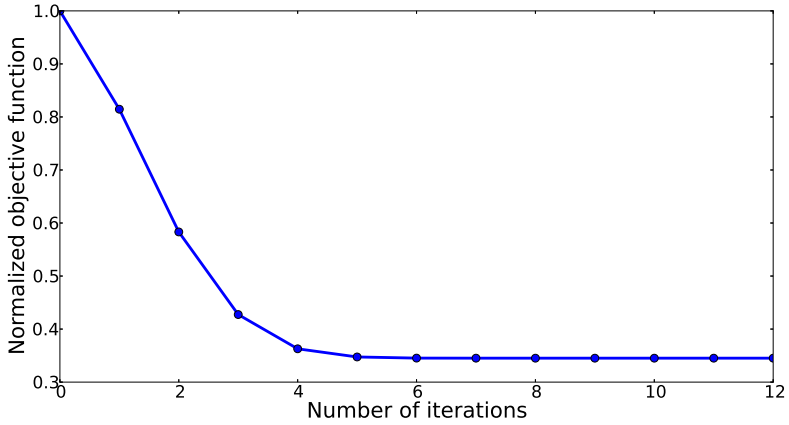
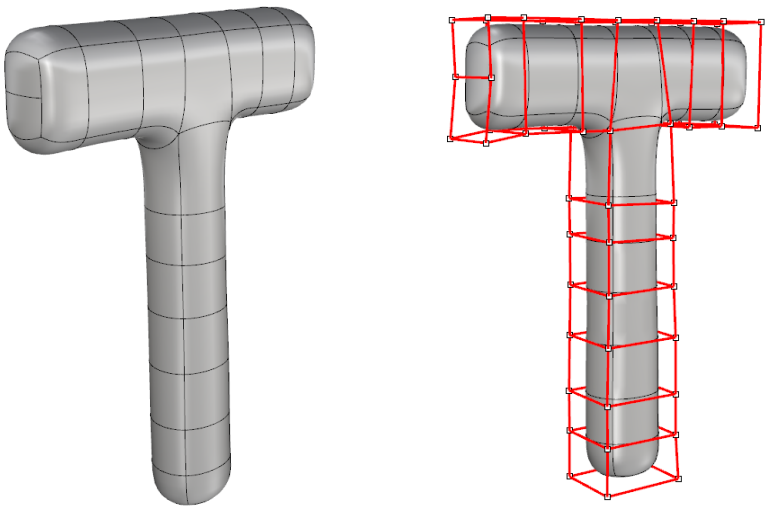


Figure 23: The iterative procedure of the hammer optimization (the third test)

Design variable	Lower bound	Upper bound	Initial value	Final value
t_1	8.2	12.5	10.5	9.48
t_2	5.4	10.4	7.5	9.84
t_3	-4.0	-0.5	-2.0	-2.56

Table 4: Design variables in the chair optimization procedure



(a) (b)

Figure 24: (a) T-shape component geometry, and (b) T-shape component control points

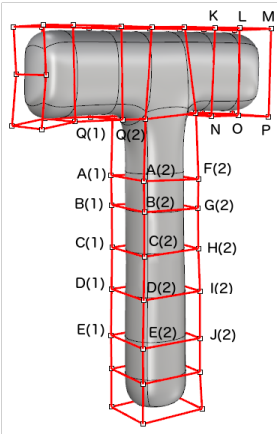
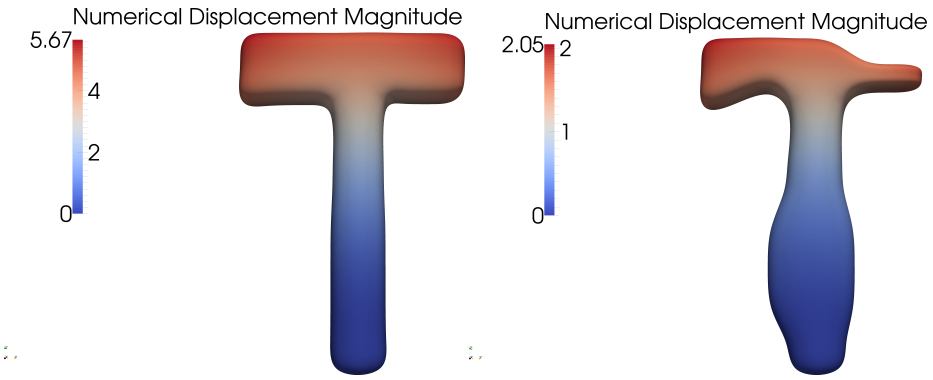


Figure 25: The design control points of the T-shape component



(a) (b)

Figure 26: (a) The initial shape of the T-shape component, and (2) the optimized shape of the T-shape component

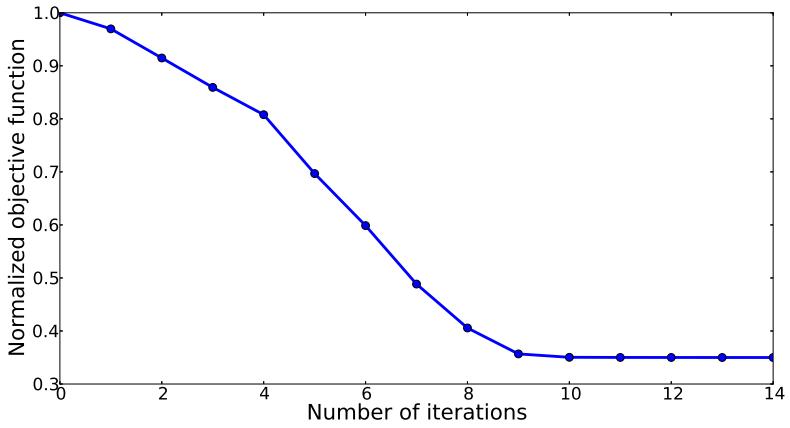
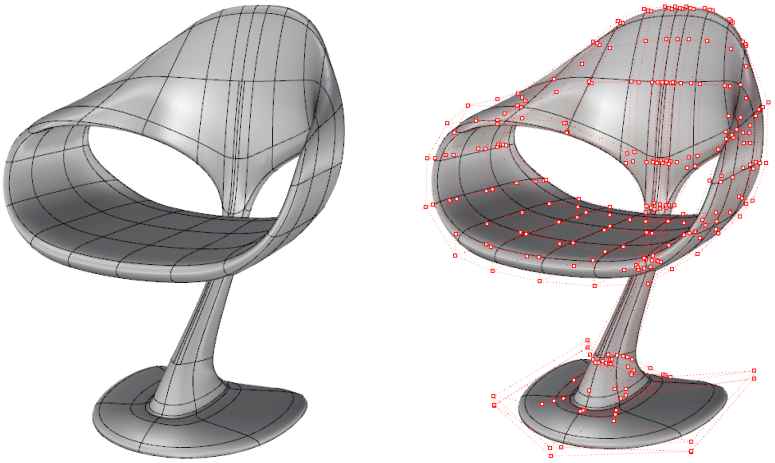


Figure 27: The iterative procedure of the T-shape component optimization



(a) (b)

Figure 28: (a) Chair geometry, and (b) chair control points

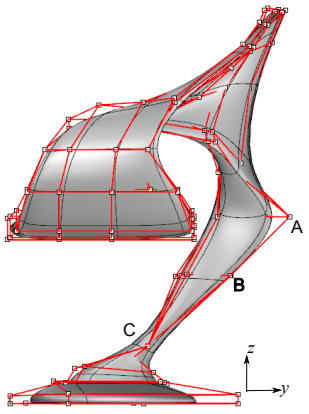


Figure 29: The design points of the chair

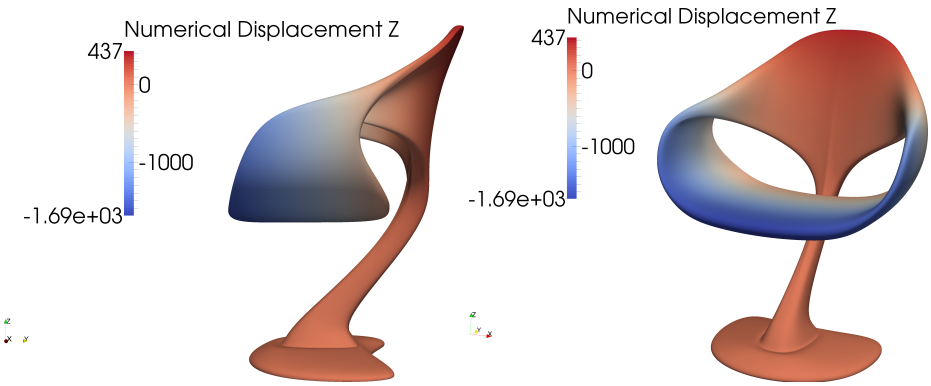


Figure 30: The initial shape of the chair

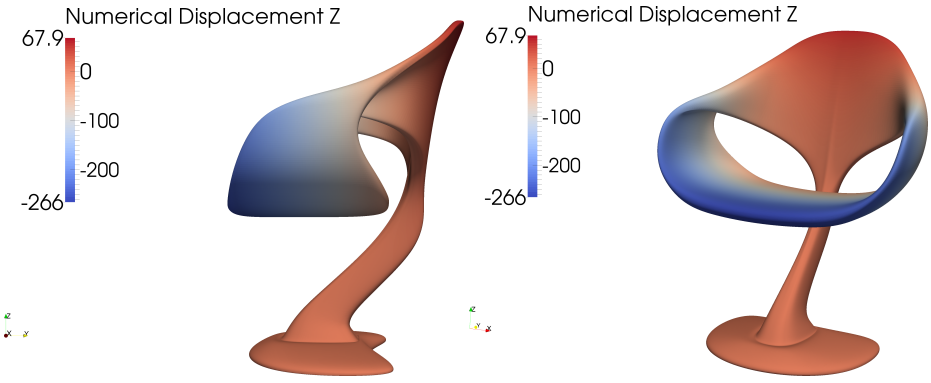
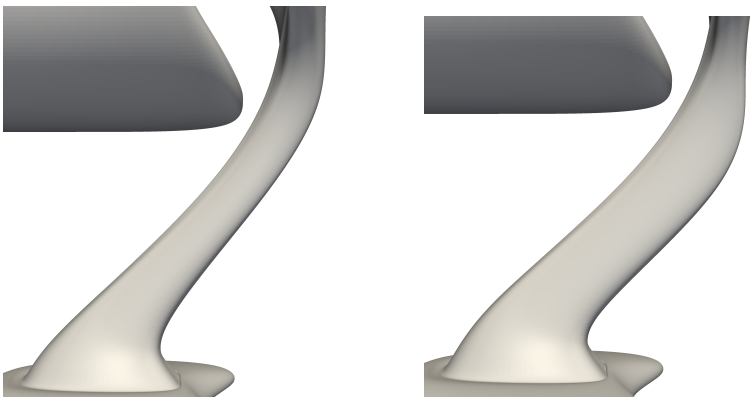


Figure 31: The optimized shape of the chair



(a) (b)

Figure 32: Close-up image of the deformed section in chair shape optimization: (a) initial geometry, and (b) optimized geometry

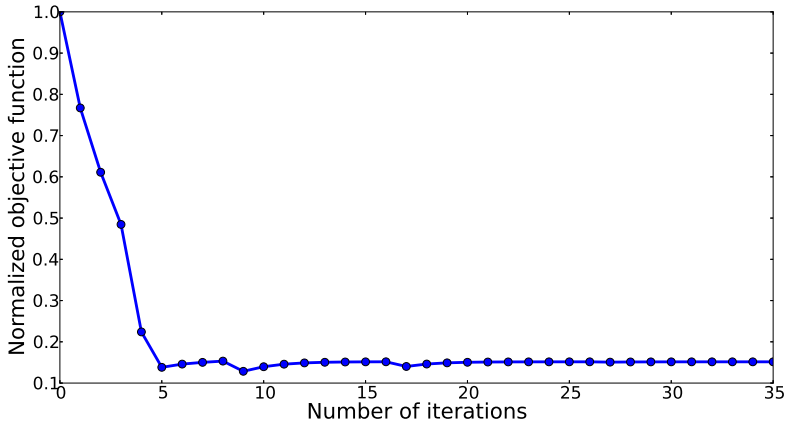


Figure 33: The iterative procedure of the chair optimization

7. Conclusions

We formulated and demonstrated a gradient-based shape optimization for three dimensional elasticity, which requires no mesh generation and produces an output geometry which is directly readable by CAD software. The geometry is represented using T-splines, thus guaranteeing a water-tight and locally refined geometry. In addition, the T-splines are also used to discretize the shape differentiation form of the regularized boundary integral equation. The shape sensitivity analysis is based on the exact geometry provided by CAD, thereby removing geometric errors and improving the accuracy. Through numerical examples, the accuracy of IGABEM in shape sensitivity analysis was clearly demonstrated. The control points are chosen as design variables to construct a smoothly varying geometry. In all the optimization numerical examples, the meshing/remeshing procedure was avoided completely, exhibiting the advantage of IGABEM arising from the integration of CAD and analysis. Moreover, the optimized geometry can be directly returned to the designer without any postprocessing as a CAD model. It is also noted that our optimization algorithm is gradient-based, so it has a solid mathematical foundation compared with gradient-less or heuristic optimization approach.

An adaptive local refinement combined with error estimation is to be developed in future. This will require the use of independent basis functions for the geometry and the field variables, within the context of Geometry Independent Field approximaTions (GIFT) [85].

From a more practical side, future work also includes the involvement of anisotropic behaviour as well as a coupling of the method with a material optimization process, for example to optimize ply orientation in composite structures in parallel to the shape of the structure itself.

Acknowledgement

The authors are sincerely grateful for the support of the European Research Council Starting Independent Research Grant (ERC stg grant agreement No.

279578) entitled “Towards real time multiscale simulation of cutting in non-linear materials with applications to surgical simulation and computer guided surgery”. The authors also appreciate the help of Dr. Robert Simpson and Dr. Michael Scott in the IGABEM codes and the T-spline model construction.

References

[1] R. N. Simpson, S. P. A. Bordas, J. Trevelyan, T. Rabczuk, A two-dimensional isogeometric boundary element method for elastostatic analysis, Computer Methods in Applied Mechanics and Engineering 209-212 (2012) 87–100.

[2] 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.

[3] R. N. Simpson, M. A. Scott, M. Taus, D. C. Thomas, H. Lian, Acoustic isogeometric boundary element analysis, Computer Methods in Applied Mechanics and Engineering 269 (2014) 265–290.

[4] M. P. Bendsøe, Optimal shape design as a material distribution problem, Structural Optimization 1 (4) (1989) 193–202.

[5] O. Sigmund, J. Petersson, Numerical instabilities in topology optimization: a survey on procedures dealing with checkerboards, mesh-dependencies and local minima, Structural Optimization 16 (1) (1998) 68–75.

[6] M. P. Bendsøe, O. Sigmund, Topology optimization: theory, methods and applications, Springer, 2003.

[7] O. C. Zienkiewicz, R. L. Taylor, The Finite Element Method, McGraw-hill London, 1977.

[8] T. A. Cruse, A direct formulation and numerical solution of the general transient elastodynamic problem. II, Journal of Mathematical Analysis and Applications 22 (2) (1968) 341–355.

[9] T. A. Cruse, Numerical solutions in three dimensional elastostatics, International Journal of Solids and Structures 5 (1969) 1259–1274.

[10] F. J. Rizzo, An integral equation approach to boundary value problems of classical elastostatics, Quarterly of Applied Mathematics 25 (1) (1967) 83–95.

[11] M. A. Jaswon, Integral equation methods in potential theory. I, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 275 (1360) (1963) 23–32.

[12] G. T. Symm, Integral equation methods in potential theory. II, Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences 275 (1360) (1963) 33–46.

[13] T. Belytschko, Y. Y. Lu, L. Gu, Element-free Galerkin methods, International Journal for Numerical Methods in Engineering 37 (1994) 229–256.

[14] W. Liu, S. Jun, Y. Zhang, Reproducing kernel particle methods, International Journal for Numerical Methods in Engineering 20 (1995) 1081–1106.

[15] S. N. Atluri, T. Zhu, A new meshless local Petrov-Galerkin (MLPG) approach in computational mechanics, Computational Mechanics 22 (2) (1998) 117–127.

[16] V. Nguyen, T. Rabczuk, S. Bordas, M. Duflot, Meshless methods: A review and computer implementation aspects, Mathematics and Computers in Simulation 79 (3) (2008) 763–813.

[17] D.-A. Hu, Y.-G. Wang, Y.-Y. Li, X. Han, Y.-T. Gu, A meshfree-based local Galerkin method with condensation of degree of freedom for elastic dynamic analysis, Acta Mechanica Sinica 30 (1) (2014) 92–99.

[18] F. Bobaru, S. Mukherjee, Meshless approach to shape optimization of linear thermoelastic solids, International Journal for Numerical Methods in Engineering 53 (4) (2002) 765–796.

[19] F. Bobaru, S. Mukherjee, Shape sensitivity analysis and shape optimization in planar elasticity using the element-free Galerkin method, Computer Methods in Applied Mechanics and Engineering 190 (32) (2001) 4319–4337.

[20] Z. Q. Zhang, J. X. Zhou, N. Zhou, X. M. Wang, L. Zhang, Shape optimization using reproducing kernel particle method and an enriched genetic algorithm, Computer Methods in Applied Mechanics and Engineering 194 (39) (2005) 4048–4070.

[21] T. Belytschko, T. Black, Elastic crack growth in finite elements with minimal remeshing, International Journal for Numerical Methods in Engineering 45 (5) (1999) 601–620.

[22] J. Dolbow, T. Belytschko, Numerical integration of the Galerkin weak form in meshfree methods, Computational Mechanics 23 (3) (1999) 219–230.

[23] S. P. A. Bordas, B. Moran, Enriched finite elements and level sets for damage tolerance assessment of complex structures, Engineering Fracture Mechanics 73 (9) (2006) 1176–1201.

[24] S. P. A. Bordas, P. V. Nguyen, C. Dunant, A. Guidoum, H. Nguyen-Dang, An extended finite element library, International Journal for Numerical Methods in Engineering 71 (6) (2007) 703–732.

[25] S. C. Wu, S. Q. Zhang, Z. W. Xu, Thermal crack growth-based fatigue life prediction due to braking for a high-speed railway brake disc, International Journal of Fatigue 87 (2016) 359–369.

[26] X. Peng, S. Kulasegaram, S. C. Wu, S. P. A. Bordas, An extended finite element method (XFEM) for linear elastic fracture with smooth nodal stress., Computers and Structures. In Press.

[27] P. Duysinx, L. Van Miegroet, T. Jacobs, C. Fleury, Generalized shape optimization using XFEM and level set methods, in: IUTAM Symposium on Topological Design Optimization of Structures, Machines and Materials, Springer, 2006, pp. 23–32.

[28] L. Van Miegroet, P. Duysinx, Stress concentration minimization of 2D filets using X-FEM and level set description, Structural and Multidisciplinary Optimization 33 (4-5) (2007) 425–438.

[29] S. Kreissl, K. Maute, Levelset based fluid topology optimization using the extended finite element method, Structural and Multidisciplinary Optimization 46 (3) (2012) 311–326.

[30] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer Methods in Applied Mechanics and Engineering 194 (39–41) (2005) 4135–4195.

[31] V. P. Nguyen, C. Anitescu, S. P. A. Bordas, T. Rabczuk, Isogeometric analysis: an overview and computer implementation aspects, Mathematics and Computers in Simulation 117 (2015) 89–116.

[32] K. M. Mathisen, K. M. Okstad, T. Kvamsdal, S. B. Raknes, Isogeometric analysis of finite deformation nearly incompressible solids, Rakenteiden Mekaniikka (Journal of Structural Mechanics) 44 (3) (2011) 260–278.

[33] D. J. Benson, Y. Bazilevs, M. C. Hsu, T. J. R. Hughes, Isogeometric shell analysis: The Reissner-Mindlin shell, Computer Methods in Applied Mechanics and Engineering 199 (5-8) (2010) 276–289.

[34] D. J. Benson, Y. Bazilevs, E. De Luycker, M. C. Hsu, M. A. Scott, T. J. R. Hughes, T. Belytschko, A generalized finite element formulation for arbitrary basis functions: from isogeometric analysis to XFEM, International Journal for Numerical Methods in Engineering 83 (2010) 765–785.

[35] J. A. Cottrell, A. Reali, Y. Bazilevs, T. J. R. Hughes, Isogeometric analysis of structural vibrations, Computer methods in applied mechanics and engineering 195 (41-43) (2006) 5257–5296.

[36] I. Temizer, P. Wriggers, T. J. R. Hughes, Contact treatment in isogeometric analysis with NURBS, Computer Methods in Applied Mechanics and Engineering 200 (9) (2011) 1100–1112.

[37] L. De Lorenzis, I. Temizer, P. Wriggers, G. Zavarise, A large deformation frictional contact formulation using NURBS-based isogeometric analysis,

International Journal for Numerical Methods in Engineering 87 (13) (2011) 1278–1300.

[38] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, Y. Zhang, Isogeometric fluid-structure interaction: Theory, algorithms, and computations, Computational Mechanics 43 (2008) 3–37.

[39] A. Buffa, G. Sangalli, R. Vázquez, Isogeometric analysis in electromagnetics: B-splines approximation, Computer Methods in Applied Mechanics and Engineering 199 (17-20) (2010) 1143–1152.

[40] W. A. Wall, M. A. Frenzel, C. Cyron, Isogeometric structural shape optimization, Computer Methods in Applied Mechanics and Engineering 197 (33) (2008) 2976–2988.

[41] S. Cho, S. H. Ha, Isogeometric shape design optimization: exact geometry and enhanced sensitivity, Structural and Multidisciplinary Optimization 38 (1) (2009) 53–70.

[42] S.-H. Ha, K. Choi, S. Cho, Numerical method for shape optimization using T-spline based isogeometric method, Structural and Multidisciplinary Optimization 42 (3) (2010) 417–428.

[43] N. D. Manh, A. Evgrafov, A. R. Gersborg, J. Gravesen, Isogeometric shape optimization of vibrating membranes, Computer Methods in Applied Mechanics and Engineering 200 (13) (2011) 1343–1353.

[44] X. Qian, Full analytical sensitivities in NURBS based isogeometric shape optimization, Computer Methods in Applied Mechanics and Engineering 199 (29-32) (2010) 2059–2071.

[45] K. Bandara, F. Cirak, Isogeometric shape optimisation of shell structures using multiresolution subdivision surfaces, arXiv preprint arXiv:1605.06288.

[46] V. P. Nguyen, P. Kerfriden, M. Brino, S. P. A. Bordas, E. Bonisoli, Nitsches method for two and three dimensional NURBS patch coupling, Computational Mechanics 53 (6) (2014) 1163–1182.

[47] M. Ruess, D. Schillinger, A. I. Özcan, E. Rank, Weak coupling for isogeometric analysis of non-matching and trimmed multi-patch geometries, Computer Methods in Applied Mechanics and Engineering 269 (2014) 46–71.

[48] M. Aigner, C. Heinrich, B. Juettler, E. Pilgerstorfer, B. Simeon, A. V. Vuong, Swept volume parameterization for isogeometric analysis, in: Hancock, E.R. and Martin, R.R. and Sabin, M.A. (Ed.), Mathematics of Surfaces XIII, Vol. 5654 of Lecture Notes in Computer Science, 2009, pp. 19–44.

[49] T. Takacs, B. Jüttler, Existence of stiffness matrix integrals for singularly parameterized domains in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 200 (49) (2011) 3568–3582.

[50] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Parameterization of computational domain in isogeometric analysis: methods and comparison, Computer Methods in Applied Mechanics and Engineering 200 (23) (2011) 2021–2031.

[51] G. Xu, B. Mourrain, R. Duvigneau, A. Galligo, Constructing analysis-suitable parameterization of computational domain from CAD boundary by variational harmonic method, Journal of Computational Physics 252 (2013) 275–289.

[52] T. W. Sederberg, J. Zheng, A. Bakenov, A. Nasri, T-splines and T-NURCCs, ACM Transactions on Graphics 22 (2003) 477–484.

[53] Y. Bazilevs, V. M. Calo, J. A. Cottrell, J. A. Evans, T. J. R. Hughes, S. Lipton, M. A. Scott, T. W. Sederberg, Isogeometric analysis using T-

splines, Computers and Mathematics with Applications 199 (5–8) (2010) 229–263.

[54] M. Dörfel, B. Jüttler, B. Simeon, Adaptive isogeometric analysis by local h-refinement with T-splines, Computer Methods in Applied Mechanics and Engineering 199 (5-8) (2010) 264–275.

[55] R. Dimitri, L. De Lorenzis, M. A. Scott, P. Wriggers, R. L. Taylor, G. Zavarise, Isogeometric large deformation frictionless contact using T-splines, Computer Methods in Applied Mechanics and Engineering 269 (2014) 394–414.

[56] N. Nguyen-Thanh, H. Nguyen-Xuan, S. P. A. Bordas, T. Rabczuk, Isogeometric analysis using polynomial splines over hierarchical T-meshes for two-dimensional elastic solids, Computer Methods in Applied Mechanics and Engineering 200 (21) (2011) 1892–1908.

[57] P. Wang, J. Xu, J. Deng, F. Chen, Adaptive isogeometric analysis using rational PHT-splines, Computer-Aided Design 43 (11) (2011) 1438–1448.

[58] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wüchner, K. Bletzinger, Y. Bazilevs, T. Rabczuk, Rotation free isogeometric thin shell analysis using PHT-splines, Computer Methods in Applied Mechanics and Engineering 200 (47) (2011) 3410–3424.

[59] D. R. Forsey, R. H. Bartels, Hierarchical B-spline refinement, ACM SIGGRAPH Computer Graphics 22 (4) (1988) 205–212.

[60] A.-V. Vuong, C. Giannelli, B. Jüttler, B. Simeon, A hierarchical approach to adaptive local refinement in isogeometric analysis, Computer Methods in Applied Mechanics and Engineering 200 (49) (2011) 3554–3567.

[61] T. Dokken, V. Skytt, Locally refined splines, Preprint.

[62] C. Politis, A. I. Ginnis, P. D. Kaklis, K. Belibassakis, C. Feurer, An isogeometric BEM for exterior potential-flow problems in the plane, in: Proceed-

ings of SIAM/ACM joint conference on geometric and physical modeling, 2009.

[63] R. N. Simpson, S. P. A. Bordas, H. Lian, J. Trevelyan, An isogeometric boundary element method for elastostatic analysis: 2D implementation aspects, Computers & Structures 118 (2013) 2–12.

[64] H. Lian, R. N. Simpson, S. Bordas, Stress analysis without meshing: Isogeometric boundary-element method, Proceedings of the Institution of Civil Engineers: Engineering and Computational Mechanics 166 (2) (2013) 88–99.

[65] X. Peng, E. Atroshchenko, P. Kerfriden, S. Bordas, Isogeometric boundary element methods for three dimensional static fracture and fatigue crack growth, Computer Methods in Applied Mechanics and Engineering doi:10.1016/j.cma.2016.05.038.

[66] H. Lian, P. Kerfriden, S. Bordas, Implementation of regularized isogeometric boundary element methods for gradient-based shape optimization in two-dimensional linear elasticity, International Journal for Numerical Methods in Engineering 106 (12) (2016) 972–1017.

[67] X. Peng, E. Atroshchenko, P. Kerfriden, S. Bordas, Linear elastic fracture simulation directly from cad: 2D NURBS-based implementation and role of tip enrichment, International Journal of Fracture doi:10.1007/s10704-016-0153-3.

[68] F. Auricchio, L. B. Da Veiga, T. J. R. Hughes, A. Reali, G. Sangalli, Isogeometric collocation methods, Mathematical Models and Methods in Applied Sciences 20 (11) (2010) 2075–2107.

[69] D. Schillinger, J. A. Evans, A. Reali, M. A. Scott, T. J. R. Hughes, Isogeometric collocation: cost comparison with Galerkin methods and extension to adaptive hierarchical NURBS discretizations, Computer Methods in Applied Mechanics and Engineering 267 (2013) 170–232.

[70] L. De Lorenzis, J. Evans, T. Hughes, A. Reali, Isogeometric collocation: Neumann boundary conditions and contact, Computer Methods in Applied Mechanics and Engineering 284 (2015) 21–54.

[71] K. Li, X. Qian, Isogeometric analysis and shape optimization via boundary integral, Computer-Aided Design 43 (11) (2011) 1427–1437.

[72] K. Kostas, A. Ginnis, C. Politis, P. Kaklis, Ship-hull shape optimization with a T-spline based BEM–isogeometric solver, Computer Methods in Applied Mechanics and Engineering 284 (2015) 611–622.

[73] K. Bandara, F. Cirak, G. Of, O. Steinbach, J. Zapletal, Boundary element based multiresolution shape optimisation in electrostatics, Journal of Computational Physics 297 (2015) 584–598.

[74] Y. Liu, T. J. Rudolphi, Some identities for fundamental solutions and their applications to weakly-singular boundary element formulations, Engineering Analysis with Boundary Elements 8 (6) (1991) 301–311.

[75] Y. Liu, T. J. Rudolphi, New identities for fundamental solutions and their applications to non-singular boundary element formulations, Computational Mechanics 24 (4) (1999) 286–292.

[76] Y. Liu, On the simple-solution method and non-singular nature of the BIE/BEM–a review and some new results, Engineering Analysis with Boundary Elements 24 (10) (2000) 789–795.

[77] M. A. Scott, M. J. Borden, C. V. Verhoosel, T. W. Sederberg, T. J. Hughes, Isogeometric finite element data structures based on Bézier extraction of T-splines, International Journal for Numerical Methods in Engineering 88 (2) (2011) 126–156.

[78] A. Buffa, D. Cho, G. Sangalli, Linear independence of the T-spline blending functions associated with some particular T-meshes, Computer Methods in Applied Mechanics and Engineering 199 (23-24) (2010) 1437–1445.

[79] X. Li, J. Zheng, T. W. Sederberg, T. J. R. Hughes, M. A. Scott, On linear independence of T-spline blending functions, Computer Aided Geometric Design 29 (2012) 63 – 76.

[80] M. J. Borden, M. A. Scott, J. A. Evans, T. J. R. Hughes, Isogeometric finite element data structures based on Bézier extraction of NURBS, International Journal for Numerical Methods in Engineering 87 (1-5) (2011) 15–47.

[81] M. Guiggiani, The evaluation of cauchy principal value integrals in the boundary element method–a review, Mathematical and Computer Modelling 15 (3) (1991) 175–184.

[82] T-Splines, Inc., http://www.tsplines.com/rhino (2011).

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

[84] T-Splines, Inc., http://www.tsplines.com/tsplinemodels (2011).

[85] G. Xu, E. Atroshchenko, S. P. A. Bordas, Geometry independent field approximation for spline-based finite element methods, 11^{th} World Congress on Computational Mechanics, Barcelona, 2014.

[86] A. Carle, M. Fagan, L. L. Green, Preliminary results from the application of automated adjoint code generation to CFL3D, AIAA paper 4807 (1998) 1998–4807.

[87] K. D. Bhalerao, M. Poursina, K. S. Anderson, An efficient direct differentiation approach for sensitivity analysis of flexible multibody systems, Multibody System Dynamics 23 (2) (2010) 121–140.

Appendix A. The fundamental solution sensitivities

The shape derivatives of fundamental solutions play a key role in shape sensitivity analysis in IGABEM. The analytical form can be obtained by taking shape derivatives on fundamental solutions (Eqs. (10, 11)) as

$$\begin{aligned}\dot{U}_{ij}\left(\mathbf{s},\ \mathbf{x}\right) = &\frac{1}{16\pi\mu(1-\nu)}\left\{\left(\dot{\frac{1}{r}}\right)[(3-4\nu)\delta_{ij}+r_{,i}r_{,j}]\right\} \\ &+\frac{1}{16\pi\mu(1-\nu)}\left\{\frac{1}{r}[(\dot{r_{,i}})r_{,j}+r_{,i}(\dot{r_{,j}})]\right\},\end{aligned} \tag{75}$$

$$\begin{aligned}\dot{T}_{ij}\left(\mathbf{s},\ \mathbf{x}\right) = &\frac{-1}{8\pi(1-\nu)}\left(\dot{\frac{1}{r^2}}\right)\left\{\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij}+3r_{,i}r_{,j}]\right\} \\ &+\frac{-1}{8\pi(1-\nu)}\left(\dot{\frac{1}{r^2}}\right)\{(1-2\nu)(n_i r_{,j}-n_j r_{,i})\} \\ &+\frac{-1}{8\pi(1-2\nu)r^2}\left\{\left(\frac{\dot{\partial r}}{\partial n}\right)[(1-2\nu)\delta_{ij}+3r_{,i}r_{,j}]\right\} \\ &+\frac{-1}{8\pi(1-2\nu)r^2}\left\{3\frac{\partial r}{\partial n}[(\dot{r_{,i}})r_{,j}+r_{,i}(\dot{r_{,j}})]\right\} \\ &+\frac{-1}{8\pi(1-2\nu)r^2}\{(1-2\nu)[\dot{n}_i r_{,j}+n_i(\dot{r_{,j}})-\dot{n}_j r_{,i}-n_j(\dot{r_{,i}})]\},\end{aligned} \tag{76}$$

where

$$\left(\dot{\frac{1}{r}}\right) = -\frac{\dot{r}}{r^2}, \quad \left(\dot{\frac{1}{r^2}}\right) = -\frac{2\dot{r}}{r^3}, \tag{77}$$

$$(\dot{r_{,i}}) = \left(\frac{\dot{x_i - s_i}}{r}\right) = \frac{(\dot{x}_i-\dot{s}_i)r-(x_i-s_i)\dot{r}}{r^2}, \tag{78}$$

$$\left(\frac{\dot{\partial r}}{\partial n}\right) = (\dot{r_{,i}n_i}) = (\dot{r_{,i}})n_i + r_{,i}\dot{n}_i, \tag{79}$$

$$\dot{r} = \left[\dot{\sqrt{(x_i-s_i)(x_i-s_i)}}\right] = \frac{(\dot{x_i-s_i})(x_i-s_i)}{r} = \frac{(\dot{x}_i-\dot{s}_i)(x_i-s_i)}{r}. \tag{80}$$

The sensitivity of the unit outward normal n_i on the boundary is

$$\dot{n}_i = \left[\frac{\dot{J_i(\tilde{\xi})}}{J(\tilde{\xi})}\right] = \frac{\dot{J}_i(\tilde{\boldsymbol{\xi}})J(\tilde{\boldsymbol{\xi}})-J_i(\tilde{\boldsymbol{\xi}})\dot{J}(\tilde{\boldsymbol{\xi}})}{J^2(\tilde{\boldsymbol{\xi}})}, \tag{81}$$

where J is the Jacobian determinant. Its expression and shape derivatives are given by

$$J_i(\tilde{\boldsymbol{\xi}}) = \varepsilon_{ijk}\frac{\partial x_j}{\partial\tilde{\xi}_1}\frac{\partial x_k}{\partial\tilde{\xi}_2}, \tag{82}$$

$$J(\tilde{\boldsymbol{\xi}}) = \sqrt{J_i(\tilde{\boldsymbol{\xi}}) J_i(\tilde{\boldsymbol{\xi}})}, \tag{83}$$

$$\dot{J}_i(\tilde{\boldsymbol{\xi}}) = \varepsilon_{ijk} \left(\frac{\partial \dot{x}_j}{\partial \tilde{\xi}_1} \right) \frac{\partial x_k}{\partial \tilde{\xi}_2} + \varepsilon_{ijk} \frac{\partial x_j}{\partial \tilde{\xi}_1} \left(\frac{\partial \dot{x}_k}{\partial \tilde{\xi}_2} \right), \tag{84}$$

$$\dot{J}(\tilde{\boldsymbol{\xi}}) = \frac{\dot{J}_i(\tilde{\boldsymbol{\xi}}) J_i(\tilde{\boldsymbol{\xi}})}{J(\tilde{\boldsymbol{\xi}})}, \tag{85}$$

with ε_{ijk} the permutation operator

$$\varepsilon_{ijk} = \begin{cases} 1 & \text{for cyclic suffix order : 123, 231, 312,} \\ -1 & \text{for cyclic suffix order : 132, 213, 321,} \\ 0 & \text{if any two indices are the same.} \end{cases} \tag{86}$$

The hypersingular fundamental solutions are

$$D_{kij} = \frac{1}{8\pi(1-\nu)r^2}[(1-2\nu)(\delta_{ik}r_{,j} + \delta_{jk}r_{,i} - \delta_{ij}r_{,k}) + 3r_{,i}r_{,j}r_{,k}], \tag{87}$$

$$\begin{aligned} S_{kij} = {} & \frac{\mu}{4\pi(1-\nu)r^3} \left\{ 3\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij}r_{,k} + \nu(r_{,j}\delta_{ik} + r_{,i}\delta_{jk}) - 5r_{,i}r_{,j}r_{,k}] \right\} \\ & + \frac{\mu}{4\pi(1-\nu)r^3}\{3\nu(n_i r_{,j}r_{,k} + n_j r_{,i}r_{,k})\} \\ & + \frac{\mu}{4\pi(1-\nu)r^3}\{(1-2\nu)(3n_k r_{,i}r_{,j} + n_j\delta_{ik} + n_i\delta_{jk}) - (1-4\nu)n_k\delta_{ij}\}. \end{aligned} \tag{88}$$

Therefore the sensitivities of the hypersingular fundamental solutions are given by

$$\begin{aligned} \dot{D}_{kij} = {} & \frac{1}{8\pi(1-\nu)} \left(\frac{\dot{1}}{r^2} \right) [(1-2\nu)(\delta_{ik}r_{,j} + \delta_{jk}r_{,i} - \delta_{ij}r_{,k}) + 3r_{,i}r_{,j}r_{,k}] \\ & + \frac{1}{8\pi(1-\nu)r^2}[(1-2\nu)(\delta_{ik}(\dot{r_{,j}}) + \delta_{jk}(\dot{r_{,i}}) - \delta_{ij}(\dot{r_{,k}}))] \\ & + \frac{1}{8\pi(1-\nu)r^2}[3((\dot{r_{,i}})r_{,j}r_{,k} + r_{,i}(\dot{r_{,j}})r_{,k} + r_{,i}r_{,j}(\dot{r_{,k}}))], \end{aligned} \tag{89}$$

$$\dot{S}_{kij} = \dot{S}^1_{kij}(\mathbf{s},\mathbf{x}) + \dot{S}^2_{kij}(\mathbf{s},\ \mathbf{x}) + \dot{S}^3_{kij}(\mathbf{s},\mathbf{x}) + \dot{S}^4_{kij}(\mathbf{s},\mathbf{x}), \tag{90}$$

with

$$\begin{aligned} \dot{S}^1_{kij} = {} & \frac{\mu}{4\pi(1-\nu)} \left(\frac{\dot{1}}{r^3} \right) \left\{ 3\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij}r_{,k} + \nu(r_{,j}\delta_{ik} + r_{,i}\delta_{jk}) - 5r_{,i}r_{,j}r_{,k}] \right\} \\ & + \frac{\mu}{4\pi(1-\nu)} \left(\frac{\dot{1}}{r^3} \right) \{3\nu(n_i r_{,j}r_{,k} + n_j r_{,i}r_{,k})\} \end{aligned} \tag{91}$$

$$+\frac{\mu}{4\pi(1-\nu)}\left(\frac{\dot{1}}{r^3}\right)\{(1-2\nu)(3n_k r_{,i} r_{,j}+n_j\delta_{ik}+n_i\delta_{jk})-(1-4\nu)n_k\delta_{ij}\},$$

$$\begin{aligned}\dot{S}^2_{kij} = &\frac{\mu}{4\pi(1-\nu)r^3}\left\{3\left(\frac{\dot{\partial r}}{\partial n}\right)[(1-2\nu)\delta_{ij}r_{,k}+\nu(r_{,j}\delta_{ik}+r_{,i}\delta_{jk})-5r_{,i}r_{,j}r_{,k}]\right\}\\ &+\frac{\mu}{4\pi(1-\nu)r^3}\left\{3\frac{\partial r}{\partial n}[(1-2\nu)\delta_{ij}(\dot{r_{,k}})+\nu((\dot{r_{,j}})\delta_{ik}+(\dot{r_{,i}})\delta_{jk})]\right\}\\ &-\frac{\mu}{4\pi(1-\nu)r^3}\left\{3\frac{\partial r}{\partial n}[5((\dot{r_{,i}})r_{,j}r_{,k}+r_{,i}(\dot{r_{,j}})r_{,k}+r_{,i}r_{,j}(\dot{r_{,k}}))]\right\},\end{aligned} \tag{92}$$

$$\begin{aligned}\dot{S}^3_{kij} = &\frac{\mu}{4\pi(1-\nu)r^3}\{3\nu(\dot{n}_i r_{,j}r_{,k}+n_i(\dot{r_{,j}})r_{,k}+n_i r_{,j}(\dot{r_{,k}}))\}\\ &+\frac{\mu}{4\pi(1-\nu)r^3}\{3\nu(\dot{n}_j r_{,i}r_{,k}+n_j(\dot{r_{,i}})r_{,k}+n_j r_{,i}(\dot{r_{,k}}))\},\end{aligned} \tag{93}$$

$$\begin{aligned}\dot{S}^4_{kij} = &\frac{\mu}{4\pi(1-\nu)r^3}\{(1-2\nu)(3\dot{n}_k r_{,i}r_{,j}+3n_k(\dot{r_{,i}})r_{,j}+3n_k r_{,i}(\dot{r_{,j}}))\}\\ &+\frac{\mu}{4\pi(1-\nu)r^3}\{(1-2\nu)(\dot{n}_j\delta_{ik}+\dot{n}_i\delta_{jk})-(1-4\nu)\dot{n}_k\delta_{ij}\}.\end{aligned} \tag{94}$$

Tab. 5 shows the singularity order of the fundamental solution sensitivities. We can see that they have the same order as the fundamental solutions.

Kernel	Kernel sensitivity	Order	Singularity type	Dimension
U_{ij}	$\dot{U}_{ij}$	$O(1/r)$	weakly singular	3D
T_{ij}	$\dot{T}_{ij}$	$O(1/r^2)$	strongly singular	3D
D_{ij}	$\dot{D}_{kij}$	$O(1/r^2)$	strongly singular	3D
S_{ij}	$\dot{S}_{kij}$	$O(1/r^3)$	hypersingular	3D

Table 5: The singularity of kernel function sensitivities

It is noted that the above analytical differentiation can be bypassed by using automatic differentiation. However, analytical derivation still possesses the advantage in the computational efficiency and numerical stability [86, 87].

Appendix B. Shape sensitivities recovery

Appendix B.1. Evaluating shape sensitivities at interior points

By ignoring body forces, for interior points $\mathbf{S}$, the displacement is

$$u_i(\mathbf{S}) = \int_S U_{ij}(\mathbf{S},\ \mathbf{x}) t_j(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) - \int_S T_{ij}(\mathbf{S},\ \mathbf{x}) u_j(\mathbf{x})\,\mathrm{d}S(\mathbf{x}), \tag{95}$$

and the stress writes:

$$\sigma_{ij}(\mathbf{S}) = \int_S D_{kij}(\mathbf{S},\ \mathbf{x}) t_k(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) - \int_S S_{kij}(\mathbf{S},\mathbf{x})\,\mathbf{x}) u_k \mathrm{d}S(\mathbf{x}). \tag{96}$$

By taking shape derivatives with respect to the design variables, the displacement shape sensitivity is

$$\begin{aligned}\dot{u}_i(\mathbf{S}) = &\int_S \dot{U}_{ij}(\mathbf{S},\mathbf{x})\, t_j(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) + \int_\Omega U_{ij}(\mathbf{S},\mathbf{x})\,\dot{t}_j(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) \\ &+ \int_\Omega U_{ij}(\mathbf{S},\mathbf{x})\, t_j(\mathbf{x})\,[\mathrm{d}\dot{S}(\mathbf{x})] - \int_S \dot{T}_{ij}(\mathbf{S},\mathbf{x})\, u_j(\mathbf{x})(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) \\ &- \int_S T_{ij}(\mathbf{S},\mathbf{x})\,\dot{u}_j(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) - \int_S T_{ij}(\mathbf{S},\mathbf{x})\, u_j(\mathbf{x})\,[\mathrm{d}\dot{S}(\mathbf{x})],\end{aligned} \tag{97}$$

and the stress shape sensitivity is

$$\begin{aligned}\dot{\sigma}_{ij}(\mathbf{S}) = &\int_S \dot{D}_{kij}(\mathbf{S},\mathbf{x})\, t_k(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) + \int_S D_{kij}(\mathbf{S},\mathbf{x})\,\dot{t}_k(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) \\ &+ \int_S D_{kij}(\mathbf{S},\mathbf{x})\, t_k(\mathbf{x})\,[\mathrm{d}\dot{S}(\mathbf{x})] - \int_S \dot{S}_{kij}(\mathbf{S},\ \mathbf{x}) u_k(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) \\ &- \int_S S_{kij}(\mathbf{S},\ \mathbf{x})\dot{u}_k(\mathbf{x})\,\mathrm{d}S(\mathbf{x}) - \int_S S_{kij}(\mathbf{S},\ \mathbf{x}) u_k(\mathbf{x})\,[\mathrm{d}\dot{S}(\mathbf{x})].\end{aligned} \tag{98}$$

Appendix B.2. Evaluating stress shape sensitivities on the boundary

For three dimensional problems, the first step is to construct a tangential coordinate system as

$$\mathbf{m}_1(\tilde{\boldsymbol{\xi}}) = \frac{\partial \mathbf{x}}{\partial \xi_1}(\tilde{\boldsymbol{\xi}}), \tag{99}$$

$$\mathbf{m}_2(\tilde{\boldsymbol{\xi}}) = \frac{\partial \mathbf{x}}{\partial \xi_2}(\tilde{\boldsymbol{\xi}}), \tag{100}$$

$$\mathbf{n}(\tilde{\boldsymbol{\xi}}) = \mathbf{m}_1(\tilde{\boldsymbol{\xi}}) \times \mathbf{m}_2(\tilde{\boldsymbol{\xi}}), \tag{101}$$

where $\mathbf{m}_1$ and $\mathbf{m}_2$ are the two tangent vectors, and $\mathbf{n}$ the normal vector to the surface. The tangential coordinate system is neither orthogonal nor normal

generally, which is a subtle difference from two-dimensional problems. So we need to establish an unit orthogonal coordinate system based on the tangential coordinate system (Fig. 34), and its three basis vectors are given by

$$\hat{\mathbf{e}}_1 = \frac{\mathbf{m}_1}{|\mathbf{m}_1|}, \tag{102}$$

$$\hat{\mathbf{e}}_3 = \frac{\mathbf{n}}{|\mathbf{n}|}, \tag{103}$$

$$\hat{\mathbf{e}}_2 = \hat{\mathbf{e}}_1 \times \hat{\mathbf{e}}_3. \tag{104}$$

The rotation tensor A_{ij} for the coordinates system transition can be written in matrix form

$$A = \begin{bmatrix} \hat{\mathbf{e}}_1 \\ \hat{\mathbf{e}}_2 \\ \hat{\mathbf{e}}_3 \end{bmatrix}. \tag{105}$$

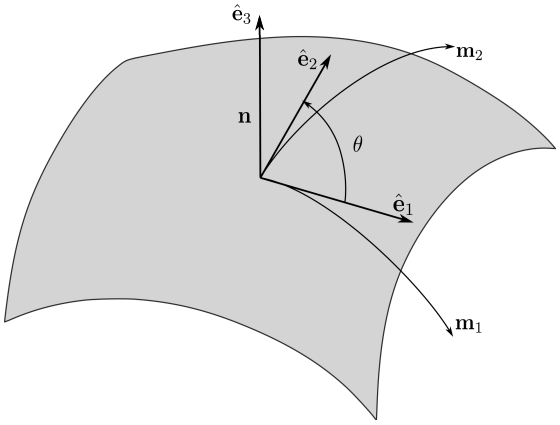


Figure 34: Local coordinate system on surface

We can also get the derivatives of the intrinsic coordinates of parent element with respect to that of local orthogonal system

$$\frac{\partial \tilde{\xi}_1}{\partial \hat{x}_1} = \frac{1}{|\mathbf{m}_1|}, \quad \frac{\partial \tilde{\xi}_1}{\partial \hat{x}_2} = \frac{-\cos\theta}{|\mathbf{m}_1|\sin\theta}, \tag{106}$$

$$\frac{\partial \tilde{\xi}_2}{\partial \hat{x}_1} = 0, \qquad \frac{\partial \tilde{\xi}_2}{\partial \hat{x}_2} = \frac{1}{|\mathbf{m}_2|}\sin\theta, \tag{107}$$

where $\hat{x}_1$, $\hat{x}_2$ and $\hat{x}_3$ denote the local orthogonal coordinates.

The strain components in the $\hat{\mathbf{e}}_1$-$\hat{\mathbf{e}}_2$ of the local orthogonal system is

$$\hat{\epsilon}_{ij} = \frac{\partial \hat{u}_i}{\partial \hat{x}_j} = \frac{\partial \hat{u}_i}{\partial \tilde{\xi}_k}\frac{\partial \tilde{\xi}_k}{\partial \hat{x}_j} \qquad i,j,k = 1,2, \tag{108}$$

with

$$\frac{\partial \hat{u}_i}{\partial \tilde{\xi}_k} = A_{il}\frac{\partial u_l}{\partial \tilde{\xi}_k} \qquad k = 1,2, \quad \text{and} \quad i,l = 1,2,3. \tag{109}$$

From the constitutive equations and the relationships between stress and traction,

$$\hat{\sigma}_{11} = \frac{E}{1-\nu^2}(\hat{\epsilon}_{11} + \nu\hat{\epsilon}_{22}) + \frac{\nu}{1-\nu}\hat{t}_3, \tag{110}$$

$$\hat{\sigma}_{12} = \frac{E}{1+\nu}\hat{\epsilon}_{12}, \tag{111}$$

$$\hat{\sigma}_{22} = \frac{E}{1-\nu^2}(\hat{\epsilon}_{22} + \nu\hat{\epsilon}_{11}) + \frac{\nu}{1-\nu}\hat{t}_3, \tag{112}$$

$$\hat{\sigma}_{33} = \hat{t}_3, \tag{113}$$

$$\hat{\sigma}_{23} = \hat{t}_2, \tag{114}$$

$$\hat{\sigma}_{13} = \hat{t}_1. \tag{115}$$

Then we transfer the stress from the local orthogonal system to the global Cartesian system

$$\sigma_{ij} = A_{ki}A_{nj}\hat{\sigma}_{kn}. \tag{116}$$

Now we consider the stress sensitivities on the surface. The boundary strain sensitivity can be evaluated using Eqs. (108,106,107) as

$$\begin{aligned}\dot{\hat{\epsilon}}_{ij} &= \left(\dot{\frac{\partial \hat{u}_i}{\partial \hat{x}_j}}\right) = \left(\dot{\frac{\partial \hat{u}_i}{\partial \tilde{\xi}_k}}\frac{\partial \tilde{\xi}_k}{\partial \hat{x}_j}\right) = \left(A_{il}\dot{\frac{\partial u_l}{\partial \tilde{\xi}_k}}\frac{\partial \tilde{\xi}_k}{\partial \hat{x}_j}\right) \\ &= \dot{A}_{il}\frac{\partial u_l}{\partial \tilde{\xi}_k}\frac{\partial \xi_k}{\partial \hat{x}_j} + A_{il}\left(\dot{\frac{\partial u_l}{\partial \tilde{\xi}_k}}\right)\frac{\partial \xi_k}{\partial \hat{x}_j} + A_{il}\frac{\partial u_l}{\partial \tilde{\xi}_k}\left(\dot{\frac{\partial \tilde{\xi}_k}{\partial \hat{x}_j}}\right),\end{aligned} \tag{117}$$

where the shape sensitivities of the derivatives of $\tilde{\xi}_j$ with respect to $\hat{x}_j$ are

$$\left(\dot{\frac{\partial \tilde{\xi}_1}{\partial \hat{x}_1}}\right) = \left(\dot{\frac{1}{|\mathbf{m}_1|}}\right), \tag{118}$$

$$\left(\dot{\frac{\partial \tilde{\xi}_1}{\partial \hat{x}_2}}\right) = \left(\dot{\frac{-\cos\theta}{|\mathbf{m}_1|\sin\theta}}\right) = \left(\dot{\frac{1}{|\mathbf{m}_1|}}\right)\frac{-\cos\theta}{\sin\theta} + \frac{1}{|\mathbf{m}_1|}\left(\dot{\frac{-\cos\theta}{\sin\theta}}\right), \tag{119}$$

$$\left(\dot{\frac{\partial \tilde{\xi}_2}{\partial \hat{x}_1}}\right) = 0, \tag{120}$$

$$\left(\dot{\frac{\partial\tilde{\xi}_2}{\partial\hat{x}_2}}\right) = \left(\dot{\frac{1}{|\mathbf{m}_2|}}\sin\theta\right) = \left(\dot{\frac{1}{|\mathbf{m}_2|}}\right)\sin\theta + \frac{1}{|\mathbf{m}_2|}(\dot{\sin\theta}). \tag{121}$$

The shape sensitivities related to the angle θ are

$$(\dot{\cos\theta}) = \left(\frac{\dot{\mathbf{m}_1\cdot\mathbf{m}_2}}{|\mathbf{m}_1|\,|\mathbf{m}_2|}\right) = \frac{(\dot{\mathbf{m}_1\cdot\mathbf{m}_2})\,(|\mathbf{m}_1|\,|\mathbf{m}_2|) - (\mathbf{m}_1\cdot\mathbf{m}_2)\,(\dot{|\mathbf{m}_1|}\,|\mathbf{m}_2|)}{(|\mathbf{m}_1|\,|\mathbf{m}_2|)^2}, \tag{122}$$

$$\dot{\theta} = \left[\arccos\left(\frac{\dot{\mathbf{m}_1\cdot\mathbf{m}_2}}{|\mathbf{m}_1|\,|\mathbf{m}_2|}\right)\right] = \frac{-1}{1-\left(\frac{\mathbf{m}_1\cdot\mathbf{m}_2}{|\mathbf{m}_1||\mathbf{m}_2|}\right)^2}\left(\frac{\dot{\mathbf{m}_1\cdot\mathbf{m}_2}}{|\mathbf{m}_1|\,|\mathbf{m}_2|}\right), \tag{123}$$

$$(\dot{\sin\theta}) = \dot{\theta}\cos\theta, \tag{124}$$

$$\left(\dot{\frac{\cos\theta}{\sin\theta}}\right) = \frac{(\dot{\cos\theta})\sin\theta - \cos\theta(\dot{\sin\theta})}{\sin^2\theta}, \tag{125}$$

with

$$\left(\dot{\frac{1}{|\mathbf{m}_1|}}\right) = -\frac{\dot{|\mathbf{m}_1|}}{|\mathbf{m}_1|^2}, \tag{126}$$

$$\left(\dot{\frac{1}{|\mathbf{m}_2|}}\right) = -\frac{\dot{|\mathbf{m}_2|}}{|\mathbf{m}_2|^2}. \tag{127}$$

The shape sensitivities of the base vectors of the tangential coordinate system are

$$\dot{|\mathbf{m}_1|} = \left(\dot{\sqrt{\frac{\partial x_i}{\partial\tilde{\xi}_1}\frac{\partial x_i}{\partial\tilde{\xi}_1}}}\right) = \frac{\left(\frac{\partial\dot{x}_i}{\partial\tilde{\xi}_1}\right)\frac{\partial x_i}{\partial\tilde{\xi}_1}}{|\mathbf{m}_1|}, \tag{128}$$

$$\dot{|\mathbf{m}_2|} = \left(\dot{\sqrt{\frac{\partial x_i}{\partial\tilde{\xi}_2}\frac{\partial x_i}{\partial\tilde{\xi}_2}}}\right) = \frac{\left(\frac{\partial\dot{x}_i}{\partial\tilde{\xi}_2}\right)\frac{\partial x_i}{\partial\tilde{\xi}_2}}{|\mathbf{m}_2|}. \tag{129}$$

According to Hooke's law,

$$\dot{\hat{\sigma}}_{11} = \frac{E}{1-\nu^2}(\dot{\hat{\epsilon}}_{11} + \nu\dot{\hat{\epsilon}}_{22}) + \frac{\nu}{1-\nu}\dot{\hat{t}}_3, \tag{130}$$

$$\dot{\hat{\sigma}}_{12} = \frac{E}{1+\nu}\dot{\hat{\epsilon}}_{12}, \tag{131}$$

$$\dot{\hat{\sigma}}_{22} = \frac{E}{1-\nu^2}(\dot{\hat{\epsilon}}_{22} + \nu\dot{\hat{\epsilon}}_{11}) + \frac{\nu}{1-\nu}\dot{\hat{t}}_3, \tag{132}$$

$$\dot{\hat{\sigma}}_{33} = \dot{\hat{t}}_3, \tag{133}$$

$$\dot{\hat{\sigma}}_{23} = \dot{\hat{t}}_2, \tag{134}$$

$$\dot{\hat{\sigma}}_{13} = \dot{\hat{t}}_1. \tag{135}$$

The shape sensitivity of the boundary stress is finally transferred to the global Cartesian coordinate system

$$\sigma_{ij} = \dot{A}_{ki}A_{nj}\hat{\sigma}_{kn} + A_{ki}\dot{A}_{nj}\hat{\sigma}_{kn} + A_{ki}A_{nj}\dot{\hat{\sigma}}_{kn}. \tag{136}$$

Appendix C. The analytical solution of the spherical cavity problem

For the spherical cavity problem, the analytical solution of the displacement is given by

$$\begin{aligned}
2\mu u_R(R,\beta,\theta) &= -A_1 R + \frac{3}{2}\frac{A_2}{R^4} - \frac{A_3}{R^2} \\
&\quad + \left(3A_1 R - \frac{9}{2}\frac{A_2}{R^4} + B_1(4\nu-2)R + \frac{B_2(4\nu-5)}{R^2}\right)\cos\beta, \quad (137)\\
2\mu u_\beta(R,\beta,\theta) &= -\left[-3A_1 R - \frac{3A_2}{R^4} + \left(B_1 R + \frac{B_2}{R^2}\right)(2-4\nu)\right]\sin\beta\cos\beta, \quad (138)\\
u_\theta(R,\beta,\theta) &= 0, \quad (139)
\end{aligned}$$

where

$$\begin{aligned}
&A_1 = \frac{S\nu}{1+\nu}, \qquad A_2 = \frac{Sa^5}{7-5\nu}, \qquad A_3 = \frac{Sa^3(6-5\nu)}{2(7-5\nu)},\\
&B_1 = -\frac{S}{2(1+\nu)}, \quad B_2 = -\frac{5Sa^3}{2(7-5\nu)}. \quad (140)
\end{aligned}$$

The analytical stress is given by

$$\begin{aligned}
\sigma_{RR}(R,\beta,\theta) &= S\cos^2\beta + \frac{S}{7-5\nu}\left\{\frac{a^3}{R^3}[6-5(5-\nu)\cos^2\beta] + \frac{6a^5}{R^5}(3\cos^2\beta-1)\right\}\\
&\quad + \frac{S}{7-5\nu}\left\{\frac{6a^5}{R^5}(3\cos^2\beta-1)\right\}, \quad (141)\\
\sigma_{\theta\theta}(R,\beta,\theta) &= \frac{S}{2(7-5\nu)}\left\{\frac{a^3}{R^3}[5\nu-2+5(1-2\nu)\cos^2\beta] + \frac{a^5}{R^5}(1-5\cos^2\beta)\right\}\\
&\quad + \frac{S}{2(7-5\nu)}\left\{\frac{a^5}{R^5}(1-5\cos^2\beta)\right\}, \quad (142)\\
\sigma_{\beta\beta}(R,\beta,\theta) &= S\sin^2\beta + \frac{S}{2(7-5\nu)}\left\{\frac{a^3}{R^3}[4-5\nu+5(1-2\nu)\cos^2\beta] + \frac{3a^5}{R^5}(3-7\cos^2\beta)\right\}\\
&\quad + \frac{S}{2(7-5\nu)}\left\{\frac{3a^5}{R^5}(3-7\cos^2\beta)\right\}, \quad (143)\\
\sigma_{R\beta}(R,\beta,\theta) &= S\left\{-1 + \frac{1}{7-5\nu}\left[-\frac{5a^3(1+\nu)}{R^3} + \frac{12a^5}{R^5}\right]\right\}\sin\beta\cos\beta. \quad (144)
\end{aligned}$$

We take the cavity radius a as the design variable, and the analytical displacement sensitivity can be written as

$$2\mu\dot{u}_R(R,\beta,\theta) = -(A_1\dot{R} + \dot{A}_1 R) + \frac{3}{2}\left(\frac{\dot{A}_2}{R^4}\right) - \left(\frac{\dot{A}_3}{R^2}\right) + \left[3A_1\dot{R} - \frac{9}{2}\left(\frac{\dot{A}_2}{R^4}\right) + B_1(4\nu-2)\dot{R} + (4\nu-5)\left(\frac{\dot{B}_2}{R^2}\right)\right]\cos\beta, \tag{145}$$

$$2\mu\dot{u}_\beta(R,\beta,\theta) = -\left\{-3A_1\dot{R} - 3\left(\frac{\dot{A}_2}{R^4}\right) + \left[B_1\dot{R} + \dot{B}_1 R + \left(\frac{\dot{B}_2}{R^2}\right)\right](2-4\nu)\right\} \times \sin\beta\cos\beta, \tag{146}$$

$$\dot{u}_\theta(R,\beta,\theta) = 0, \tag{147}$$

with

$$\dot{A}_1 = 0, \quad \dot{A}_2 = \frac{5Sa^4}{7-5\nu}, \qquad \dot{A}_3 = \frac{3Sa^2(6-5\nu)}{2(7-5\nu)},$$
$$\dot{B}_1 = 0, \quad \dot{B}_2 = -\frac{15Sa^2}{2(7-5\nu)}, \tag{148}$$

and

$$\left(\frac{\dot{A}_2}{R^4}\right) = \frac{\dot{A}_2 R^4 - 4A_2R^3\dot{R}}{R^8}, \tag{149}$$

$$\left(\frac{\dot{A}_3}{R^2}\right) = \frac{\dot{A}_3 R^2 - 2A_3R\dot{R}}{R^4}, \tag{150}$$

$$\left(\frac{\dot{B}_2}{R^2}\right) = \frac{\dot{B}_2 R^2 - 2B_2R\dot{R}}{R^4}. \tag{151}$$

The analytical stress sensitivity is expressed by

$$\dot{\sigma}_{RR}(R,\beta,\theta) = \frac{S}{7-5\nu}\left[\left(\frac{\dot{a^3}}{R^3}\right)(6-5(5-\nu)\cos^2\beta)\right] + \frac{S}{7-5\nu}\left[6\left(\frac{\dot{a^5}}{R^5}\right)(3\cos^2\beta-1)\right], \tag{152}$$

$$\dot{\sigma}_{\theta\theta}(R,\beta,\theta) = \frac{S}{2(7-5\nu)}\left[\left(\frac{\dot{a^3}}{R^3}\right)(5\nu-2+5(1-2\nu)\cos^2\beta)\right] + \frac{S}{2(7-5\nu)}\left[\left(\frac{\dot{a^5}}{R^5}\right)(1-5\cos^2\beta)\right], \tag{153}$$

$$\dot{\sigma}_{\beta\beta}(R,\beta,\theta) = \frac{S}{2(7-5\nu)}\left[\left(\dot{\frac{a^3}{R^3}}\right)(4-5\nu+5(1-2\nu)\cos^2\beta)\right] + \frac{S}{2(7-5\nu)}\left[3\left(\dot{\frac{a^5}{R^5}}\right)(3-7\cos^2\beta)\right], \tag{154}$$

$$\dot{\sigma}_{R\beta}(R,\beta,\theta) = S\left\{\frac{1}{7-5\nu}\left[-5(1+\nu)\left(\dot{\frac{a^3}{R^3}}\right)+12\left(\dot{\frac{a^5}{R^5}}\right)\right]\right\}\sin\beta\cos\beta \tag{155}$$

with

$$\left(\dot{\frac{a^3}{R^3}}\right) = \frac{3a^2R^3-3a^3R^2\dot{R}}{R^6}, \tag{156}$$

$$\left(\dot{\frac{a^5}{R^5}}\right) = \frac{5a^4R^5-5a^5R^4\dot{R}}{R^{10}}. \tag{157}$$

Appendix D. The control point positions of the chair problem

Index	x	y	z	Index	x	y	z
0	-11.8248	-5.5254	18.1044	137	0.0000	-5.5254	15.8521
1	0.7906	7.5000	11.0567	138	4.3897	-5.5254	16.0429
2	0.0000	7.5000	10.9089	139	8.1354	-5.5254	16.5386
3	11.8248	-5.5254	18.1044	140	6.9099	8.5726	26.9186
4	9.7096	1.7408	25.1715	141	11.8531	-0.7124	21.7036
5	0.0000	4.3550	10.9496	142	-3.2353	9.8577	20.5650
6	9.0343	5.7337	27.3118	143	-3.1381	10.6282	26.9340
7	11.8248	-1.0635	18.1044	144	0.0000	-1.0635	15.8521
8	0.7230	5.2923	11.1037	145	4.3897	-1.0635	16.0429
9	1.2093	0.8429	4.1005	146	8.1354	-1.0635	16.5386
10	-9.7096	1.7408	25.1715	147	11.5536	-2.0665	24.2342
11	-9.0343	5.7337	27.3118	148	-11.4759	1.6468	24.5411
12	0.0000	1.4393	3.9528	149	3.2353	9.8577	20.5650
13	12.6838	-4.6143	21.6804	150	-1.2129	9.8197	19.9113
14	0.0000	-2.0000	5.1199	151	-1.0959	10.9445	26.9385
15	8.8457	6.8115	23.2897	152	0.0000	2.1691	2.6246

Continued...

Index	x	y	z	Index	x	y	z
16	1.1059	2.0000	5.2739	153	0.0000	2.8779	1.7493
17	0.0000	-5.5254	14.0394	154	3.1381	10.6282	26.9340
18	4.4307	-5.5254	14.2320	155	11.4759	1.6468	24.5411
19	8.3585	-5.5254	14.7834	156	-10.8708	-0.1457	23.8054
20	6.7088	7.8334	27.0720	157	1.2129	9.8197	19.9113
21	12.6838	-0.7093	21.6804	158	0.0000	12.5468	30.7859
22	2.6954	3.6164	1.8235	159	-4.3897	-9.6560	16.0429
23	-11.8248	-1.0635	18.1044	160	1.0959	10.9445	26.9385
24	0.0000	-1.0635	14.0394	161	-8.1354	-9.1066	16.5386
25	4.4307	-1.0635	14.2320	162	0.0000	9.9873	19.9147
26	8.3585	-1.0635	14.7834	163	-11.0548	-8.1920	18.3343
27	12.2012	-2.1165	25.4895	164	0.0000	11.1088	26.9390
28	-12.6838	-4.6143	21.6804	165	-7.4417	10.9747	30.3230
29	3.1344	8.3358	20.6542	166	10.8708	-0.1457	23.8054
30	0.0000	2.6904	1.6758	167	0.0000	13.0616	33.6076
31	0.0000	-4.9728	1.7165	168	-3.7288	5.9288	16.0429
32	2.5584	-4.1710	1.8706	169	-6.9106	5.7090	16.5386
33	3.6891	3.1133	1.9998	170	-9.1637	7.7478	22.3643
34	3.0405	9.1183	27.0720	171	0.0000	-9.8751	15.8521
35	12.1862	1.5945	25.7384	172	-11.8536	-6.7256	21.7112
36	-8.8457	6.8115	23.2897	173	-5.5891	12.1772	32.2636
37	1.1423	8.2892	19.9854	174	-9.5096	4.5044	18.3343
38	2.8981	-4.9347	1.9441	175	-3.2015	9.4970	19.7039
39	0.0000	3.1403	1.8521	176	4.3897	-9.6560	16.0429
40	1.0484	9.4248	27.0720	177	-3.1521	14.0000	33.3505
41	0.0000	-5.4990	1.8928	178	-1.1009	10.5000	15.9350
42	0.0000	8.1864	19.9854	179	-1.1307	13.6880	33.5333
43	3.3736	-6.0039	2.0469	180	1.1174	13.0608	33.6074
44	0.0000	9.4248	27.0720	181	8.1354	-9.1066	16.5386

Continued...

Index	x	y	z	Index	x	y	z
45	9.1301	9.6359	0.8981	182	1.1162	12.5449	30.7855
46	10.4367	-0.6280	24.2471	183	-11.6124	-4.6372	23.5141
47	-4.4307	-5.5254	14.2320	184	-11.8527	2.0109	21.6890
48	-8.3585	-5.5254	14.7834	185	-11.2311	4.3120	25.4607
49	0.0000	1.6739	0.7504	186	-0.7906	7.5000	11.0567
50	-6.7088	7.8334	27.0720	187	3.1266	12.7512	33.4266
51	0.0000	-9.8751	14.0394	188	3.1463	12.2312	30.6770
52	-12.6838	-0.7093	21.6804	189	-0.7230	5.2923	11.1037
53	1.9532	-3.3553	3.2886	190	-1.2093	0.8429	4.1005
54	-4.4307	-1.0635	14.2320	191	5.6167	11.8676	32.3628
55	0.0000	-9.6735	0.7911	192	6.1394	10.6753	30.0365
56	4.4307	-9.6560	14.2320	193	-1.1059	2.0000	5.2739
57	-8.3585	-1.0635	14.7834	194	-2.6954	3.6164	1.8235
58	-12.2012	-2.1165	25.4895	195	7.4667	10.5979	30.4504
59	-3.1344	8.3358	20.6542	196	8.2112	9.0518	28.5842
60	-3.0405	9.1183	27.0720	197	-2.5584	-4.1710	1.8706
61	8.3585	-9.1066	14.7834	198	11.0548	-8.1920	18.3343
62	9.4216	-4.7046	0.9452	199	7.4417	10.9747	30.3230
63	9.1301	9.6359	0.1140	200	-3.6891	3.1133	1.9998
64	-12.1862	1.5945	25.7384	201	0.0000	5.9288	15.8521
65	-1.1423	8.2892	19.9854	202	3.7288	5.9288	16.0429
66	-1.0484	9.4248	27.0720	203	11.5387	4.2867	25.7199
67	0.0000	1.6739	0.1257	204	11.3331	3.2015	25.0775
68	0.0000	-4.1572	3.1346	205	-3.3736	-6.0039	2.0469
69	0.0000	-5.1920	1.7900	206	-9.1301	9.6359	0.8981
70	0.0000	-9.6735	0.1225	207	-1.1162	12.5449	30.7855
71	9.4216	-4.7046	0.1102	208	6.9106	5.7090	16.5386
72	-10.4367	-0.6280	24.2471	209	9.1637	7.7478	22.3643
73	9.1301	9.6359	0.1450	210	11.8536	-6.7256	21.7112

Continued...

Index	x	y	z	Index	x	y	z
74	3.1094	3.4067	1.8970	211	5.5891	12.1772	32.2636
75	-4.4307	-9.6560	14.2320	212	9.5096	4.5044	18.3343
76	-8.3585	-9.1066	14.7834	213	3.2015	9.4970	19.7039
77	-11.8248	-8.1920	18.1044	214	3.1521	13.3762	33.3505
78	11.8248	-8.1920	18.1044	215	1.1009	10.5000	15.9350
79	7.5017	10.0703	30.6287	216	1.1307	13.6880	33.5333
80	-7.5017	10.0703	30.6287	217	0.0000	10.5000	15.7873
81	0.0000	5.9288	14.0394	218	0.0000	13.6894	33.5337
82	3.7636	5.9288	14.2320	219	-1.1174	13.0608	33.6074
83	0.0000	0.8141	0.1568	220	11.6124	-4.6372	23.5141
84	2.0762	2.4608	2.7723	221	11.8527	2.0109	21.6890
85	0.0000	-8.3391	0.1535	222	-9.4216	-4.7046	0.9452
86	-3.7636	5.9288	14.2320	223	-9.1301	9.6359	0.1140
87	9.4216	-4.7046	0.1413	224	-3.1463	12.2312	30.6770
88	7.1001	5.7090	14.7834	225	11.2311	4.3120	25.4607
89	8.8172	7.0953	22.4229	226	-3.1266	12.7512	33.4266
90	12.6838	-6.7241	21.6804	227	-9.4216	-4.7046	0.1102
91	5.6552	11.4341	32.5015	228	-9.1301	9.6359	0.1450
92	10.6762	4.5044	18.1044	229	-6.1394	10.6753	30.0365
93	3.1059	7.9727	19.7874	230	-5.6167	11.8676	32.3628
94	3.0909	11.8762	33.5332	231	-9.4216	-4.7046	0.1413
95	1.0067	7.9262	15.9821	232	-8.2112	9.0518	28.5842
96	1.0988	12.1827	33.7110	233	-7.4667	10.5979	30.4504
97	0.0000	7.5988	15.8280	234	-11.3331	3.2015	25.0775
98	0.0000	12.1827	33.7110	235	-11.5387	4.2867	25.7199
99	-7.1001	5.7090	14.7834	236	-11.8529	0.8762	21.6951
100	12.2012	-4.7106	24.2648	237	-12.1990	2.0133	21.6854
101	12.6838	2.0165	21.6804	238	-11.0548	2.1845	18.3343
102	-8.8172	7.0953	22.4229	239	-9.7856	4.5044	18.2385

Continued...

Index	x	y	z	Index	x	y	z
103	-12.6838	-6.7241	21.6804	240	-8.1354	2.8871	16.5386
104	-5.6552	11.4341	32.5015	241	-6.9895	5.7090	15.8073
105	11.9693	4.2514	26.0829	242	-4.3897	3.0153	16.0429
106	-10.6762	4.5044	18.1044	243	-3.7433	5.9288	15.2884
107	-3.1059	7.9727	19.7874	244	0.0000	3.0153	15.8521
108	-3.0909	11.8762	33.5332	245	0.0000	5.9288	15.0968
109	-1.0067	7.9262	15.9821	246	3.7288	3.0153	16.0429
110	-1.0988	12.1827	33.7110	247	3.7433	5.9288	15.2884
111	-1.9532	-3.3553	3.2886	248	6.9106	2.8871	16.5386
112	-2.8981	-4.9347	1.9441	249	6.9895	5.7090	15.8073
113	-12.2012	-4.7106	24.2648	250	11.0548	2.1845	18.3343
114	-12.6838	2.0165	21.6804	251	9.7856	4.5044	18.2385
115	-11.9693	4.2514	26.0829	252	11.8529	0.8762	21.6951
116	-11.0548	-5.5254	18.3343	253	12.1990	2.0133	21.6854
117	-10.1089	2.2567	24.1159	254	-0.4566	10.9461	26.9388
118	-9.2885	6.3599	26.1498	255	0.4566	10.9461	26.9388
119	-11.0548	-1.0635	18.3343	256	0.5054	9.8233	19.9133
120	-11.8534	-4.6151	21.7077	257	-0.5054	9.8233	19.9133
121	-9.1568	7.4997	23.2265	258	0.4587	10.5000	15.8488
122	-2.0762	2.4608	2.7723	259	-0.4587	10.5000	15.8488
123	11.0548	-5.5254	18.3343	260	0.3294	7.5000	10.9705
124	10.1089	2.2567	24.1159	261	-0.3294	7.5000	10.9705
125	-4.3897	-5.5254	16.0429	262	-0.5039	1.1908	4.0143
126	9.2885	6.3599	26.1498	263	0.5039	1.1908	4.0143
127	11.0548	-1.0635	18.3343	264	-0.4195	7.7952	15.8922
128	-8.1354	-5.5254	16.5386	265	0.4195	7.7952	15.8922
129	-6.9099	8.5726	26.9186	266	0.4759	8.2892	19.9854
130	-11.8531	-0.7124	21.7036	267	-0.4759	8.2892	19.9854
131	-3.1094	3.4067	1.8970	268	-0.3012	4.8056	11.0138

Continued...

Index	x	y	z	Index	x	y	z
132	-4.3897	-1.0635	16.0429	269	0.3012	4.8056	11.0138
133	11.8534	-4.6151	21.7077	270	-0.4368	9.4248	27.0720
134	-8.1354	-1.0635	16.5386	271	0.4608	2.0000	5.1840
135	9.1568	7.4997	23.2265	272	0.4368	9.4248	27.0720
136	-11.5536	-2.0665	24.2342	273	-0.4608	-2.0000	5.1840

Table 6: The control points of the initial chair geometry (all of the weights $w_A = 1$)

Index	x	y	z	Index	x	y	z
1	-11.8248	-5.5254	18.1044	138	0.0000	-5.5254	15.8521
2	0.7906	9.2200	11.0567	139	4.3897	-5.5254	16.0429
3	0.0000	9.2200	10.9089	140	8.1354	-5.5254	16.5386
4	11.8248	-5.5254	18.1044	141	6.9099	8.5726	26.9186
5	9.7096	1.7408	25.1715	142	11.8531	-0.7124	21.7036
6	0.0000	4.3550	10.9496	143	-3.2353	9.8577	20.5650
7	9.0343	5.7337	27.3118	144	-3.1381	10.6282	26.9340
8	11.8248	-1.0635	18.1044	145	0.0000	-1.0635	15.8521
9	0.7230	5.2923	11.1037	146	4.3897	-1.0635	16.0429
10	1.2093	0.8429	4.1005	147	8.1354	-1.0635	16.5386
11	-9.7096	1.7408	25.1715	148	11.5536	-2.0665	24.2342
12	-9.0343	5.7337	27.3118	149	-11.4759	1.6468	24.5411
13	0.0000	1.4393	3.9528	150	3.2353	9.8577	20.5650
14	12.6838	-4.6143	21.6804	151	-1.2129	9.8197	19.9113
15	0.0000	1.0000	5.1199	152	-1.0959	10.9445	26.9385
16	8.8457	6.8115	23.2897	153	0.0000	2.1691	2.6246
17	1.1059	1.0000	5.2739	154	0.0000	2.8779	1.7493
18	0.0000	-5.5254	14.0394	155	3.1381	10.6282	26.9340
19	4.4307	-5.5254	14.2320	156	11.4759	1.6468	24.5411

Continued. . .

Index	x	y	z	Index	x	y	z
20	8.3585	-5.5254	14.7834	157	-10.8708	-0.1457	23.8054
21	6.7088	7.8334	27.0720	158	1.2129	9.8197	19.9113
22	12.6838	-0.7093	21.6804	159	0.0000	12.5468	30.7859
23	2.6954	3.6164	1.8235	160	-4.3897	-9.6560	16.0429
24	-11.8248	-1.0635	18.1044	161	1.0959	10.9445	26.9385
25	0.0000	-1.0635	14.0394	162	-8.1354	-9.1066	16.5386
26	4.4307	-1.0635	14.2320	163	0.0000	9.9873	19.9147
27	8.3585	-1.0635	14.7834	164	-11.0548	-8.1920	18.3343
28	12.2012	-2.1165	25.4895	165	0.0000	11.1088	26.9390
29	-12.6838	-4.6143	21.6804	166	-7.4417	10.9747	30.3230
30	3.1344	8.3358	20.6542	167	10.8708	-0.1457	23.8054
31	0.0000	2.6904	1.6758	168	0.0000	13.0616	33.6076
32	0.0000	-4.9728	1.7165	169	-3.7288	5.9288	16.0429
33	2.5584	-4.1710	1.8706	170	-6.9106	5.7090	16.5386
34	3.6891	3.1133	1.9998	171	-9.1637	7.7478	22.3643
35	3.0405	9.1183	27.0720	172	0.0000	-9.8751	15.8521
36	12.1862	1.5945	25.7384	173	-11.8536	-6.7256	21.7112
37	-8.8457	6.8115	23.2897	174	-5.5891	12.1772	32.2636
38	1.1423	8.2892	19.9854	175	-9.5096	4.5044	18.3343
39	2.8981	-4.9347	1.9441	176	-3.2015	9.4970	19.7039
40	0.0000	3.1403	1.8521	177	4.3897	-9.6560	16.0429
41	1.0484	9.4248	27.0720	178	-3.1521	13.3762	33.3505
42	0.0000	-5.4990	1.8928	179	-1.1009	12.8900	15.9350
43	0.0000	8.1864	19.9854	180	-1.1307	13.6880	33.5333
44	3.3736	-6.0039	2.0469	181	1.1174	13.0608	33.6074
45	0.0000	9.4248	27.0720	182	8.1354	-9.1066	16.5386
46	9.1301	9.6359	0.8981	183	1.1162	12.5449	30.7855
47	10.4367	-0.6280	24.2471	184	-11.6124	-4.6372	23.5141
48	-4.4307	-5.5254	14.2320	185	-11.8527	2.0109	21.6890

Continued...

Index	x	y	z	Index	x	y	z
49	-8.3585	-5.5254	14.7834	186	-11.2311	4.3120	25.4607
50	0.0000	1.6739	0.7504	187	-0.7906	9.2200	11.0567
51	-6.7088	7.8334	27.0720	188	3.1266	12.7512	33.4266
52	0.0000	-9.8751	14.0394	189	3.1463	12.2312	30.6770
53	-12.6838	-0.7093	21.6804	190	-0.7230	5.2923	11.1037
54	1.9532	-3.3553	3.2886	191	-1.2093	0.8429	4.1005
55	-4.4307	-1.0635	14.2320	192	5.6167	11.8676	32.3628
56	0.0000	-9.6735	0.7911	193	6.1394	10.6753	30.0365
57	4.4307	-9.6560	14.2320	194	-1.1059	1.0000	5.2739
58	-8.3585	-1.0635	14.7834	195	-2.6954	3.6164	1.8235
59	-12.2012	-2.1165	25.4895	196	7.4667	10.5979	30.4504
60	-3.1344	8.3358	20.6542	197	8.2112	9.0518	28.5842
61	-3.0405	9.1183	27.0720	198	-2.5584	-4.1710	1.8706
62	8.3585	-9.1066	14.7834	199	11.0548	-8.1920	18.3343
63	9.4216	-4.7046	0.9452	200	7.4417	10.9747	30.3230
64	9.1301	9.6359	0.1140	201	-3.6891	3.1133	1.9998
65	-12.1862	1.5945	25.7384	202	0.0000	5.9288	15.8521
66	-1.1423	8.2892	19.9854	203	3.7288	5.9288	16.0429
67	-1.0484	9.4248	27.0720	204	11.5387	4.2867	25.7199
68	0.0000	1.6739	0.1257	205	11.3331	3.2015	25.0775
69	0.0000	-4.1572	3.1346	206	-3.3736	-6.0039	2.0469
70	0.0000	-5.1920	1.7900	207	-9.1301	9.6359	0.8981
71	0.0000	-9.6735	0.1225	208	-1.1162	12.5449	30.7855
72	9.4216	-4.7046	0.1102	209	6.9106	5.7090	16.5386
73	-10.4367	-0.6280	24.2471	210	9.1637	7.7478	22.3643
74	9.1301	9.6359	0.1450	211	11.8536	-6.7256	21.7112
75	3.1094	3.4067	1.8970	212	5.5891	12.1772	32.2636
76	-4.4307	-9.6560	14.2320	213	9.5096	4.5044	18.3343
77	-8.3585	-9.1066	14.7834	214	3.2015	9.4970	19.7039

Continued...

Index	x	y	z	Index	x	y	z
78	-11.8248	-8.1920	18.1044	215	3.1521	13.3762	33.3505
79	11.8248	-8.1920	18.1044	216	1.1009	12.8900	15.9350
80	7.5017	10.0703	30.6287	217	1.1307	13.6880	33.5333
81	-7.5017	10.0703	30.6287	218	0.0000	12.8900	15.7873
82	0.0000	5.9288	14.0394	219	0.0000	13.6894	33.5337
83	3.7636	5.9288	14.2320	220	-1.1174	13.0608	33.6074
84	0.0000	0.8141	0.1568	221	11.6124	-4.6372	23.5141
85	2.0762	2.4608	2.7723	222	11.8527	2.0109	21.6890
86	0.0000	-8.3391	0.1535	223	-9.4216	-4.7046	0.9452
87	-3.7636	5.9288	14.2320	224	-9.1301	9.6359	0.1140
88	9.4216	-4.7046	0.1413	225	-3.1463	12.2312	30.6770
89	7.1001	5.7090	14.7834	226	11.2311	4.3120	25.4607
90	8.8172	7.0953	22.4229	227	-3.1266	12.7512	33.4266
91	12.6838	-6.7241	21.6804	228	-9.4216	-4.7046	0.1102
92	5.6552	11.4341	32.5015	229	-9.1301	9.6359	0.1450
93	10.6762	4.5044	18.1044	230	-6.1394	10.6753	30.0365
94	3.1059	7.9727	19.7874	231	-5.6167	11.8676	32.3628
95	3.0909	11.8762	33.5332	232	-9.4216	-4.7046	0.1413
96	1.0067	7.9262	15.9821	233	-8.2112	9.0518	28.5842
97	1.0988	12.1827	33.7110	234	-7.4667	10.5979	30.4504
98	0.0000	7.5988	15.8280	235	-11.3331	3.2015	25.0775
99	0.0000	12.1827	33.7110	236	-11.5387	4.2867	25.7199
100	-7.1001	5.7090	14.7834	237	-11.8529	0.8762	21.6951
101	12.2012	-4.7106	24.2648	238	-12.1990	2.0133	21.6854
102	12.6838	2.0165	21.6804	239	-11.0548	2.1845	18.3343
103	-8.8172	7.0953	22.4229	240	-9.7856	4.5044	18.2385
104	-12.6838	-6.7241	21.6804	241	-8.1354	2.8871	16.5386
105	-5.6552	11.4341	32.5015	242	-6.9895	5.7090	15.8073
106	11.9693	4.2514	26.0829	243	-4.3897	3.0153	16.0429

Continued...

Index	x	y	z	Index	x	y	z
107	-10.6762	4.5044	18.1044	244	-3.7433	5.9288	15.2884
108	-3.1059	7.9727	19.7874	245	0.0000	3.0153	15.8521
109	-3.0909	11.8762	33.5332	246	0.0000	5.9288	15.0968
110	-1.0067	7.9262	15.9821	247	3.7288	3.0153	16.0429
111	-1.0988	12.1827	33.7110	248	3.7433	5.9288	15.2884
112	-1.9532	-3.3553	3.2886	249	6.9106	2.8871	16.5386
113	-2.8981	-4.9347	1.9441	250	6.9895	5.7090	15.8073
114	-12.2012	-4.7106	24.2648	251	11.0548	2.1845	18.3343
115	-12.6838	2.0165	21.6804	252	9.7856	4.5044	18.2385
116	-11.9693	4.2514	26.0829	253	11.8529	0.8762	21.6951
117	-11.0548	-5.5254	18.3343	254	12.1990	2.0133	21.6854
118	-10.1089	2.2567	24.1159	255	-0.4566	10.9461	26.9388
119	-9.2885	6.3599	26.1498	256	0.4566	10.9461	26.9388
120	-11.0548	-1.0635	18.3343	257	0.5054	9.8233	19.9133
121	-11.8534	-4.6151	21.7077	258	-0.5054	9.8233	19.9133
122	-9.1568	7.4997	23.2265	259	0.4587	12.8900	15.8488
123	-2.0762	2.4608	2.7723	260	-0.4587	12.8900	15.8488
124	11.0548	-5.5254	18.3343	261	0.3294	9.2200	10.9705
125	10.1089	2.2567	24.1159	262	-0.3294	9.2200	10.9705
126	-4.3897	-5.5254	16.0429	263	-0.5039	1.1908	4.0143
127	9.2885	6.3599	26.1498	264	0.5039	1.1908	4.0143
128	11.0548	-1.0635	18.3343	265	-0.4195	7.7952	15.8922
129	-8.1354	-5.5254	16.5386	266	0.4195	7.7952	15.8922
130	-6.9099	8.5726	26.9186	267	0.4759	8.2892	19.9854
131	-11.8531	-0.7124	21.7036	268	-0.4759	8.2892	19.9854
132	-3.1094	3.4067	1.8970	269	-0.3012	4.8056	11.0138
133	-4.3897	-1.0635	16.0429	270	0.3012	4.8056	11.0138
134	11.8534	-4.6151	21.7077	271	-0.4368	9.4248	27.0720
135	-8.1354	-1.0635	16.5386	272	0.4608	1.0000	5.1840

Continued...

Index	x	y	z	Index	x	y	z
136	9.1568	7.4997	23.2265	273	0.4368	9.4248	27.0720
137	-11.5536	-2.0665	24.2342	274	-0.4608	1.0000	5.1840

Table 7: The control points of the optimized chair geometry