

ICES REPORT 17-30

November 2017

Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the STEP exchange format

by

Yujie Guo, Jason Heller, Thomas J.R. Hughes, Martin Ruess, and Dominik Schillinger



The Institute for Computational Engineering and Sciences
The University of Texas at Austin
Austin, Texas 78712

Reference: Yujie Guo, Jason Heller, Thomas J.R. Hughes, Martin Ruess, and Dominik Schillinger, "Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the STEP exchange format," ICES REPORT 17-30, The Institute for Computational Engineering and Sciences, The University of Texas at Austin, November 2017.

Variationally consistent isogeometric analysis of trimmed thin shells at finite deformations, based on the STEP exchange format

Yujie Guo^{a,*}, Jason Heller^b, Thomas J.R. Hughes^c, Martin Ruess^d, Dominik Schillinger^e

^a*Interdisciplinary Research Institute of Aeronautics and Astronautics, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, 210016, Nanjing, P.R. China.*

^b*Faircad3d, LLC, Santa Fe, NM, USA*

^c*Institute for Computational Engineering and Sciences, The University of Texas at Austin, USA*

^d*School of Engineering, University of Glasgow, UK*

^e*Department of Civil, Environmental, and Geo-Engineering, University of Minnesota, Twin Cities, MN, USA*

Abstract

Following a series of recent innovations, isogeometric shell analysis based on trimmed CAD surfaces is currently being developed into an accurate, efficient and mature design-through-analysis methodology. This work contributes to this emerging technology with respect to the following aspects. On the analysis side, we present a robust variationally consistent Nitsche-type formulation for thin shells at large deformations that weakly enforces coupling constraints at trimming curves. On the geometry side, we present a set of algorithms that enable automatic interaction of trimmed shell analysis with CAD data structures based on the STEP exchange format. We integrate these methodologies in a comprehensive framework for isogeometric trimmed shell analysis. We demonstrate that our framework is able to seamlessly perform large-deformation stress analysis of an industry-scale 76-patch surface model of a Dodge RAM hood, while delivering comparable accuracy with respect to Simulia's commercial software package Abaqus.

Keywords: Isogeometric analysis, trimmed shell surfaces, Kirchhoff-Love shells, weakly enforced interface constraints, STEP exchange format, Abaqus

*Corresponding author;

Yujie Guo, College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, #29 Yu Dao Street, 210016, Nanjing, P. R. China; E-mail: yujieguo@nuaa.edu.cn

Email addresses: yujieguo@nuaa.edu.cn (Yujie Guo), jason@faircad3d.com (Jason Heller), hughes@ices.utexas.edu (Thomas J.R. Hughes), martin.ruess@glasgow.ac.uk (Martin Ruess), dominik@umn.edu (Dominik Schillinger)

Preprint submitted to Computer Methods in Applied Mechanics and Engineering

October 28, 2017

Contents

1	Introduction	3
2	Isogeometric Kirchhoff-Love shells on trimmed spline surfaces	5
2.1	Thin shell kinematics and constitutive relations	5
2.2	Variational formulation, isogeometric discretization and linearization	7
2.3	Geometrically accurate representation of trimmed elements	8
2.3.1	The blending function method	9
2.3.2	The finite cell method	10
3	A variationally consistent geometrically nonlinear coupling formulation	11
3.1	Multipatch trimming as a mechanics-based interface problem	11
3.2	Relevant identities from shell kinematics and differential geometry	12
3.3	Variational formulation of Nitsche-type coupling terms	13
3.4	Isogeometric discretization and linearization	14
3.5	Element-wise estimation of stabilization parameters	16
4	Extracting trimming and coupling information from ISO 10303 STEP	17
4.1	The b-rep structure of STEP	19
4.2	Quadrature of trimmed elements based on STEP	20
4.2.1	Robust point location query	20
4.2.2	Determination of element status	22
4.2.3	Generating element-wise trimming curve segments	22
4.3	Quadrature along trimming curves based on STEP	22
5	Numerical examples	25
5.1	Slit annular plate	25
5.2	Hemispherical shell subjected to alternating radial forces	28
5.3	Hinged cylindrical roof subjected to a pinching force	29
5.4	Large deformation trimmed shell analysis of a Dodge RAM hood	31
6	Summary and conclusions	36
Appendix A	Some relevant details from differential geometry and shell kinematics	38
Appendix A.1	Derivatives of the surface's contravariant metric coefficients	38
Appendix A.2	Derivatives of the normal vector	38
Appendix A.3	Derivatives of the stress resultants and bending moments	40
Appendix B	STEP source code of the trimmed surface example	41

1. Introduction

Over the last decade, there has been increasing momentum to re-think the process of how geometric models generated by computer-aided design (CAD) are transferred into finite element discretizations. A fundamental pillar in this effort is isogeometric analysis, initiated by the pioneering work of HUGHES and collaborators [1, 2]. Its core idea is to use the same smooth and higher-order basis functions, e.g. non-uniform rational B-splines (NURBS), for the representation of both the geometry in CAD and the approximation of the solution fields in finite element analysis (FEA). The design of thin structural components in the automotive and aerospace industries is a salient example that illustrates the demand for a better integration of CAD and FEA. In current industrial practice, CAD procedures largely rely on free-form surfaces and the trimming paradigm, whereas FEA uses nonlinear shell finite element technology, based on nodal basis functions and associated meshing tools.

Trimming is currently one of the most fundamental procedures in CAD that enables the construction of complex geometries. At the same time, it also constitutes a serious roadblock to interoperability between CAD systems and FEA software. Geometric objects based on the trimming paradigm consist of a number of individual surfaces that are constructed independently from each other. In each surface, only certain regions are part of the actual object, defined by trimming curves. This guarantees flexibility of CAD procedures, since overlapping surface areas can be simply cut away. Trimming only affects the visualization of the surface, while its parameterization remains unchanged. Trimming operations, however, cannot practically be performed exactly, so that trimming curves are only approximations of exact intersection curves, depending on a given tolerance. This leads to small gaps and overlaps between the space curves of two neighboring trimmed surfaces, which are the main source of the lack of interoperability. They necessitate geometry repair before the actual mesh generation, and mesh healing to make up for any inaccuracies that could not be detected beforehand. A comprehensive summary on trimming and its impact on interoperability is given in the recent review article by MARUSSIG and HUGHES [3].

Following a series of recent innovations, isogeometric analysis for trimmed thin shell surfaces is currently developing into an accurate, efficient and mature design-through-analysis methodology. At the core stands the pioneering work by BREITENBERGER who recently presented the first fully operable isogeometric framework for trimmed shell analysis in his PhD thesis [4]. Based on a series of recent publications on the topic [3, 5–7], we identify four key components for establishing a successful isogeometric methodology for trimmed shell analysis:

1. an efficient and accurate isogeometric shell technology,
2. methods to enforce boundary and coupling conditions at non-matching trimming curves,
3. quadrature methods for the integration of stiffness and residual forms in trimmed elements and along surface-to-surface intersection curves,
4. and the ability to query accurate geometric information related to trimmed surfaces and surface-to-surface intersection curves from CAD data structures.

The first three components have profited from significant progress in both isogeometric and embedded domain finite element methods in recent years. The interested reader is referred to the following publications on isogeometric shell formulations [8–19], on variationally imposing

boundary and coupling constraints [20–26], and on geometrically accurate quadrature methods for trimmed elements [27–33] as well as the references cited therein. The fourth component, the interaction of trimmed shell analysis with CAD data structures, is essential for demonstrating the competitiveness for complex real-world trimmed objects. In his thesis, BREITENBERGER devised a set of algorithms that transfer trimming information available in CAD systems into analysis-suitable isogeometric shell discretizations [4].

The current paper contributes to the emerging technology in the following way. First, we generalize our prior work on variationally consistent Nitsche-type methods for the weak imposition of Dirichlet boundary and coupling constraints to geometrically nonlinear Kirchhoff-Love shells [6, 7, 34]. In the large-deformation setting, penalty-type methods have dominated the imposition of coupling constraints along surface-to-surface trimming curves so far [5, 35, 36]. It is well-known, however, that variationally consistent Nitsche-type methods have a number of advantages over penalty methods, for example optimal rates of convergence in the asymptotic limit, independence from ad hoc rules to choose suitable penalty parameters, and a better conditioning of the discrete system [6, 21, 37–39]. We show that the extension of Nitsche-type methods to the large-deformation setting is a nontrivial task, which requires a number of additional considerations on shell kinematics and differential geometry. We also generalize the element-wise estimation of stabilization parameters to the geometrically nonlinear shell setting.

Second, we implement the new variationally consistent method to impose coupling constraints into a fully operable framework for isogeometric trimmed shell analysis that satisfies all four requirements stated above. In particular, our framework is able to automatically interact with CAD data structures based on the STEP exchange format. STEP (standard for the exchange of product model data) is a widely used standard (ISO 10303) for the platform-independent exchange of CAD geometries. Since all modern CAD systems are able to reliably export trimmed CAD surfaces in the STEP format, our framework can operate immediately with any CAD system. For quadrature of trimmed shell elements, we extend the two-dimensional blending function method devised by KUDELA et al. [40] to curved trimmed shell elements. For readers interested in implementation details, we provide a concise overview of all relevant algorithms for STEP integration, focusing on the classification of elements as full, cut and void, and the generation of quadrature points for trimmed elements and along surface-to-surface intersection curves.

Our article is organized as follows: In Section 2, we review an isogeometric Kirchhoff-Love shell formulation on trimmed spline surfaces, including the blending function method for integrating trimmed shell elements. In Section 3, we present the derivation of a variationally consistent Nitsche-type method for imposing coupling constraints along trimming curves in the large-deformation setting. In Section 4, we summarize how trimmed shell analysis can interact with STEP. In Section 5, we present large-deformation numerical experiments, including stress analysis results for an industry-scale car hood, that demonstrate the accuracy and the seamless design-through-analysis workflow of isogeometric trimmed shell analysis. Section 6 puts the numerical results into perspective and motivates future work.

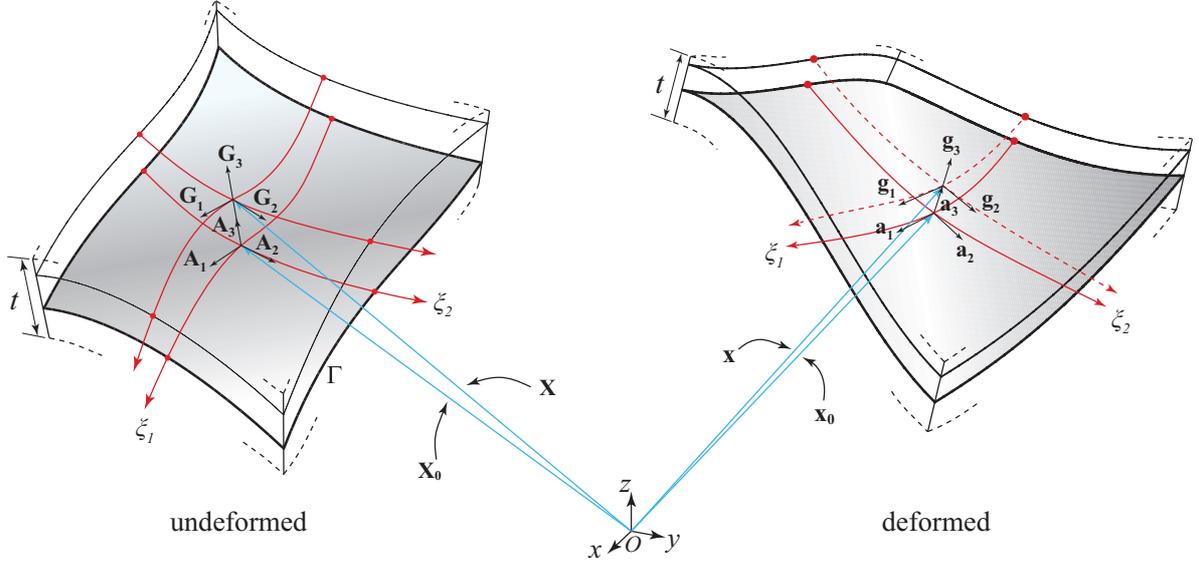


Figure 1: Shell geometry description in undeformed and deformed configurations.

2. Isogeometric Kirchhoff-Love shells on trimmed spline surfaces

We first review a compact rotation-free Kirchhoff-Love shell formulation, based on the work of KIENDL, BLETZINGER, LINHARD and WÜCHNER [8], and its C^1 -continuous isogeometric discretization and consistent linearization. We also describe an element-wise blending function method for the geometrically accurate parametrization of trimmed shell elements and outline its advantages with respect to the finite cell method used in our prior work [7]. We note that we use upper case notations for quantities which refer to the undeformed reference configuration, and lower case notations for quantities which refer to the current deformed configuration. Without specific notation, Greek indices take values $\{1, 2\}$ and Latin indices take values $\{1, 2, 3\}$.

2.1. Thin shell kinematics and constitutive relations

As illustrated in Fig. 1, the position \mathbf{x} of a material point within the shell body is described in the current configuration by

$$\mathbf{x}(\xi_1, \xi_2, \xi_3) = \mathbf{x}_0(\xi_1, \xi_2) + \xi_3 t \mathbf{a}_3(\xi_1, \xi_2). \quad (1)$$

In this equation, \mathbf{x}_0 is the location vector of the shell mid-surface, ξ_i are the curvilinear coordinates, where $\xi_3 \in [-0.5, 0.5]$, t is the shell thickness, and \mathbf{a}_3 is the normal director of the mid-surface.

Based on the Kirchhoff-Love assumptions [41, 42], the 3D Green-Lagrange strain tensor \mathbf{E} reduces to the in-plane strain components

$$\mathbf{E} = E_{\alpha\beta} \mathbf{G}^\alpha \otimes \mathbf{G}^\beta. \quad (2)$$

The covariant components $E_{\alpha\beta}$ are represented as

$$E_{\alpha\beta} = \frac{1}{2} (g_{\alpha\beta} - G_{\alpha\beta}). \quad (3)$$

Detailed descriptions of the covariant and contravariant bases can be found in [43]. The strain tensor components (3) are further split into in-plane and out-of-plane contributions

$$E_{\alpha\beta} = \varepsilon_{\alpha\beta} + \xi_3 t \kappa_{\alpha\beta}, \quad (4)$$

with $\varepsilon_{\alpha\beta}$ and $(\xi_3 t \kappa_{\alpha\beta})$ independently representing membrane and bending effects. Membrane and bending strains are defined as

$$\varepsilon_{\alpha\beta} = \frac{1}{2}(a_{\alpha\beta} - A_{\alpha\beta}), \quad (5)$$

$$a_{\alpha\beta} = \mathbf{a}_\alpha \cdot \mathbf{a}_\beta, \quad (6)$$

$$A_{\alpha\beta} = \mathbf{A}_\alpha \cdot \mathbf{A}_\beta, \quad (7)$$

and

$$\kappa_{\alpha\beta} = B_{\alpha\beta} - b_{\alpha\beta}, \quad (8)$$

$$b_{\alpha\beta} = \mathbf{a}_{\alpha,\beta} \cdot \mathbf{a}_3, \quad (9)$$

$$B_{\alpha\beta} = \mathbf{A}_{\alpha,\beta} \cdot \mathbf{A}_3, \quad (10)$$

where $\kappa_{\alpha\beta}$ represents the curvature of the shell mid-surface, $\mathbf{a}_\alpha = \mathbf{x}_{0,\alpha}$, and $\mathbf{a}_{\alpha,\beta} = \mathbf{x}_{0,\alpha\beta}$.

The strain relations $E_{\alpha\beta}$ are defined in the contravariant basis and a transformation to the local Cartesian coordinate system \mathbf{E}_γ follows as

$$\bar{E}_{\gamma\delta} = E_{\alpha\beta}(\mathbf{E}_\gamma \cdot \mathbf{G}^\alpha)(\mathbf{G}^\beta \cdot \mathbf{E}_\delta), \quad (11)$$

containing only in-plane strain components. Detailed descriptions of the covariant and contravariant bases involved can be found for example in [43].

In our work, we assume the St. Venant-Kirchhoff constitutive model. The relation between the second Piola-Kirchhoff stress tensor \mathbf{S} and the Green-Lagrange strains \mathbf{E} is established with the constitutive equations in Voigt notation

$$\begin{bmatrix} \bar{S}^{11} \\ \bar{S}^{22} \\ \bar{S}^{12} \end{bmatrix} = \hat{\mathbf{C}} \begin{bmatrix} \bar{E}_{11} \\ \bar{E}_{22} \\ 2 \bar{E}_{12} \end{bmatrix}, \quad (12)$$

where $\bar{S}^{\alpha\beta}$ denotes the stress tensor coefficients and $\hat{\mathbf{C}}$ is the reduced material matrix for plane stress problems [42]. Integration of the stress components over the shell thickness provides the force and moment stress resultants $\bar{\mathbf{N}}$ and $\bar{\mathbf{M}}$, written in Voigt notation as

$$\begin{bmatrix} \bar{N}^{11} \\ \bar{N}^{22} \\ \bar{N}^{12} \end{bmatrix} = t \cdot \hat{\mathbf{C}} \begin{bmatrix} \bar{\varepsilon}_{11} \\ \bar{\varepsilon}_{22} \\ 2 \bar{\varepsilon}_{12} \end{bmatrix}, \quad (13)$$

and

$$\begin{bmatrix} \bar{M}^{11} \\ \bar{M}^{22} \\ \bar{M}^{12} \end{bmatrix} = \frac{t^3}{12} \cdot \hat{\mathbf{C}} \begin{bmatrix} \bar{\kappa}_{11} \\ \bar{\kappa}_{22} \\ 2 \bar{\kappa}_{12} \end{bmatrix}. \quad (14)$$

2.2. Variational formulation, isogeometric discretization and linearization

The shell total potential energy, Π , can be represented as the sum of the elastic strain energy stored in the deformed body and the potential energy associated to the body force, \mathbf{b} , and surface traction, \mathbf{t}_0 , as follows

$$\Pi(\mathbf{u}, \lambda) = \frac{1}{2} \int_{\Omega} \mathbf{S} : \mathbf{E} d\Omega - \lambda \left[\int_{\Omega} \mathbf{b} \cdot \mathbf{u} d\Omega + \int_{\Gamma_t} \mathbf{t}_0 \cdot \mathbf{u} d\Gamma \right] \quad (15)$$

where Ω denotes the shell body, Γ_t the Neumann boundary and loads are scaled by a load factor λ . The stationary condition of the potential energy gives the equilibrium equation:

$$\begin{aligned} \delta\Pi(\mathbf{u}, \lambda) &= \int_{\Omega} \mathbf{S} : \delta\mathbf{E} d\Omega - \lambda \left[\int_{\Omega} \mathbf{b} \cdot \delta\mathbf{u} d\Omega + \int_{\Gamma_t} \mathbf{t}_0 \cdot \delta\mathbf{u} d\Gamma \right] \\ &= \mathcal{W}_I - \mathcal{W}_E = 0 \end{aligned} \quad (16)$$

where $\delta\mathbf{E}$ and $\delta\mathbf{u}$ denote the variation of the strain and displacement, respectively. We associate the first integral with the work done by internal forces, \mathcal{W}_I , and the second integral with the work done by external forces, \mathcal{W}_E . We also assume that the external loads \mathbf{b} and \mathbf{t}_0 are independent of the displacement \mathbf{u} .

Following the concept of isogeometric analysis, the displacement field and its variation are discretized with the same spline basis functions R_i as the CAD surface, such that

$$\mathbf{u} = \sum_{i=1}^n R_i \mathbf{U}_i \quad (17)$$

$$\delta\mathbf{u} = \sum_{i=1}^n R_i \delta\mathbf{U}_i. \quad (18)$$

The vectors \mathbf{U}_i and $\delta\mathbf{U}_i$ represent sets of three unknown coefficients associated with each control point \mathbf{B}_i . Inserting (17) and (18) into (16) and leveraging the arbitrary nature of $\delta\mathbf{U}_i$, we can write the following discrete system of equations

$$\mathbf{G}(\mathbf{U}, \lambda) = \mathbf{f}_I(\mathbf{U}) - \lambda \mathbf{f}_E = \mathbf{0} \quad (19)$$

where \mathbf{G} denotes the residual vector, defined as the difference between the internal and external equivalent force vectors \mathbf{f}_I and \mathbf{f}_E . In (19), \mathbf{U} is a vector that collects all displacement unknowns. For Kirchhoff-Love shells, the internal and external equivalent force vectors are defined as:

$$\mathbf{f}_I(\mathbf{U}) = \int_{\Omega} \left(\mathbf{N} : \frac{\partial \boldsymbol{\varepsilon}}{\partial \mathbf{U}} + \mathbf{M} : \frac{\partial \boldsymbol{\kappa}}{\partial \mathbf{U}} \right) d\Omega \quad (20)$$

$$\mathbf{f}_E = \int_{\Omega} \mathbf{p} \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{U}} d\Omega + \int_{\Gamma_t} \mathbf{t}_0 \cdot \frac{\partial \mathbf{u}}{\partial \mathbf{U}} d\Gamma \quad (21)$$

where \mathbf{p} denotes the body force per unit area on the mid-surface.

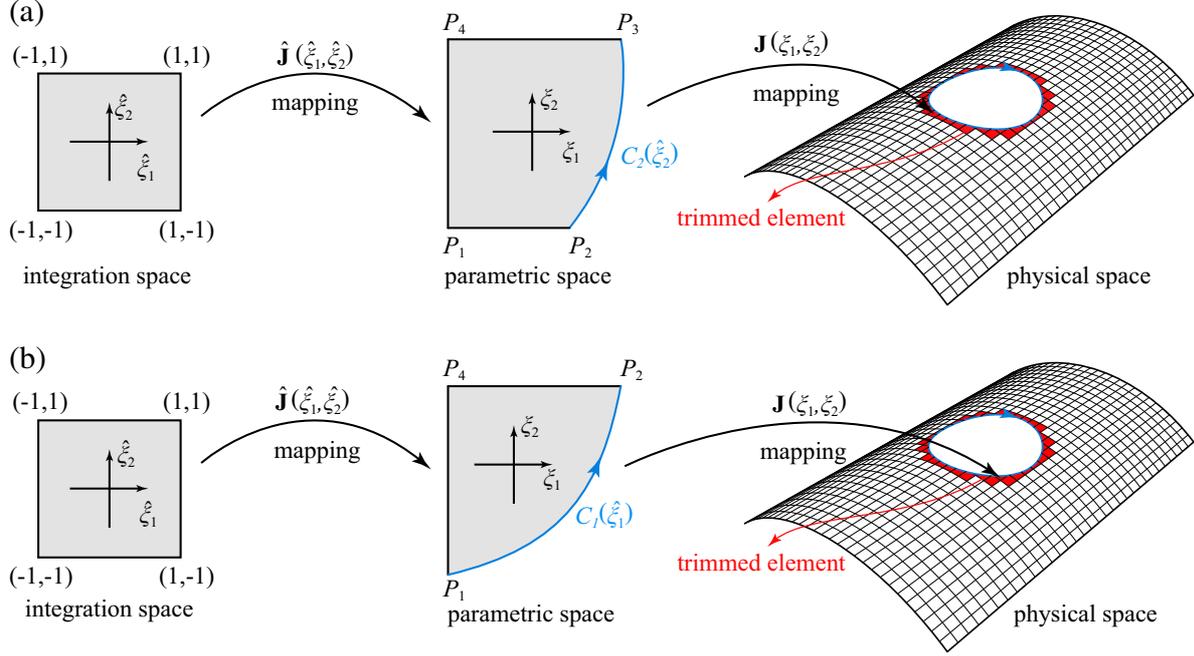


Figure 2: The blending function method for trimmed shell elements: (a) quadrilateral trimmed element, (b) triangular trimmed element.

The linearization of equilibrium equation (19) gives the following Newton-like iteration form:

$$\mathbf{K}_T d\mathbf{U} - d\lambda \mathbf{f}_E = -\mathbf{G} \quad (22)$$

which can be solved iteratively until the residual is reduced to a desired tolerance. Here \mathbf{K}_T is the tangent stiffness matrix which is represented as:

$$\mathbf{K}_T = \int_{\Omega} \left(\frac{\partial \mathbf{N}}{\partial \mathbf{U}} : \frac{\partial \boldsymbol{\varepsilon}}{\partial \mathbf{U}} + \mathbf{N} : \frac{\partial^2 \boldsymbol{\varepsilon}}{\partial \mathbf{U} \partial \mathbf{U}} + \frac{\partial \mathbf{M}}{\partial \mathbf{U}} : \frac{\partial \boldsymbol{\kappa}}{\partial \mathbf{U}} + \mathbf{M} : \frac{\partial^2 \boldsymbol{\kappa}}{\partial \mathbf{U} \partial \mathbf{U}} \right) d\Omega \quad (23)$$

where the stiffness contribution due to external load is neglected, and where $d\Omega$ involves a mapping from the flat reference element in the parametric space of the spline patch to the curved shell element in the physical space.

2.3. Geometrically accurate representation of trimmed elements

The integrals of the Kirchhoff-Love shell formulation (20), (21) and (23) are defined over the physical shell domain and corresponding boundaries and interfaces, which are parametrized in terms of trimmed surfaces and trimming curves. The evaluation of these integrals therefore requires numerical quadrature over trimmed shell elements and along trimming curves, for which we apply the composed strategy outlined below. Its integration and implementation in the context of the STEP exchange format is described later in Section 4.

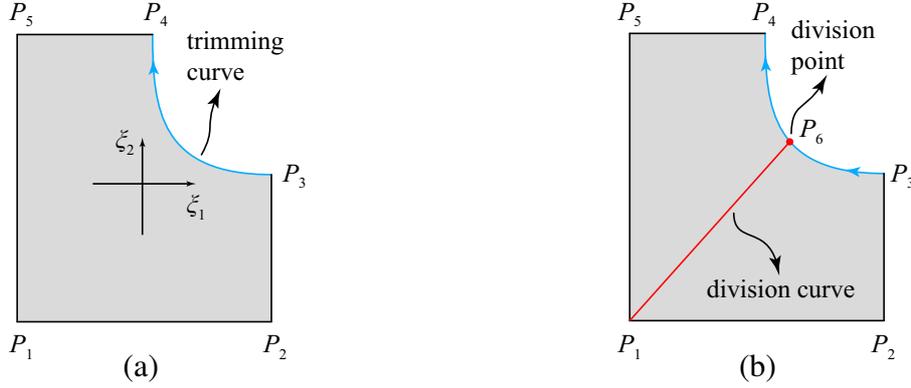


Figure 3: The subdivision concept: the physical part of a trimmed element is decomposed into two quadrilaterals.

2.3.1. The blending function method

If the topology and geometric complexity of the trimmed element allows, we apply a blending function method to accurately represent the geometry of trimmed shell elements. The blending function method was originally introduced in [44] to represent curved boundaries in untrimmed finite elements. The blending concept has been recently extended to trimmed elements, where it constitutes the basis for finding geometrically accurate quadrature points [45, 46]. In the context of trimmed elements, the basic idea of blending is to reconstruct the physical part of the trimmed element with the help of trimming curve segments. All relevant geometric operations can be naturally represented as mappings between different parametric spaces. In the following, we explain the general procedure with a simple example, illustrated in Fig. 2.

We start from the integration domain, parametrized by the integration space $\{\hat{\xi}_1, \hat{\xi}_2\} = [-1, 1]^2$, where quadrature points based on standard Gaussian rules can be easily defined. We can then define a mapping from the integration space to the physical part of the trimmed element in the parametric space $\{\xi_1, \xi_2\}$ of the spline patch. For the case shown in Fig. 2a, the mapping reads

$$\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \frac{1}{4} \left((1 - \hat{\xi}_1)(1 - \hat{\xi}_2) \mathbf{P}_1 + (1 - \hat{\xi}_1)(1 + \hat{\xi}_2) \mathbf{P}_4 \right) + \frac{1}{2} \mathbf{C}_2(\hat{\xi}_2) (1 + \hat{\xi}_1) \quad (24)$$

$\mathbf{P}_i, i = \{1, 2, 3, 4\}$ are the corner points of the trimmed element in the parametric space and $\mathbf{C}_2(\hat{\xi}_2)$ is the trimming curve segment parametrized with respect to $\hat{\xi}_2 = [-1, 1]$. Similar mappings can be easily constructed with respect to the three other boundary curves of the element or with respect to several boundary curves [46]. The Jacobian matrix of the mapping from the integration space to the parametric space simply follows from its standard definition as

$$\hat{\mathbf{J}}(\hat{\xi}_1, \hat{\xi}_2) = d\boldsymbol{\xi}/d\hat{\boldsymbol{\xi}} = \begin{bmatrix} \partial\xi_1(\hat{\xi}_1, \hat{\xi}_2)/\partial\hat{\xi}_1 & \partial\xi_2(\hat{\xi}_1, \hat{\xi}_2)/\partial\hat{\xi}_1 \\ \partial\xi_1(\hat{\xi}_1, \hat{\xi}_2)/\partial\hat{\xi}_2 & \partial\xi_2(\hat{\xi}_1, \hat{\xi}_2)/\partial\hat{\xi}_2 \end{bmatrix}. \quad (25)$$

The mapping from the 2D parametric space $\{\xi_1, \xi_2\}$ to the 3D physical space $\{x, y, z\}$ can then be accomplished in the standard way as in any uncut regular shell element.

A trimmed element can be cut in many ways such that its physical part cannot be represented by a quadrilateral. If the physical part of the trimmed element forms a triangle, the mapping can

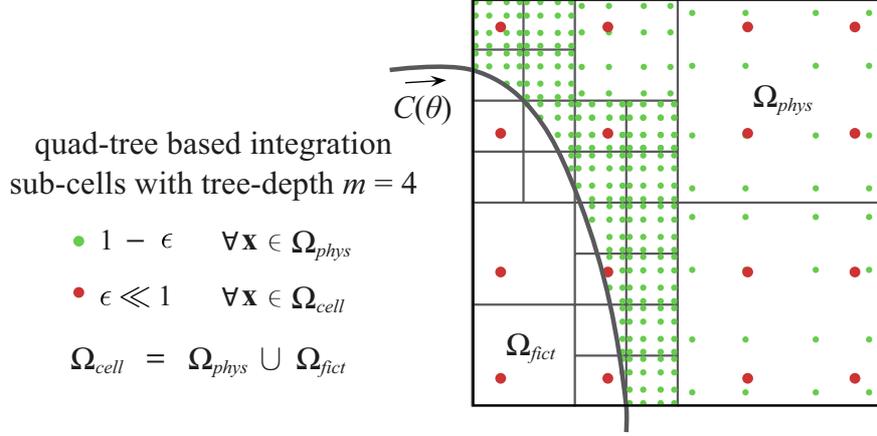


Figure 4: The finite cell principle for a trimmed shell element: sub-cells aggregate quadrature points along the trimming curve in the parameter space.

be formulated as a degenerate case of the quadrilateral mapping by collapsing two corners of the integration domain onto each other. For the case shown in Fig. 2b, the mapping reads

$$\begin{bmatrix} \xi_1 \\ \xi_2 \end{bmatrix} = \frac{1}{4} \left((1 + \hat{\xi}_1)(1 + \hat{\xi}_2) \mathbf{P}_2 + (1 - \hat{\xi}_1)(1 + \hat{\xi}_2) \mathbf{P}_4 \right) + \frac{1}{2} \mathbf{C}_1(\hat{\xi}_1) (1 - \hat{\xi}_2) \quad (26)$$

where $\mathbf{C}_1(\hat{\xi}_1)$ is the trimming curve segment parametrized with respect to $\hat{\xi}_1 = [-1, 1]$. The inverse case shown in Fig. 3a results into a pentagonal planar shape in the parametric space. We simply subdivide the pentagon into two quadrilaterals, which fall under the above strategy (see Fig. 3b). The same subdivision concept can be applied to deal with more complex cutting patterns. Each trimmed element with a complex cutting pattern is subdivided into sub-elements until each sub-element corresponds to one of the reparametrization cases mentioned here.

2.3.2. The finite cell method

There exist special cases, where subdivision of a trimmed element does not lead to a practical number of sub-elements that can be represented by one of the basic blending cases described above. In this case, we resort to the subdivision based explicit representation of the trimmed element by quadrature points, which we adopt from the finite cell method [28, 47]. In this approach, the part of the element that is trimmed away is interpreted as a fictitious domain extension. In the fictitious domain, stresses and forces are penalized such that their contribution to the total strain energy becomes insignificant. This enables a smooth extension of the solution into the fictitious domain, so that the approximation of the solution in the physical domain is higher-order accurate and its gradients remain unaffected up to the geometric boundary [48].

The penalization approach is based on an indicator function which is one in the physical domain and zero in the fictitious domain. We note that maintaining a small finite number $\epsilon \ll 1$ in the fictitious domain can improve the conditioning of the discrete system, while not affecting the error at practical engineering accuracy levels. The integral of an arbitrary function $f(\mathbf{x})$ over a

trimmed element domain Ω^e can then be evaluated as

$$\int_{\Omega^e} f(\mathbf{x}) d\Omega = \int_{\Omega^e} \epsilon f(\mathbf{x}) d\Omega + \int_{\Omega_{\text{phys}}^e} (1 - \epsilon) f(\mathbf{x}) d\Omega. \quad (27)$$

Figure 4 illustrates the quadrature concept for a trimmed shell element in the parameter space. Following (27), we first integrate the function penalized by factor $\epsilon \ll 1$ over the complete element domain, using standard Gauss quadrature (red points in Fig. 4). We then integrate the function multiplied by factor $(1 - \epsilon)$ only in the physical domain (green points in Fig. 4). To resolve the discontinuity in the indicator function along the trimming curve, we employ a quad-tree based sub-cell integration scheme, which aggregates quadrature points around the trimming curve. Our integration scheme, however, only uses the points in the physical domain. The quadrature scheme based on (27) therefore reduces the number of quadrature points by factor two with respect to the original sub-cell quadrature scheme in the finite cell method that integrated the physical and fictitious domains separately. For details on algorithms and data structures, the interested reader is referred for instance to [22].

3. A variationally consistent geometrically nonlinear coupling formulation

In the spirit of NITSCHKE [49], the weak form (16) can be supplemented by consistent flux terms along a boundary or interface that variationally impose Dirichlet boundary and coupling conditions [7, 34]. These terms naturally derive from a weighted residual formulation that satisfies displacement and rotational boundary conditions in a weak sense. In the following, we extend the existing small displacement shell formulation to geometrically nonlinear shell problems. We first transfer the trimmed surface configuration into a mechanics-based interface problem. We then present a detailed derivation of the variational coupling terms, including their consistent linearization. Finally, we discuss the estimation of suitable element-wise stabilization parameters.

3.1. Multipatch trimming as a mechanics-based interface problem

A geometric configuration represented by multiple trimmed patches and trimming curves can be interpreted as an interface problem, where different independent domains need to be coupled along a set of mutually influencing interfaces Γ_{\cup} [50]. The simplest possible configuration is the two-domain problem illustrated in Fig. 5, where $\boldsymbol{\nu}$ is the outward unit normal vector at the interface Γ_{\cup} . In the following, we will refer to the two-domain problem for the sake of clarity of our presentation. We emphasize that it is straightforward to generalize our formulation to an arbitrary number of domains and trimming curves.

From a mechanics viewpoint, we need to formulate suitable coupling conditions that define the interrelation of mechanical fields across interfaces. For a structures problem, coupling conditions are based on displacement continuity and force compatibility at the coupling interface. For the simple two-domain case, they can be expressed as follows

$$\mathbf{u}^{(1)} - \mathbf{u}^{(2)} = \mathbf{0} \quad \text{on } \Gamma_{\cup} \quad (28)$$

$$\mathbf{S}^{(1)} \boldsymbol{\nu}^{(1)} + \mathbf{S}^{(2)} \boldsymbol{\nu}^{(2)} = \mathbf{0} \quad \text{on } \Gamma_{\cup} \quad (29)$$

where $(\mathbf{S} \boldsymbol{\nu})$ defines the traction along the coupling interface Γ_{\cup} . The superscripts $(\cdot)^{(1),(2)}$ indicate for each quantity to which of the two domains it belongs.

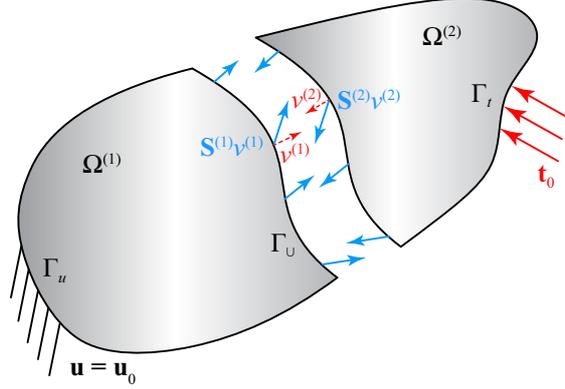


Figure 5: A mechanics-based interface problem, consisting of two domains.

3.2. Relevant identities from shell kinematics and differential geometry

In the next step, we summarize a number of identities in shell kinematics and differential geometry that we need to accurately express our Nitsche-type coupling formulation later on. We define the vector \mathbf{N}^β as

$$\mathbf{N}^\beta = Q^{\alpha\beta} \mathbf{A}_\alpha + Q^\beta \mathbf{A}_3 \quad (30)$$

where $Q^{\alpha\beta}$ and Q^β are the in-plane and out-of-plane forces in the reference configuration [51, 52]. They can be computed from

$$Q^{\alpha\beta} = N^{\alpha\beta} - b_\lambda^\alpha M^{\lambda\beta} \quad (31)$$

$$Q^\beta = M^{\alpha\beta}|_\alpha + a^{\beta\gamma} (2\varepsilon_{\gamma\lambda}|\mu - \varepsilon_{\lambda\mu}|\gamma) M^{\lambda\mu}. \quad (32)$$

Here, $N^{\alpha\beta}$ and $M^{\alpha\beta}$ denote the second Piola-Kirchhoff stress resultants and bending moment tensors, respectively. In (31) and (32), $\varepsilon_{\gamma\lambda}$ is the membrane strain tensor (5) and b_λ^α is the tensor of the mixed components of the surface's second fundamental form. The latter is expressed as

$$b_\lambda^\alpha = a^{\alpha\beta} b_{\lambda\beta} = (\mathbf{a}^\alpha \cdot \mathbf{a}^\beta) (\mathbf{a}_{\lambda,\beta} \cdot \mathbf{a}_3). \quad (33)$$

The covariant derivatives of the tensor fields $M^{\alpha\beta}$ and $\varepsilon_{\lambda\mu}$ can be written as

$$M^{\alpha\beta}|_\gamma = M^{\alpha\beta}_{,\gamma} + \Gamma_{\lambda\gamma}^\alpha M^{\lambda\beta} + \Gamma_{\lambda\gamma}^\beta M^{\alpha\lambda} \quad (34)$$

$$\varepsilon_{\lambda\mu}|_\gamma = \varepsilon_{\lambda\mu,\gamma} - \Gamma_{\lambda\gamma}^\alpha \varepsilon_{\alpha\mu} - \Gamma_{\mu\gamma}^\alpha \varepsilon_{\lambda\alpha} \quad (35)$$

where the restriction in conjunction with a Greek subscript denotes the covariant surface differentiation with respect to a curvilinear coordinate ξ_α . Here, $\Gamma_{\lambda\gamma}^\alpha$ is the Christoffel symbol on a surface in the reference configuration. It is expressed as

$$\Gamma_{\lambda\gamma}^\alpha = A^{\beta\alpha} \Gamma_{\lambda\gamma\beta} \quad (36)$$

in which

$$\Gamma_{\lambda\gamma\beta} = \frac{1}{2} (A_{\gamma\beta,\lambda} + A_{\lambda\beta,\gamma} - A_{\lambda\gamma,\beta}). \quad (37)$$

For more details, we refer interested readers to [43].

3.3. Variational formulation of Nitsche-type coupling terms

In the spirit of Nitsche's method, we can extend the equilibrium equation (16) by suitable coupling terms in variational format as follows:

$$\mathcal{W}_I + \mathcal{W}_I^N - \mathcal{W}_E = \mathbf{0} \quad (38)$$

where \mathcal{W}_I^N contains the work contributions of the Nitsche-type extension and corresponding stabilization terms. Following the classical format, our Nitsche extension \mathcal{W}_I^N can be split into consistency and adjoint-consistency terms, $\mathcal{W}_I^{N,c}$, as well as stabilization terms, $\mathcal{W}_I^{N,s}$. The consistency and adjoint-consistency terms are responsible for weakly enforcing the coupling constraints between Kirchhoff-Love shell patches at the trimming interface. They read

$$\begin{aligned} \mathcal{W}_I^{N,c} = & - \int_{\Gamma_U} \delta \{ \mathbf{F} \mathbf{N}^\beta \nu_\beta \} \cdot \llbracket \mathbf{u} \rrbracket d\Gamma_U - \int_{\Gamma_U} \{ \mathbf{F} \mathbf{N}^\beta \nu_\beta \} \cdot \delta \llbracket \mathbf{u} \rrbracket d\Gamma_U \\ & - \int_{\Gamma_U} \delta \{ M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta \} \cdot \llbracket \Phi \rrbracket d\Gamma_U - \int_{\Gamma_U} \{ M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta \} \cdot \delta \llbracket \Phi \rrbracket d\Gamma_U. \end{aligned} \quad (39)$$

The deformation gradient is defined as $\mathbf{F} = \mathbf{a}_\alpha \otimes \mathbf{A}^\alpha + \mathbf{a}_3 \otimes \mathbf{A}_3$, where we assume that the change of area of the shell element remains small. The term Φ represents the rotation. In the scope of this work, we assume G^1 continuity between patches along trimming curves. In this case, the rotation vector can be computed by the simple relation $\Phi = \mathbf{a}_3 - \mathbf{A}_3$. If a non-zero angle between patches needs to be maintained, the rotation vector needs to be defined in a different way, see e.g. [5].

We define the bracket operators $\{ \cdot \}$ and $\llbracket \cdot \rrbracket$ for the different terms as follows:

$$\{ \mathbf{F} \mathbf{N}^\beta \nu_\beta \} := \gamma (\mathbf{F} \mathbf{N}^\beta \nu_\beta)^{(1)} + (1 - \gamma) (\mathbf{F} \mathbf{N}^\beta \nu_\beta)^{(2)} \quad (40)$$

$$\{ M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta \} := \gamma (M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta)^{(1)} + (1 - \gamma) (M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta)^{(2)} \quad (41)$$

$$\llbracket \mathbf{u} \rrbracket := \mathbf{u}^{(1)} - \mathbf{u}^{(2)} \quad (42)$$

$$\llbracket \Phi \rrbracket := \Phi^{(1)} - \Phi^{(2)}. \quad (43)$$

Equations (40) and (41) use a weighted average, where the value of γ scales the contribution of each of the two coupled domains, $\Omega^{(1)}$ and $\Omega^{(2)}$, to enforce the traction compatibility condition. In the extreme case $\gamma = \{0, 1\}$, the condition is fully shifted to one of the domains, leaving the kinematic conditions (42) and (43) untouched. When not stated differently, we use $\gamma = 0.5$.

The stabilization terms are required to maintain coercivity of the bilinear form and read

$$\mathcal{W}_I^{N,s} = \int_{\Gamma_U} \alpha_u \delta \llbracket \mathbf{u} \rrbracket \cdot \llbracket \mathbf{u} \rrbracket d\Gamma_U + \int_{\Gamma_U} \alpha_\theta \delta \llbracket \Phi \rrbracket \cdot \llbracket \Phi \rrbracket d\Gamma_U \quad (44)$$

where the first integral uses jumps in displacements and the second integral uses jumps in rotations around the shell's mid-surface, with corresponding stabilization parameters α_u and α_θ , respectively. For optimal performance of the method, α_u and α_θ need to be chosen as small as possible. This can be ensured by the element-wise configuration dependent estimation of the stabilization parameters [21, 23, 24, 50]. We will come back to choosing stabilization parameters based on a local eigenvalue problem in Section 3.5.

3.4. Isogeometric discretization and linearization

The contributions of the consistency and adjoint-consistency terms (39) as well as the stabilization terms (44) to the discretized equilibrium equation (19) can be formulated as:

$$\begin{aligned} \mathbf{f}_I^{N,c} &= - \int_{\Gamma_U} \frac{\partial \{\mathbf{FN}^\beta \nu_\beta\}}{\partial \mathbf{U}} \cdot \llbracket \mathbf{u} \rrbracket d\Gamma_U - \int_{\Gamma_U} \{\mathbf{FN}^\beta \nu_\beta\} \cdot \frac{\partial \llbracket \mathbf{u} \rrbracket}{\partial \mathbf{U}} d\Gamma_U \\ &\quad - \int_{\Gamma_U} \frac{\partial \{M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\}}{\partial \mathbf{U}} \cdot \llbracket \Phi \rrbracket d\Gamma_U - \int_{\Gamma_U} \{M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\} \cdot \frac{\partial \llbracket \Phi \rrbracket}{\partial \mathbf{U}} d\Gamma_U \end{aligned} \quad (45)$$

$$\mathbf{f}_I^{N,s} = \int_{\Gamma_U} \alpha_u \frac{\partial \llbracket \mathbf{u} \rrbracket}{\partial \mathbf{U}} \cdot \llbracket \mathbf{u} \rrbracket d\Gamma_U + \int_{\Gamma_U} \alpha_\theta \frac{\partial \llbracket \Phi \rrbracket}{\partial \mathbf{U}} \cdot \llbracket \Phi \rrbracket d\Gamma_U. \quad (46)$$

The linearization of (45) and (46) in the reference configuration yields the tangent stiffness contributions of the consistency and adjoint-consistency terms, $\mathbf{K}_T^{N,c}$, and stabilization terms, $\mathbf{K}_T^{N,s}$, respectively. They are formulated as:

$$\begin{aligned} \mathbf{K}_T^{N,c} &= - \int_{\Gamma_U} \frac{\partial \{\mathbf{FN}^\beta \nu_\beta\}}{\partial \mathbf{U}} \cdot \frac{\partial \llbracket \mathbf{u} \rrbracket}{\partial \mathbf{U}} d\Gamma_U - \left[\int_{\Gamma_U} \frac{\partial \{\mathbf{FN}^\beta \nu_\beta\}}{\partial \mathbf{U}} \cdot \frac{\partial \llbracket \mathbf{u} \rrbracket}{\partial \mathbf{U}} d\Gamma_U \right]^T \\ &\quad - \int_{\Gamma_U} \frac{\partial^2 \{\mathbf{FN}^\beta \nu_\beta\}}{\partial \mathbf{U} \partial \mathbf{U}} \cdot \llbracket \mathbf{u} \rrbracket d\Gamma_U \\ &\quad - \int_{\Gamma_U} \frac{\partial \{M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\}}{\partial \mathbf{U}} \cdot \frac{\partial \llbracket \Phi \rrbracket}{\partial \mathbf{U}} d\Gamma_U - \left[\int_{\Gamma_U} \frac{\partial \{M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\}}{\partial \mathbf{U}} \cdot \frac{\partial \llbracket \Phi \rrbracket}{\partial \mathbf{U}} d\Gamma_U \right]^T \\ &\quad - \int_{\Gamma_U} \{M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\} \cdot \frac{\partial^2 \llbracket \Phi \rrbracket}{\partial \mathbf{U} \partial \mathbf{U}} d\Gamma_U - \int_{\Gamma_U} \frac{\partial^2 \{M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\}}{\partial \mathbf{U} \partial \mathbf{U}} \cdot \llbracket \Phi \rrbracket d\Gamma_U \end{aligned} \quad (47)$$

$$\begin{aligned} \mathbf{K}_T^{N,s} &= \int_{\Gamma_U} \alpha_u \frac{\partial \llbracket \mathbf{u} \rrbracket}{\partial \mathbf{U}} \cdot \frac{\partial \llbracket \mathbf{u} \rrbracket}{\partial \mathbf{U}} d\Gamma_U \\ &\quad + \int_{\Gamma_U} \alpha_\theta \frac{\partial \llbracket \Phi \rrbracket}{\partial \mathbf{U}} \cdot \frac{\partial \llbracket \Phi \rrbracket}{\partial \mathbf{U}} d\Gamma_U + \int_{\Gamma_U} \alpha_\theta \frac{\partial^2 \llbracket \Phi \rrbracket}{\partial \mathbf{U} \partial \mathbf{U}} \cdot \llbracket \Phi \rrbracket d\Gamma_U. \end{aligned} \quad (48)$$

We note that in (47) and (48), we have already dropped the following two terms

$$\int_{\Gamma_U} \{\mathbf{FN}^\beta \nu_\beta\} \cdot \frac{\partial^2 \llbracket \mathbf{u} \rrbracket}{\partial \mathbf{U} \partial \mathbf{U}} d\Gamma_U \quad (49)$$

$$\int_{\Gamma_U} \alpha_u \frac{\partial^2 \llbracket \mathbf{u} \rrbracket}{\partial \mathbf{U} \partial \mathbf{U}} \cdot \llbracket \mathbf{u} \rrbracket d\Gamma_U \quad (50)$$

due to the interpolation relation of the displacement field in (17).

Up to this point, we have expressed derivatives with respect to displacement variables in terms of the vector \mathbf{U} , which contains all displacement variables. In the remainder of this subsection, we will take derivatives of boundary forces and moments with respect to the displacement variables at

an individual control point, \mathbf{U}_r , instead of the vector \mathbf{U} , since this simplifies many derivative expressions. The vector \mathbf{U}_r represents the set of three unknown displacement coefficients associated with the control point \mathbf{B}_r . For quantities that involve second derivatives, we introduce a second control point variable index, \mathbf{U}_s .

In (47), the first and second derivatives of $\{\mathbf{F}\mathbf{N}^\beta \nu_\beta\}$ with respect to the displacement variables \mathbf{U}_r can be written as:

$$\{\mathbf{F}\mathbf{N}^\beta \nu_\beta\}_{,\mathbf{U}_r} = \{\mathbf{F}_{,\mathbf{U}_r}(\mathbf{N}^\beta \nu_\beta)\} + \{\mathbf{F}(\mathbf{N}_{,\mathbf{U}_r}^\beta \nu_\beta)\} \quad (51)$$

$$\begin{aligned} \{\mathbf{F}\mathbf{N}^\beta \nu_\beta\}_{,\mathbf{U}_r\mathbf{U}_s} &= \{\mathbf{F}_{,\mathbf{U}_r\mathbf{U}_s}(\mathbf{N}^\beta \nu_\beta)\} + \{\mathbf{F}_{,\mathbf{U}_r}(\mathbf{N}_{,\mathbf{U}_s}^\beta \nu_\beta)\} \\ &\quad + \{\mathbf{F}_{,\mathbf{U}_s}(\mathbf{N}_{,\mathbf{U}_r}^\beta \nu_\beta)\} + \{\mathbf{F}(\mathbf{N}_{,\mathbf{U}_r\mathbf{U}_s}^\beta \nu_\beta)\} \end{aligned} \quad (52)$$

where we denote the derivatives $\partial(\cdot)/\partial\mathbf{U}_r$ and $\partial(\cdot)/\partial\mathbf{U}_r\partial\mathbf{U}_s$ as $(\cdot)_{,\mathbf{U}_r}$ and $(\cdot)_{,\mathbf{U}_r\mathbf{U}_s}$. In (51) and (52), the following additional relations hold:

$$\mathbf{F}_{,\mathbf{U}_r} = \mathbf{a}_{\alpha,\mathbf{U}_r} \otimes \mathbf{A}^\alpha + \mathbf{a}_{3,\mathbf{U}_r} \otimes \mathbf{A}_3 \quad (53)$$

$$\mathbf{F}_{,\mathbf{U}_r\mathbf{U}_s} = \mathbf{a}_{3,\mathbf{U}_r\mathbf{U}_s} \otimes \mathbf{A}_3 \quad (54)$$

$$\mathbf{N}_{,\mathbf{U}_r}^\beta = Q_{,\mathbf{U}_r}^{\alpha\beta} \mathbf{A}_\alpha + Q_{,\mathbf{U}_r}^\beta \mathbf{A}_3 \quad (55)$$

$$\mathbf{N}_{,\mathbf{U}_r\mathbf{U}_s}^\beta = Q_{,\mathbf{U}_r\mathbf{U}_s}^{\alpha\beta} \mathbf{A}_\alpha + Q_{,\mathbf{U}_r\mathbf{U}_s}^\beta \mathbf{A}_3 \quad (56)$$

where the definitions of $Q^{\alpha\beta}$ and Q^β can be found in (31) and (32).

Following (31), the first and second derivatives of the in-plane force $Q^{\alpha\beta}$ with respect to the displacement variables are formulated as:

$$Q_{,\mathbf{U}_r}^{\alpha\beta} = N_{,\mathbf{U}_r}^{\alpha\beta} - b_{\lambda,\mathbf{U}_r}^\alpha M^{\lambda\beta} - b_\lambda^\alpha M_{,\mathbf{U}_r}^{\lambda\beta} \quad (57)$$

$$Q_{,\mathbf{U}_r\mathbf{U}_s}^{\alpha\beta} = N_{,\mathbf{U}_r\mathbf{U}_s}^{\alpha\beta} - b_{\lambda,\mathbf{U}_r\mathbf{U}_s}^\alpha M^{\lambda\beta} - b_{\lambda,\mathbf{U}_r}^\alpha M_{,\mathbf{U}_s}^{\lambda\beta} - b_{\lambda,\mathbf{U}_s}^\alpha M_{,\mathbf{U}_r}^{\lambda\beta} - b_\lambda^\alpha M_{,\mathbf{U}_r\mathbf{U}_s}^{\lambda\beta}. \quad (58)$$

In (57) and (58), the following additional relations hold:

$$b_{\lambda,\mathbf{U}_r}^\alpha = a_{,\mathbf{U}_r}^{\alpha\beta} (\mathbf{a}_{\lambda,\beta} \cdot \mathbf{a}_3) + a^{\alpha\beta} (\mathbf{a}_{\lambda,\beta,\mathbf{U}_r} \cdot \mathbf{a}_3 + \mathbf{a}_{\lambda,\beta} \cdot \mathbf{a}_{3,\mathbf{U}_r}) \quad (59)$$

$$\begin{aligned} b_{\lambda,\mathbf{U}_r\mathbf{U}_s}^\alpha &= a_{,\mathbf{U}_r\mathbf{U}_s}^{\alpha\beta} (\mathbf{a}_{\lambda,\beta} \cdot \mathbf{a}_3) + a_{,\mathbf{U}_r}^{\alpha\beta} (\mathbf{a}_{\lambda,\beta,\mathbf{U}_s} \cdot \mathbf{a}_3 + \mathbf{a}_{\lambda,\beta} \cdot \mathbf{a}_{3,\mathbf{U}_s}) \\ &\quad + a_{,\mathbf{U}_s}^{\alpha\beta} (\mathbf{a}_{\lambda,\beta,\mathbf{U}_r} \cdot \mathbf{a}_3 + \mathbf{a}_{\lambda,\beta} \cdot \mathbf{a}_{3,\mathbf{U}_r}) \\ &\quad + a^{\alpha\beta} (\mathbf{a}_{\lambda,\beta,\mathbf{U}_r} \cdot \mathbf{a}_{3,\mathbf{U}_s} + \mathbf{a}_{\lambda,\beta,\mathbf{U}_s} \cdot \mathbf{a}_{3,\mathbf{U}_r} + \mathbf{a}_{\lambda,\beta} \cdot \mathbf{a}_{3,\mathbf{U}_r\mathbf{U}_s}). \end{aligned} \quad (60)$$

For further details, we refer the interested reader to the following appendices: the derivatives $a_{,\mathbf{U}_r}^{\alpha\beta}$ and $a_{,\mathbf{U}_r\mathbf{U}_s}^{\alpha\beta}$ are given in Appendix A.1, the derivatives of \mathbf{a}_3 with respect to the displacement variables are given in Appendix A.2, and the derivatives of the stress resultants $N^{\alpha\beta}$ and the bending moments $M^{\alpha\beta}$ are given in Appendix A.3.

Following (32), The first and second derivatives of the out-of-plane force Q^β with respect to the displacement variables are formulated as:

$$Q_{,\mathbf{U}_r}^\beta = (M^{\alpha\beta}|_\alpha)_{,\mathbf{U}_r} + a^{\beta\gamma} (2\varepsilon_{\gamma\lambda|\mu} - \varepsilon_{\lambda\mu|\gamma}) M_{,\mathbf{U}_r}^{\lambda\mu} + \left(a_{,\mathbf{U}_r}^{\beta\gamma} (2\varepsilon_{\gamma\lambda|\mu} - \varepsilon_{\lambda\mu|\gamma}) + a^{\beta\gamma} ((2\varepsilon_{\gamma\lambda|\mu})_{,\mathbf{U}_r} - (\varepsilon_{\lambda\mu|\gamma})_{,\mathbf{U}_r}) \right) M^{\lambda\mu} \quad (61)$$

$$Q_{,\mathbf{U}_r\mathbf{U}_s}^\beta = (M^{\alpha\beta}|_\alpha)_{,\mathbf{U}_r\mathbf{U}_s} + a^{\beta\gamma} (2\varepsilon_{\gamma\lambda|\mu} - \varepsilon_{\lambda\mu|\gamma}) M_{,\mathbf{U}_r\mathbf{U}_s}^{\lambda\mu} + \left(a_{,\mathbf{U}_s}^{\beta\gamma} (2\varepsilon_{\gamma\lambda|\mu} - \varepsilon_{\lambda\mu|\gamma}) + a^{\beta\gamma} ((2\varepsilon_{\gamma\lambda|\mu})_{,\mathbf{U}_s} - (\varepsilon_{\lambda\mu|\gamma})_{,\mathbf{U}_s}) \right) M_{,\mathbf{U}_r}^{\lambda\mu} + \left(a_{,\mathbf{U}_r}^{\beta\gamma} (2\varepsilon_{\gamma\lambda|\mu} - \varepsilon_{\lambda\mu|\gamma}) + a^{\beta\gamma} ((2\varepsilon_{\gamma\lambda|\mu})_{,\mathbf{U}_r} - (\varepsilon_{\lambda\mu|\gamma})_{,\mathbf{U}_r}) \right) M_{,\mathbf{U}_s}^{\lambda\mu} + \left(a_{,\mathbf{U}_r\mathbf{U}_s}^{\beta\gamma} (2\varepsilon_{\gamma\lambda|\mu} - \varepsilon_{\lambda\mu|\gamma}) + a_{,\mathbf{U}_r}^{\beta\gamma} ((2\varepsilon_{\gamma\lambda|\mu})_{,\mathbf{U}_s} - (\varepsilon_{\lambda\mu|\gamma})_{,\mathbf{U}_s}) + a_{,\mathbf{U}_s}^{\beta\gamma} ((2\varepsilon_{\gamma\lambda|\mu})_{,\mathbf{U}_r} - (\varepsilon_{\lambda\mu|\gamma})_{,\mathbf{U}_r}) + a^{\beta\gamma} ((2\varepsilon_{\gamma\lambda|\mu})_{,\mathbf{U}_r\mathbf{U}_s} - (\varepsilon_{\lambda\mu|\gamma})_{,\mathbf{U}_r\mathbf{U}_s}) \right) M^{\lambda\mu} \quad (62)$$

where the terms $(M^{\alpha\beta}|_\alpha)_{,\mathbf{U}_r}$, $(M^{\alpha\beta}|_\alpha)_{,\mathbf{U}_r\mathbf{U}_s}$, $(\varepsilon_{\lambda\mu|\gamma})_{,\mathbf{U}_r}$ and $(\varepsilon_{\lambda\mu|\gamma})_{,\mathbf{U}_r\mathbf{U}_s}$ derive from the following relations:

$$(M^{\alpha\beta}|_\alpha)_{,\mathbf{U}_r} = M_{,\gamma}^{\alpha\beta} + \Gamma_{\lambda\gamma}^\alpha M_{,\mathbf{U}_r}^{\lambda\beta} + \Gamma_{\lambda\gamma}^\beta M_{,\mathbf{U}_r}^{\alpha\lambda} \quad (63)$$

$$(M^{\alpha\beta}|_\alpha)_{,\mathbf{U}_r\mathbf{U}_s} = M_{,\gamma}^{\alpha\beta}{}_{,\mathbf{U}_r\mathbf{U}_s} + \Gamma_{\lambda\gamma}^\alpha M_{,\mathbf{U}_r\mathbf{U}_s}^{\lambda\beta} + \Gamma_{\lambda\gamma}^\beta M_{,\mathbf{U}_r\mathbf{U}_s}^{\alpha\lambda} \quad (64)$$

$$(\varepsilon_{\lambda\mu|\gamma})_{,\mathbf{U}_r} = \varepsilon_{\lambda\mu,\gamma}{}_{,\mathbf{U}_r} - \Gamma_{\lambda\gamma}^\alpha \varepsilon_{\alpha\mu}{}_{,\mathbf{U}_r} - \Gamma_{\mu\gamma}^\alpha \varepsilon_{\lambda\alpha}{}_{,\mathbf{U}_r} \quad (65)$$

$$(\varepsilon_{\lambda\mu|\gamma})_{,\mathbf{U}_r\mathbf{U}_s} = \varepsilon_{\lambda\mu,\gamma}{}_{,\mathbf{U}_r\mathbf{U}_s} - \Gamma_{\lambda\gamma}^\alpha \varepsilon_{\alpha\mu}{}_{,\mathbf{U}_r\mathbf{U}_s} - \Gamma_{\mu\gamma}^\alpha \varepsilon_{\lambda\alpha}{}_{,\mathbf{U}_r\mathbf{U}_s} \quad (66)$$

where the Christoffel symbol defined in (36) refers to the reference configuration.

In (47), the first and second derivatives of $\{M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\}$ with respect to the displacement variables can be written as:

$$\{M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\}_{,\mathbf{U}_r} = \{M_{,\mathbf{U}_r}^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\} + \{M^{\alpha\beta} \mathbf{a}_{\alpha,\mathbf{U}_r} \nu_\beta\} \quad (67)$$

$$\{M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\}_{,\mathbf{U}_r\mathbf{U}_s} = \{M_{,\mathbf{U}_r\mathbf{U}_s}^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta\} + \{M_{,\mathbf{U}_r}^{\alpha\beta} \mathbf{a}_{\alpha,\mathbf{U}_s} \nu_\beta\} + \{M_{,\mathbf{U}_s}^{\alpha\beta} \mathbf{a}_{\alpha,\mathbf{U}_r} \nu_\beta\}. \quad (68)$$

For further details, we again refer the interested reader to the following appendices: the derivatives of bending moments with respect to the displacement variables can be found in Appendix A.3. The derivatives of the relative changes of rotations $[\Phi]$ with respect to the displacement variables, used in (45) through (48), are explained further in Appendix A.2.

3.5. Element-wise estimation of stabilization parameters

If stabilization parameters are chosen globally, the element with the worst cut will dominate the performance of the system, which is likely to lead to a loss of accuracy. We therefore adopt a local strategy based on an element-wise eigenvalue problem, where stabilization parameters can be estimated individually for each trimmed element. In the following, we briefly outline the method.

According to a generalized inverse estimate [23], there exist mesh-dependent positive constants C_u and C_θ such that the boundary reactions of the two coupled shell patches are bounded by the strain energy of the two boundary domains in the form:

$$\| \{ \mathbf{FN}^\beta \nu_\beta \} \|_{\Gamma_U}^2 \leq C_u (\mathcal{W}_I^{(1)} + \mathcal{W}_I^{(2)}) \quad (69)$$

$$\| \{ M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta \} \|_{\Gamma_U}^2 \leq C_\theta (\mathcal{W}_I^{(1)} + \mathcal{W}_I^{(2)}) \quad (70)$$

where $\| \cdot \|$ is the L^2 -norm and $\{ \cdot \}$ is the average operator defined in (40) and (41).

Making use of Young's inequality, we can find a lower bound of the internal work, including the Nitsche terms, that shows that coercivity of the discrete problem is satisfied when $\alpha_u > 2C_u$ and $\alpha_\theta > 2C_\theta$. We can determine suitable estimates of C_u and C_θ from the largest eigenvalue of a generalized eigenvalue problem of the form $\mathbf{A}\mathbf{u} = \lambda \mathbf{B}\mathbf{u}$.

The matrix \mathbf{B} reads:

$$\mathbf{B} = \sum_{k=1}^2 \int_{\Omega_k} \left(\frac{\partial \mathbf{N}}{\partial \mathbf{U}} : \frac{\partial \boldsymbol{\varepsilon}}{\partial \mathbf{U}} + \mathbf{N} : \frac{\partial^2 \boldsymbol{\varepsilon}}{\partial \mathbf{U} \partial \mathbf{U}} + \frac{\partial \mathbf{M}}{\partial \mathbf{U}} : \frac{\partial \boldsymbol{\kappa}}{\partial \mathbf{U}} + \mathbf{M} : \frac{\partial^2 \boldsymbol{\kappa}}{\partial \mathbf{U} \partial \mathbf{U}} \right) d\Omega. \quad (71)$$

For estimating C_u , the matrix \mathbf{A} reads:

$$\mathbf{A}^u = 2 \int_{\Gamma_U} \left(\frac{\partial \{ \mathbf{FN}^\beta \nu_\beta \}}{\partial \mathbf{U}} \cdot \frac{\partial \{ \mathbf{FN}^\beta \nu_\beta \}}{\partial \mathbf{U}} + \frac{\partial^2 \{ \mathbf{FN}^\beta \nu_\beta \}}{\partial \mathbf{U} \partial \mathbf{U}} \cdot \{ \mathbf{FN}^\beta \nu_\beta \} \right) d\Gamma_U. \quad (72)$$

For estimating C_θ , the matrix \mathbf{A} reads:

$$\mathbf{A}^\theta = 2 \int_{\Gamma_U} \left(\frac{\partial \{ M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta \}}{\partial \mathbf{U}} \cdot \frac{\partial \{ M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta \}}{\partial \mathbf{U}} + \frac{\partial^2 \{ M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta \}}{\partial \mathbf{U} \partial \mathbf{U}} \cdot \{ M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta \} \right) d\Gamma_U. \quad (73)$$

We note that the derivatives of $\{ \mathbf{FN}^\beta \nu_\beta \}$ and $\{ M^{\alpha\beta} \mathbf{a}_\alpha \nu_\beta \}$ with respect to the displacement variables \mathbf{U} follow from (51), (52), (67) and (68).

From an algorithmic point of view, we update the stabilization parameters at the beginning of each load step during the geometrically nonlinear analysis. In case of moderate deformations, the stabilization parameters can be updated after several load steps or even kept constant during the complete nonlinear analysis, as their changes can be expected to be insignificant.

4. Extracting trimming and coupling information from ISO 10303 STEP

In this section, we provide a concise overview on how to integrate the geometric description of trimmed CAD surfaces in STEP format and isogeometric trimmed shell analysis. To this end, we first review some essential properties of STEP in terms of file structure and formatting rules that we feel are prerequisites to understanding the information provided in the following sub-sections. We then describe how STEP information can be transferred into quadrature rules for trimmed elements and along trimming curves, illustrating relevant mechanisms with a STEP model of a simple two-patch trimmed surface. We note that throughout this section, all STEP entities are written in capital letters for better readability.

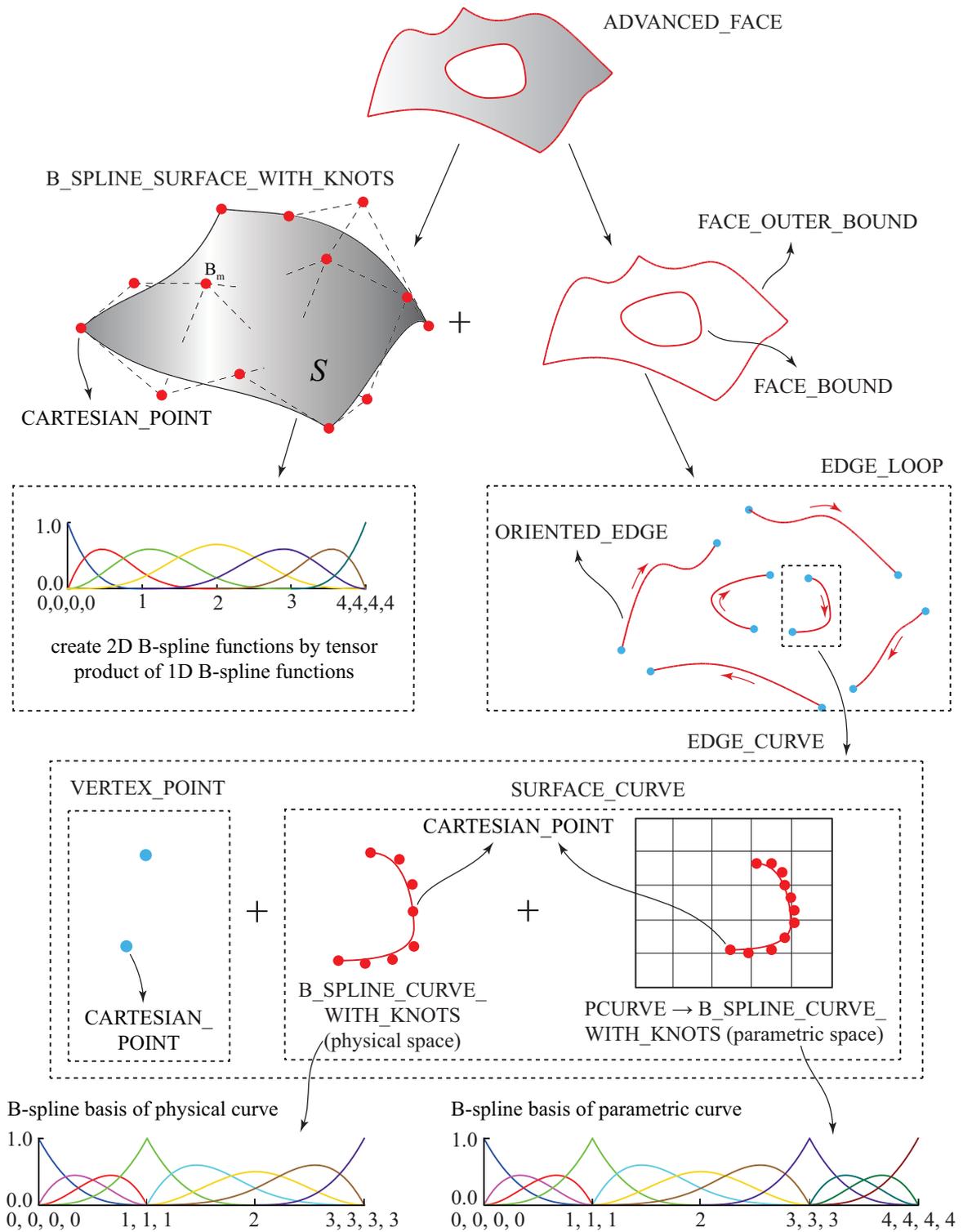


Figure 6: The topology and geometry for the boundary representation of a surface in STEP.

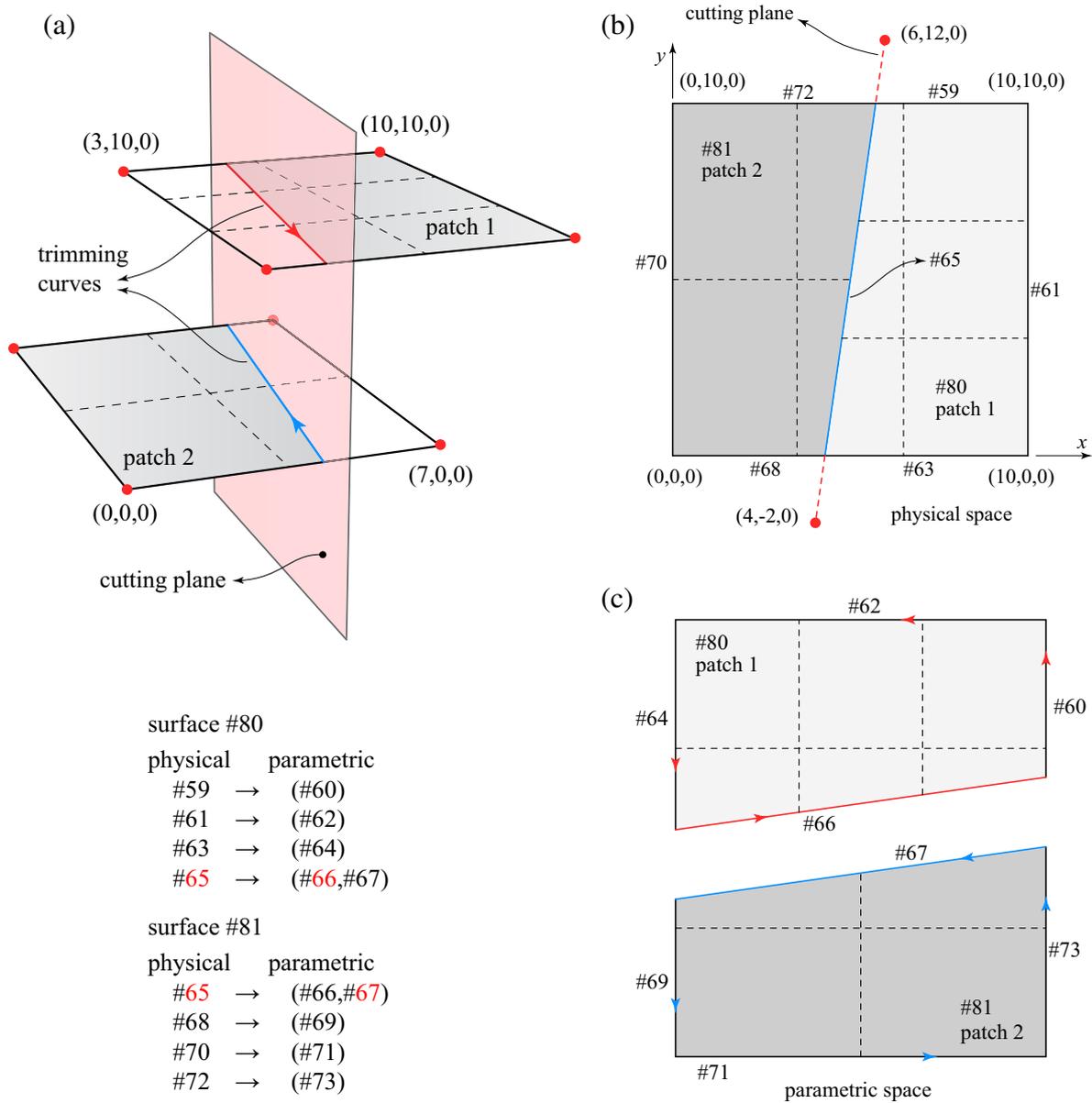


Figure 7: Two trimmed patches with coupling information in STEP format: (a) model description, (b) patches with boundary curves in physical space, (c) patches with boundary curves in parametric space.

4.1. The b-rep structure of STEP

Figure 6 illustrates the *b-rep* (boundary representation) concept, which forms the fundamental basis for representing surfaces in STEP. The entity `ADVANCED_FACE` can be split into a `B_SPLINE_SURFACE_WITH_KNOTS` entity and a set of `FACE_BOUND` entities. Each `FACE_BOUND` entity consists of a number of `ORIENTED_EDGE` entities, represented by a combination of `VERTEX` and `SURFACE_CURVE` entities. The `SURFACE_CURVE` is further mapped to the entities of `B_SPLINE_CURVE_WITH_KNOTS` in both the physical and the parametric spaces of

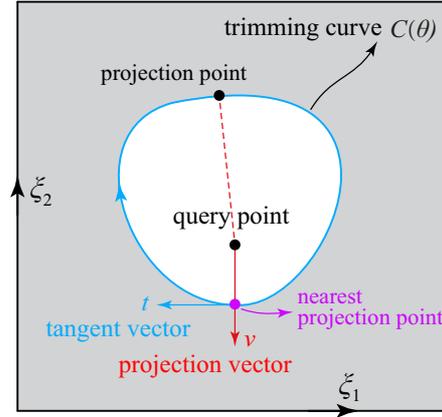


Figure 8: The inside/outside test algorithm in the parametric space.

the surface. The structure of a STEP file consists of two main parts. The HEADER section consists of general information such as FILE_NAME or FILE_SCHEMA. The DATA section contains all the geometric information of the model. In the DATA section, each geometric entity is represented in terms of a specific format, illustrated by the following example:

```
#15=ADVANCED_FACE('', (#17), #80, .T.);
```

where the first number, #15, represents the unique identifier of the instance, followed by the unique entity name, ADVANCED_FACE. Within the ensuing brackets, the entity name is fed by a set of attribute values, such as Booleans, integers, strings, and others, to uniquely define geometric properties and relations to other entities. For a thorough description of the STEP format, we refer interested readers to [3, 53, 54].

We illustrate the STEP b-rep structure with the simple example shown in Fig. 7. Its two trimmed B-spline patches consist of 3×2 and 2×2 cubic Bézier elements, respectively. A join operation is performed between the two patches to combine them as a “solid model” [3]. The complete STEP data of the example is obtained from the CAD software Rhinoceros [55] with a default output setting and is listed in the Appendix B for reference.

4.2. Quadrature of trimmed elements based on STEP

In our framework, quadrature of trimmed elements informed by STEP is implemented in a two step-procedure. We first determine the status of each element and then extract trimming curve segments for each cut element as the basis for the blending function method.

4.2.1. Robust point location query

A central component of our approach is a robust point location query that determines whether a point on a trimmed spline patch is located inside or outside of its physical part. A robust algorithm directly based on geometric information available in STEP can be established as follows. First, we extract the direction of each trimming curve from its entity

```
#24=ORIENTED_EDGE('', *, *, #55, .T.);
```

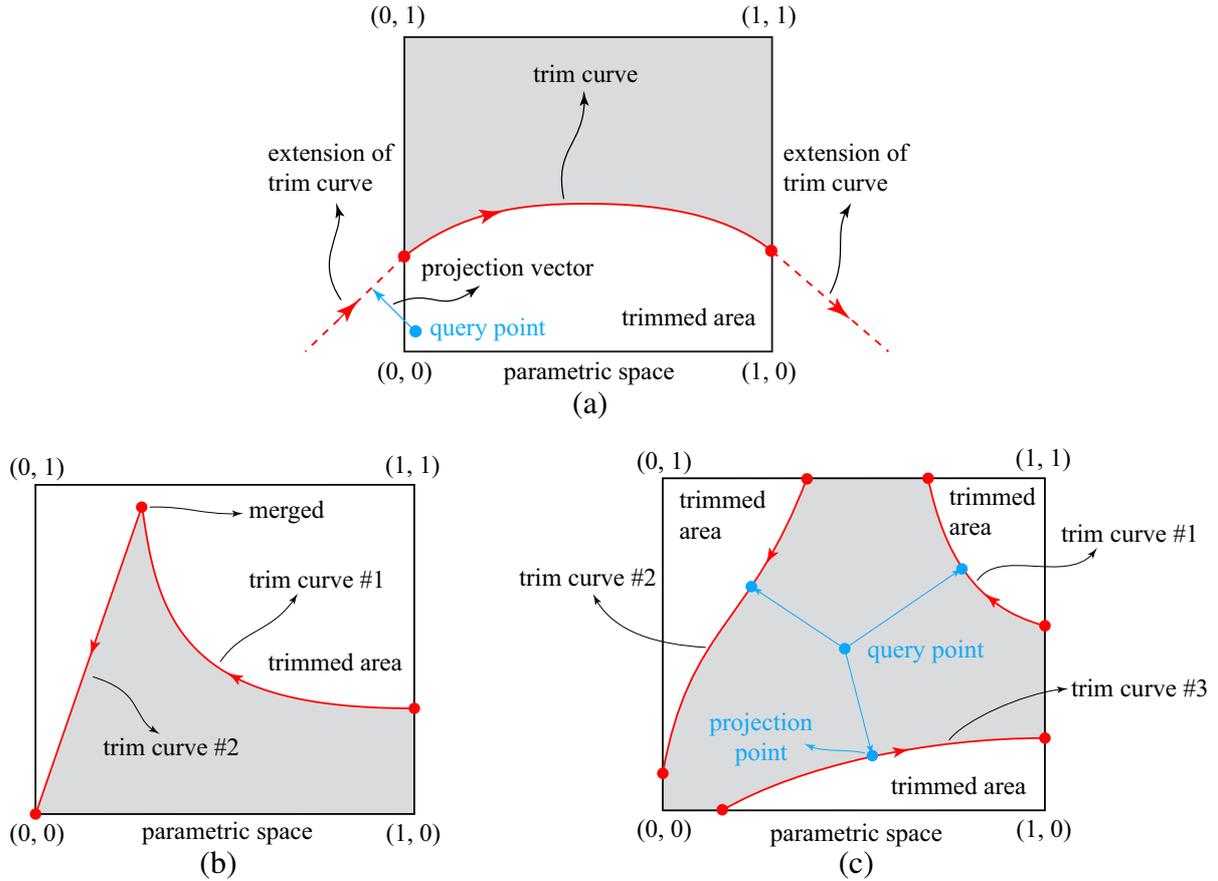


Figure 9: Special cases of trimming curves for a surface patch: (a) extensions of trimming curves in the parametric space, (b) merge operations of trimming curves in the parametric space, (c) multiple cutting-through trimming curves in the parametric space.

where the Boolean value specifies the direction of the trimming curve in the parametric space of the surface patch. In STEP, the direction of the trimming curve determines which part of the surface is visible and which is not. Second, we project the position of a given query point on the surface patch onto the trimming curve in the parametric space of the patch (see Fig. 8). If there exist multiple projection points, we select the one closest to the query point. Third, we form the cross product of the distance vector \mathbf{v} from the query point to its projection point and the tangent vector \mathbf{t} to the curve at the projection point. Based on the positive or negative direction of the resulting vector, we can infer whether the query point is inside or outside the physical part of the trimmed patch. We note that our framework leverages existing geometric algorithms available via the open-source package OPEN CASCADE [56].

There exist several special cases, which need further consideration. One important case occurs when a trimming curve cuts through a patch boundary such that some points on the patch do not have well-defined projections (see Fig. 9a). We resolve this issue by constructing suitable extensions of the trimming curves using the tangent at the end point. Another important case occurs when two or more trimming curves meet within a patch without extending to the patch boundaries

(see Fig. 9b). We resolve this issue by merge operations between the individual trimming curves that create one or multiple closed loops in the parametric space of the patch surface. The merge operation can be interpreted as joining two neighboring B-spline curves with C^0 -continuity. We note that if there are multiple trimming curves cutting through patch boundaries simultaneously, the projection operation has to be performed for each trimming curve separately (see Fig. 9c).

4.2.2. Determination of element status

Following the concept described in [57], we can determine the status of each Bézier element, that is, whether it is full, cut, or void. For each element, the distance d_p between its center point and a trimming curve is calculated (see Fig. 10). If $d_p < r_{min}$, the element is cut, if $d_p > r_{max}$, the element is full or void, depending on the sign of the cross product $\mathbf{v} \times \mathbf{t}$. For the case of $r_{min} < d_p < r_{max}$, we find the projections of the four element vertex points to the trimming curve. The resulting inside/outside information determines the cutting pattern of each trimmed Bézier element, which can consist of a pentagon, triangle or quadrilateral. Following our geometric considerations in the context of the finite cell method [33, 58], we assume that testing the corner vertices captures all relevant cutting patterns. In the scope of this work, all other cutting patterns are assumed irrelevant in the sense that they belong to a geometric scale that cannot be resolved with the current spline discretization. They are therefore neglected without significant effect on the accuracy of the analysis. We note that several methodologies have been published recently that capture arbitrary cutting patterns up to any geometric scale, which should be adopted in future implementation [30, 40, 59].

4.2.3. Generating element-wise trimming curve segments

To generate geometrically accurate quadrature rules, we need to find suitable trimming curve segments in each cut element that can be used in the blending function method reviewed in Section 2.3. In general, trimming curves in the parametric space can be identified by their order p , usually $p = 3$ or even higher, and they are constructed from C^0 interpolations of a set of sampling points (see Fig. 11). Depending on the specific algorithms of each CAD system, these sampling points are not necessarily located at the intersections between the trimming curves and the knot lines of the surface. In our framework, we first determine the parametric values θ_i of all intersection points of the trimming curves with edges of Bézier elements. Each of the two neighboring values are the start and end parametric values of an element trimming segment (see Fig. 11). We generate the trimming curve segments by imposing C^{-1} -continuity at all intersection points. These segments are then employed in the blending function method. For trimming segments with internal C^0 knots, we perform subdivision of the cut element first, such that we can apply the above procedure to each individual sub-element (see Fig. 11).

4.3. Quadrature along trimming curves based on STEP

In STEP, the entity SURFACE_CURVE defines the relationship between a physical curve and the corresponding parametric curve. This is illustrated in the simple STEP model (see Appendix B):

```
#48=SURFACE_CURVE('', #65, (#32, #33), .PCURVE_S1.);
```

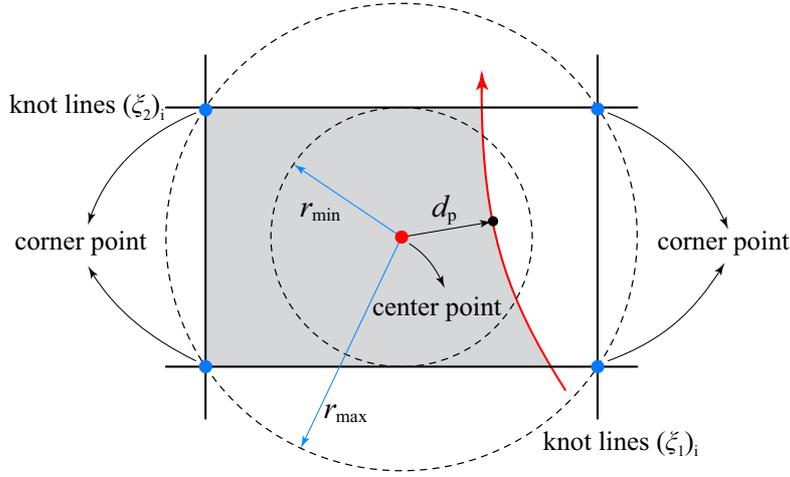


Figure 10: The determination of the element status.

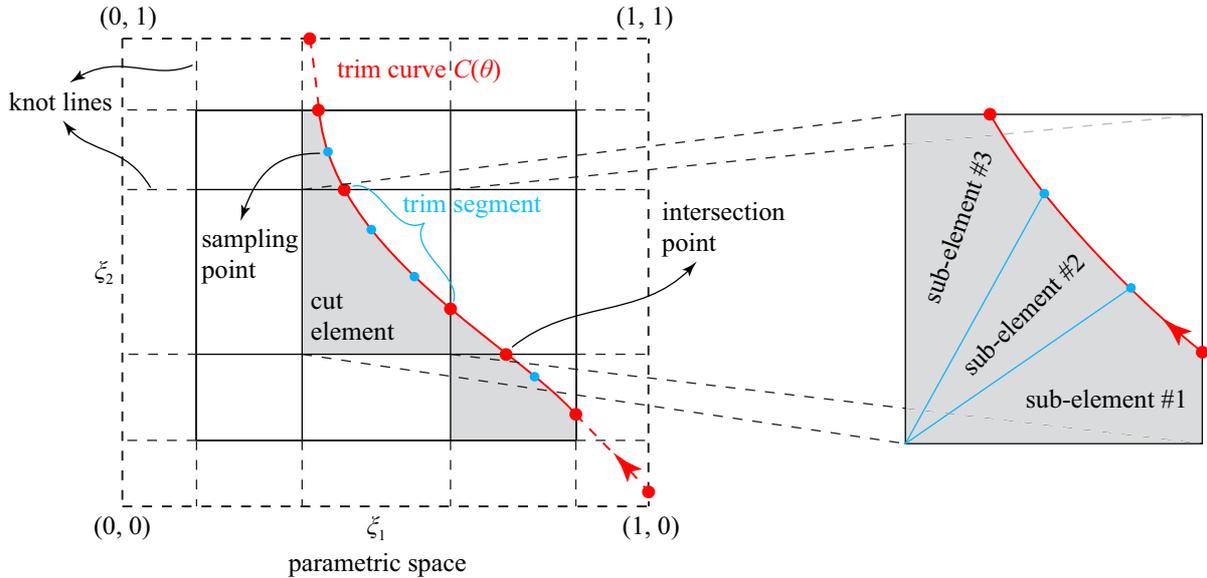


Figure 11: The extraction of trimming curve segment for the cut element.

The physical curve #65 is mapped to PCURVEs #32 and #33, which can be further mapped to two B-spline curves #66 and #67 in the parametric space of the two surface patches along the trimming curve. This information indicates that two patches are coupled along the physical curve #65. The two coupling curves in the parametric spaces can be either trimming curves or non-trimming curves. In this case, both B-spline curves #66 and #67 are trimming curves.

Integration of the variational coupling terms is performed segment-wise along the physical curve #65 (see Fig. 12). We first compute the intersection points of the parametric curves in each surface patch with all knot lines of Bézier elements. We then map the intersection points from each patch to the physical curve and compute the corresponding parametric values on the physical curve. Together with the original knot values of the physical curve, they form integration

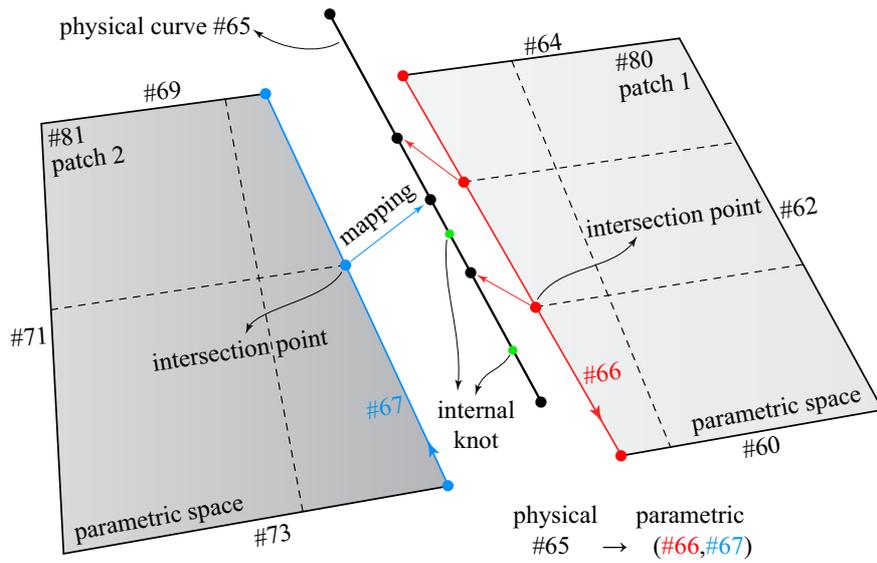


Figure 12: The mapping of intersection points in parametric space to the physical coupling curve.

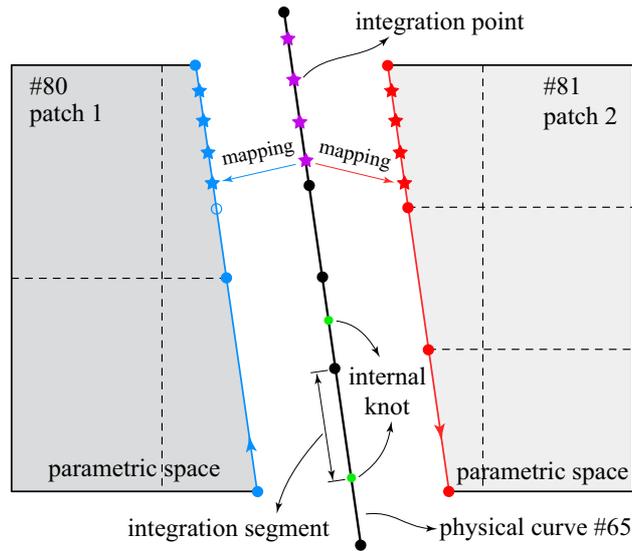


Figure 13: The segment-wise integration of coupling constraints on the physical curve.

knot spans, between which Gaussian quadrature points can be easily placed. For the point-wise evaluation of coupling terms, we map each quadrature point on the physical curve #65 back to the two parametric spaces of the two patches #80 and #81. Simultaneously, we can identify the element IDs to be coupled at this point from the positions of the inversely mapped quadrature point within each patch (see Fig. 13).

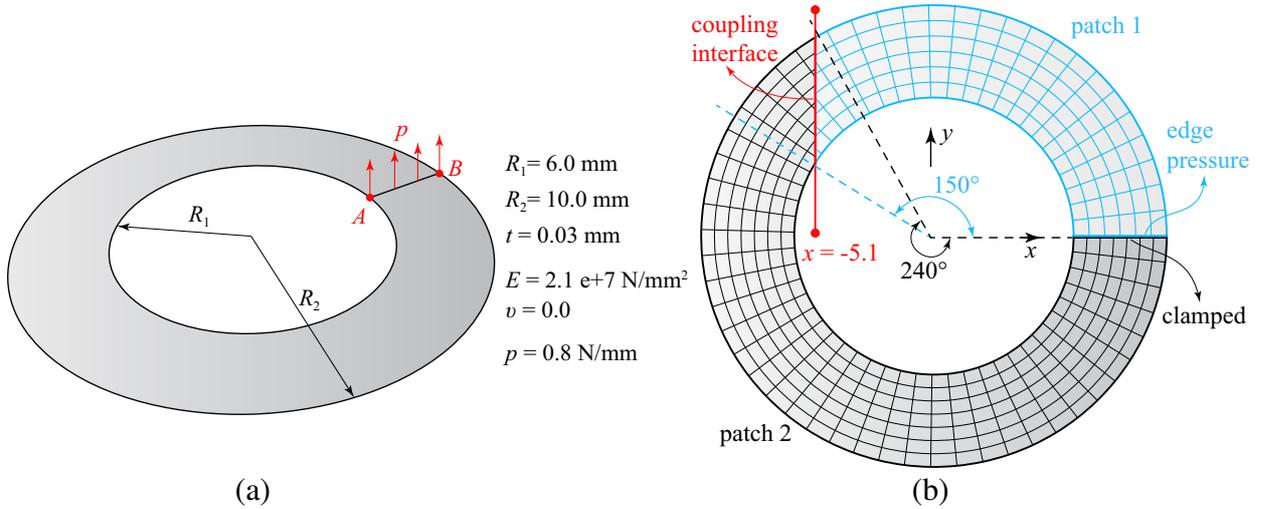


Figure 14: Slit annular plate: (a) model description, (b) two-patch trimmed configuration and NURBS meshes.

5. Numerical examples

In this section, we demonstrate both accuracy and seamless design-through-analysis integration of isogeometric trimmed shell analysis. All isogeometric computations are based on our framework presented in Sections 2, 3 and 4. It features the geometrically nonlinear Nitsche formulation for weakly imposing interface constraints along all trimming curves, where stability of trimmed shell analysis is ensured by element-wise estimation of the stabilization parameters. Quadrature rules for trimmed elements are derived with the help of the blending function method, where trimmed elements that cannot be classified in terms of our set of implemented cutting patterns are integrated with the finite cell method. We created NURBS surfaces for all benchmark examples in the CAD software Rhinoceros [55] and imported them into our trimmed shell analysis framework via the STEP exchange format. To illustrate the advantages of trimmed shell analysis, we also apply our framework for the isogeometric analysis of a Dodge RAM hood that we compare to conventional shell analysis based on Simulia's commercial software package Abaqus.

5.1. Slit annular plate

As the first example, we consider the benchmark of a slit annular plate under uniform line pressure and fixed boundary conditions [60, 61]. The geometry and material properties of the plate are illustrated in Fig. 14a. As detailed in Fig. 14b, the two patches are discretized with 23×6 and 37×7 cubic NURBS elements, and connected along the vertical trimming line $x = -5.1 \text{ mm}$. We use the geometrically nonlinear Nitsche-type formulation, where the load increments of the Newton-Raphson method are set to $\lambda = 0.01$ for the first ten steps, and to $\lambda = 0.05$ for the remaining 18 steps. The values of the stabilization parameters α_u and α_θ are computed with a local eigenvalue analysis at the beginning of the first load step and kept constant during the entire loading process. Comparing the element-wise computed eigenvalues of the first and last load steps, we found that on average, the values changed by approximately 2% for α_θ and by approximately 0.1% for α_u , which justifies our strategy.

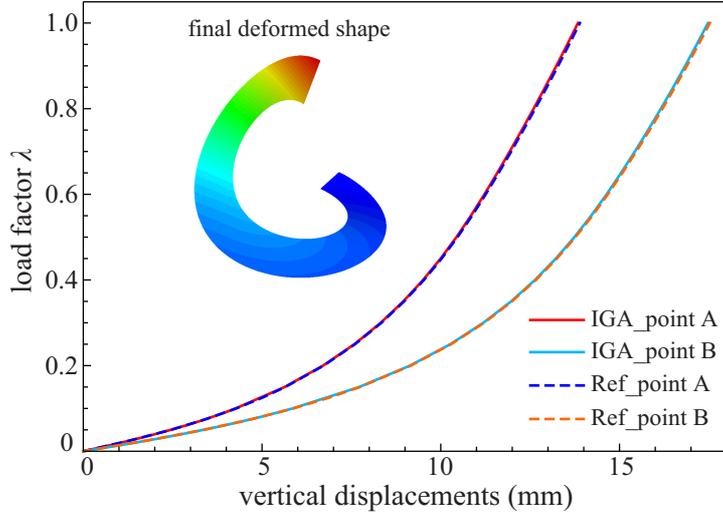


Figure 15: Load-displacement curves for the slit annular plate.

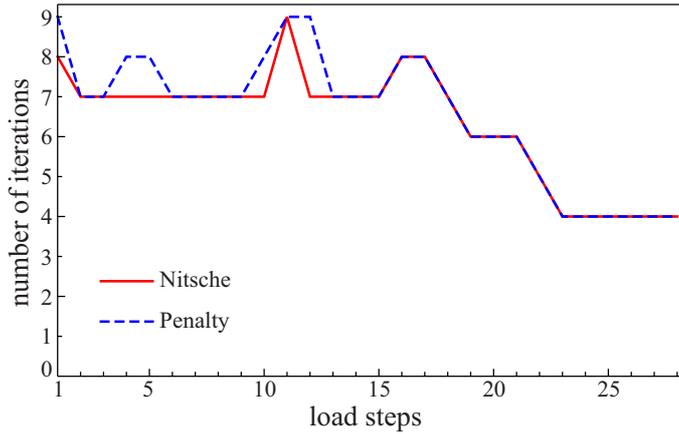


Figure 16: Slit annular plate: number of iterations for Nitsche's method and the penalty method.

Figure 15 compares the load-displacement curves for points A and B obtained with the variationally consistent trimmed shell analysis with reference solutions given in [62], showing excellent agreement. In addition, Figure 16 compares the number of iterations required by the geometrically nonlinear Nitsche-type formulation with the iterations required by a pure penalty approach. For the latter, we choose the penalty factor α to be of the order $10^{k+m/2}$, where k is the order of magnitude of the largest entry of the stiffness matrix and m is the number of significant decimal digits allowed by machine precision, with the proviso that such a choice would not cause arithmetic overflow [63]. For example, if the largest stiffness entry is 10^4 and the machine precision allows 16 significant decimal digits, the penalty parameter is chosen as $\alpha=10^{12}$.

We then compare the accuracy of the von Mises stress across the trimming interface for the geometrically nonlinear Nitsche-type formulation and the penalty method. To this end, we compute the average of the von Mises stress, evaluated at both sides of the trimming interface, and plot the relative deviation from a single-patch overkill discretization along the trimming curve

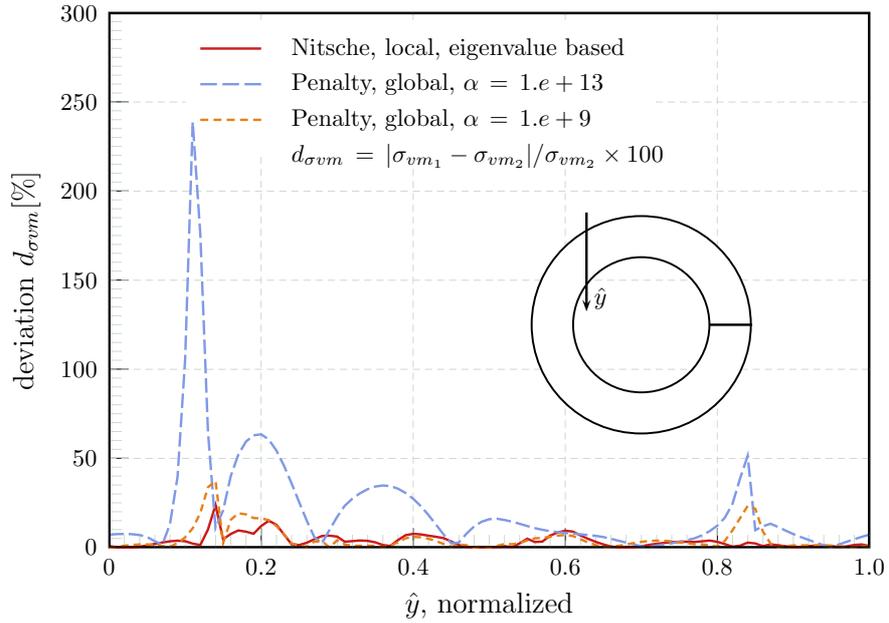


Figure 17: Slit annular plate: relative deviation of the von Mises stress at the interface from a single-patch overkill discretization.

in Fig. 17. We observe that the variationally consistent formulation with element-wise estimates of stabilization parameters yields superior accuracy with respect to the penalty method that uses a global penalty parameter. We observe by comparing Figs. 14 and 17 that the location of the highest relative deviation along the interface occurs at elements with small cuts.

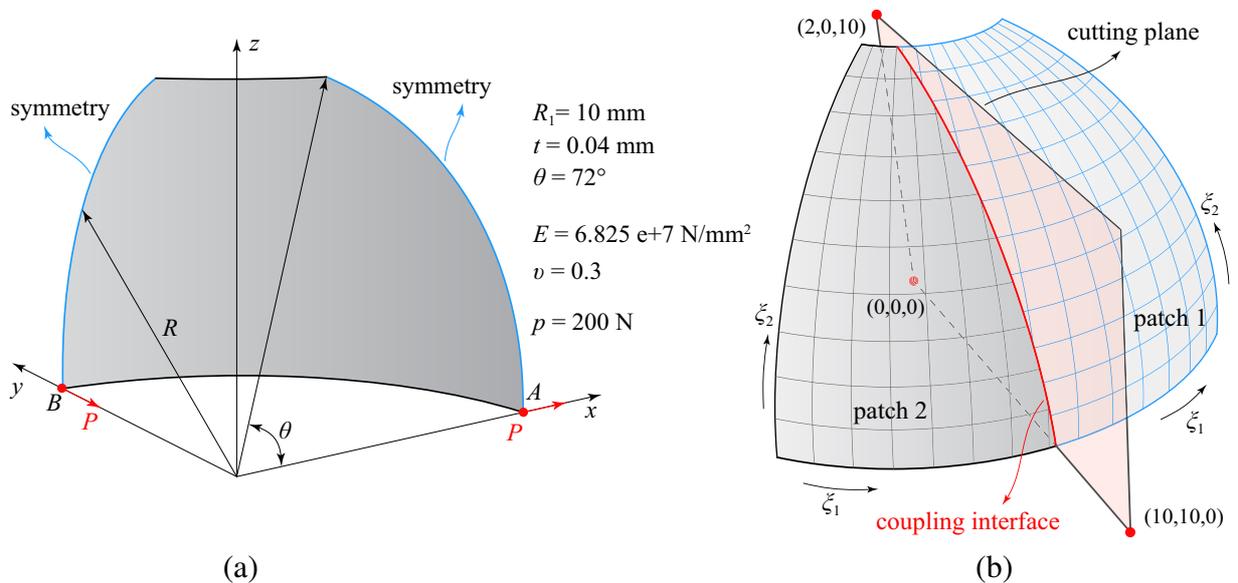


Figure 18: Hemispherical shell subjected to alternating radial forces: (a) model description, (b) trimmed two-patch configuration and NURBS meshes.

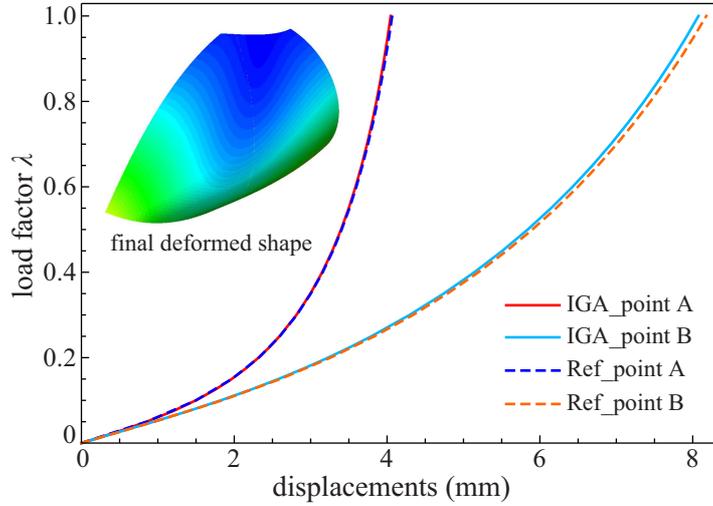


Figure 19: Load-displacement curves of the hemispherical shell.

5.2. Hemispherical shell subjected to alternating radial forces

As the second example, we consider a hemispherical shell with a circular hole on top [64]. The geometry and material properties are illustrated in Fig. 18a. Due to symmetry, only a quarter of the shell is modeled and symmetry boundary conditions are applied on the two corresponding boundary curves. The upper and lower boundaries are unconstrained. The shell is subjected to two alternating radial point forces P at the lower corners. As shown in Fig. 18b, the geometry is represented by two trimmed patches that are discretized with 11×12 and 8×10 quartic NURBS elements and connected along a trimming curve defined by the intersections between the cutting plane and the hemispherical shell. The load increments are $\lambda = 0.025$ for the first ten steps, and $\lambda = 0.05$ for the remaining 15 steps.

Figure 19 plots the displacements at point A along the x -axis and at point B along the y -axis versus the load factor λ . For both points, the trimmed shell analysis results show excellent agree-

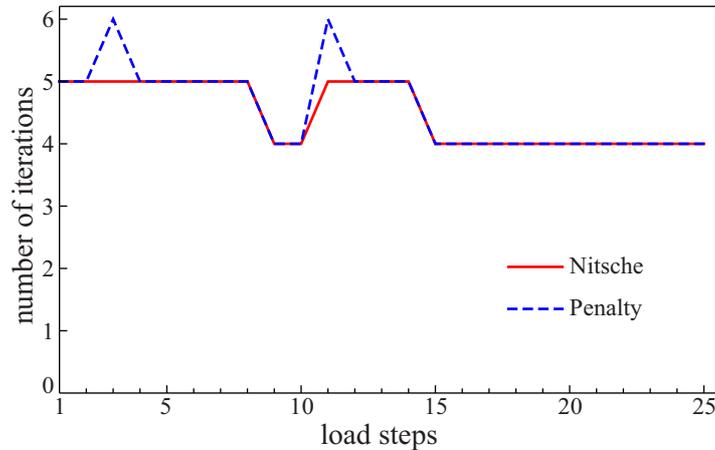


Figure 20: Hemispherical shell: number of iterations for the Nitsche-type formulation and the penalty method.

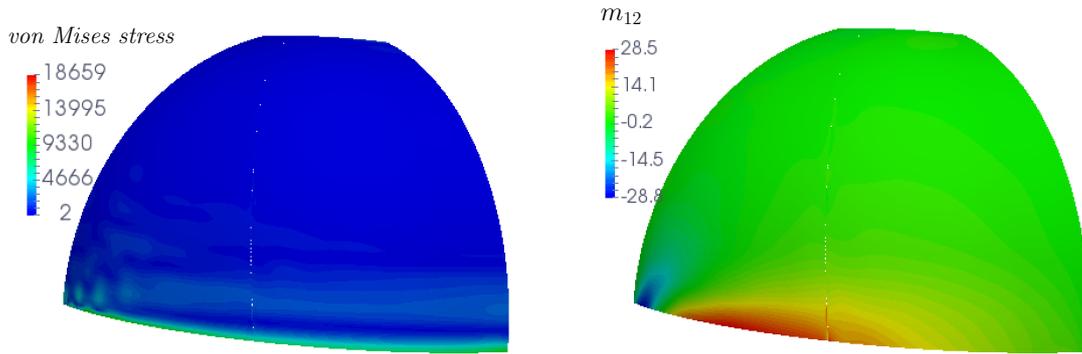


Figure 21: Hemispherical shell: von Mises stress and twisting moment.

ment with the reference solutions from [62]. Figure 20 demonstrates again that the geometrically nonlinear Nitsche-type formulation requires the same number or fewer iterations than the penalty method. We note that the former employs element-wise estimates of stabilization parameters and the latter uses a penalty factor α as detailed in the previous example. In Fig. 21, we observe that the von Mises stresses along the trimming curve show no oscillations or large jumps, even at the presence of the sharp boundary layer at the lower end of the shell.

5.3. Hinged cylindrical roof subjected to a pinching force

As the third example, we consider the hinged cylindrical roof subjected to a central pinching force P [65–68]. The model setup is illustrated in Fig. 22a, where only a quarter of the roof is modeled. We remark that no bifurcation points exist on the equilibrium path which preserves the

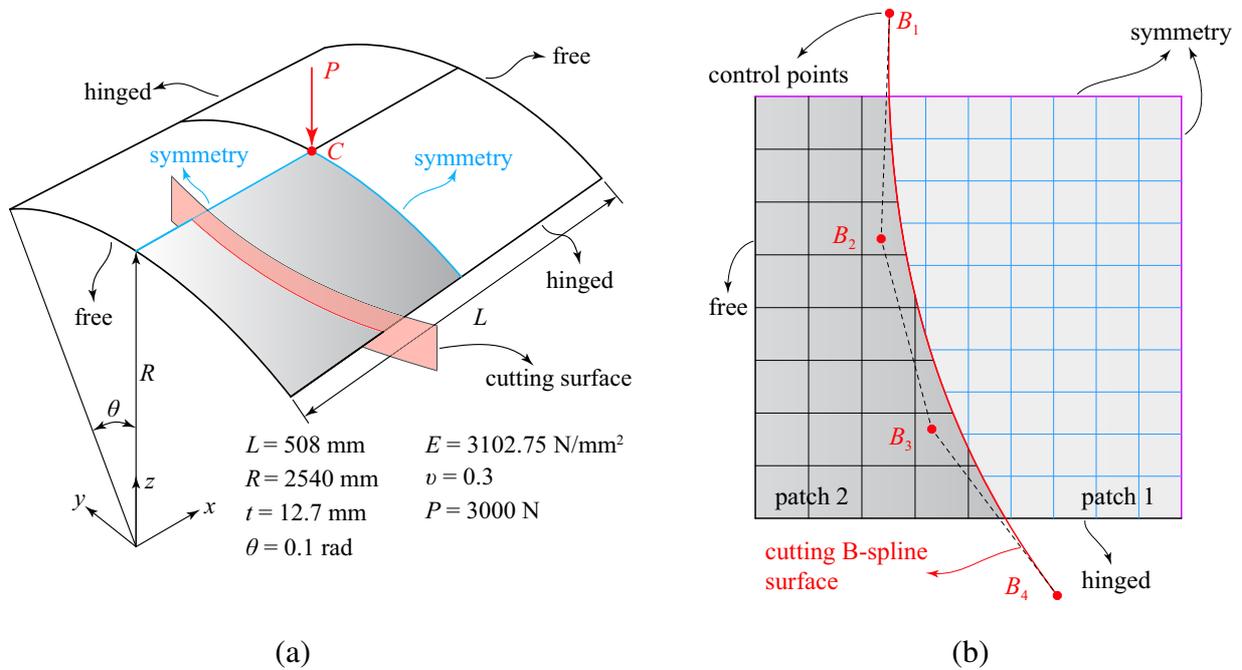


Figure 22: Hinged cylindrical roof: (a) model description, (b) trimmed two-patch configuration and NURBS meshes.

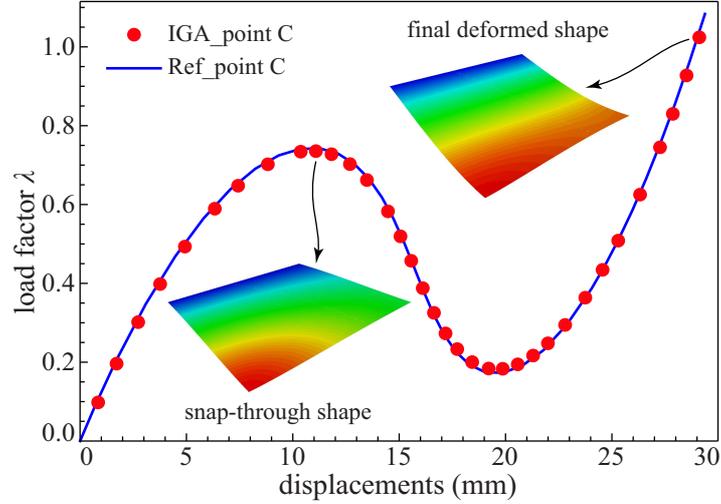


Figure 23: Load-displacement curves for the hinged cylindrical roof.

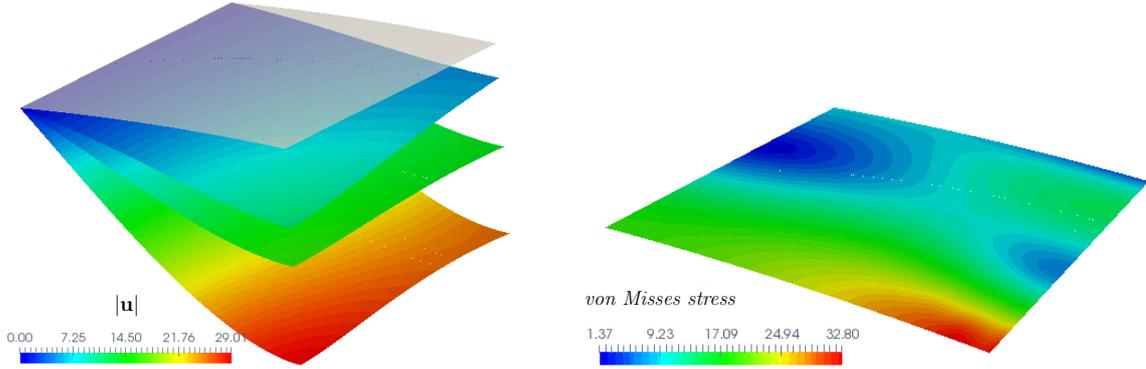


Figure 24: Left: undeformed (gray), before & after snap-through and at maximum loading (scaling factor: 6). Right: von Mises stress distribution of the mid-surface at the final configuration.

symmetry properties of the model [67]. The geometric surface is represented by two NURBS patches, discretized with 10×10 and 8×8 cubic NURBS elements as illustrated in Fig. 22b. The trimming surface is the extrusion of a B-spline curve along the z -axis, where the cubic B-spline curve is defined on the $(x, y, 0)$ plane and control points $B_1 = (80, 50, 0)$, $B_2 = (75, -85, 0)$, $B_3 = (105, -200, 0)$ and $B_4 = (180, -300, 0)$. To avoid snap-through convergence issues, we use an arc-length path-following technique [69–71].

Figure 23 plots the z -displacement at point C versus the load factor λ . We observe excellent agreement between the trimmed shell analysis results and the reference solution from [62]. Figure 24 plots the total displacements on the deformed configuration at various stages of the geometrically nonlinear computation and the von Mises stress distribution of the shell mid-surface in the final configuration. Displacements and stresses are smooth along the trimming curve and show a quality fully comparable to a single-patch solution.

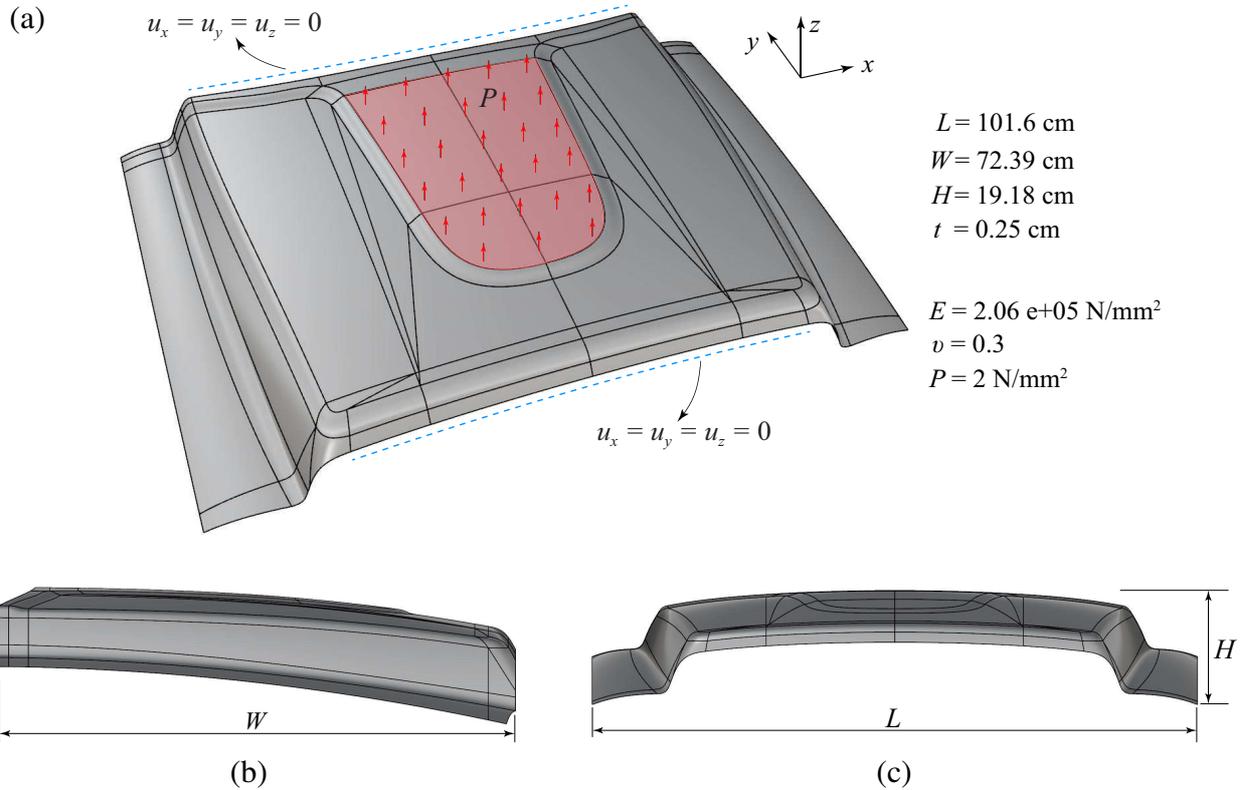


Figure 25: Hood of a RAM truck: (a) trimmed 76-patch surface, boundary conditions, (b) left view, (c) front view.

5.4. Large deformation trimmed shell analysis of a Dodge RAM hood

As the last example, we consider the hood of a Dodge RAM truck whose geometric surface model is generated by the CAD software tool *Faircad3D*¹. This numerical example will illustrate the effectiveness and accuracy of isogeometric trimmed shell analysis for an industry-scale geometry. Figure 25 illustrates the geometric model that is exported from *Faircad3D* via the STEP exchange format. It consists of a trimmed NURBS surface with 76 patches and 3,896 cubic NURBS elements, 7,832 control points, and 2,259 individual trimming segments. It also provides details on the overall dimensions and the loading and support conditions that we will use for isogeometric trimmed shell analysis. We derive an isogeometric trimmed shell discretization directly from the trimmed NURBS surface, without geometry clean-up or other preprocessing steps, leading to a NURBS mesh with 23,496 degrees of freedom, cf. Fig. 26.

To put the advantages of isogeometric shell analysis in perspective, we also set up a standard finite element analysis in the commercial CAE package *Abaqus*² [72]. To obtain an analysis-suitable mesh in *Abaqus*, however, we have to perform a number of geometry clean-up procedures first. Figure 27 shows the first output of the internal mesh generator after we loaded the STEP file into *Abaqus*. Each red dot corresponds to a location in the “dirty” original geometric model,

¹<http://www.faircad3d.com>

²<https://www.3ds.com/products-services/simulia/products/abaqus/>

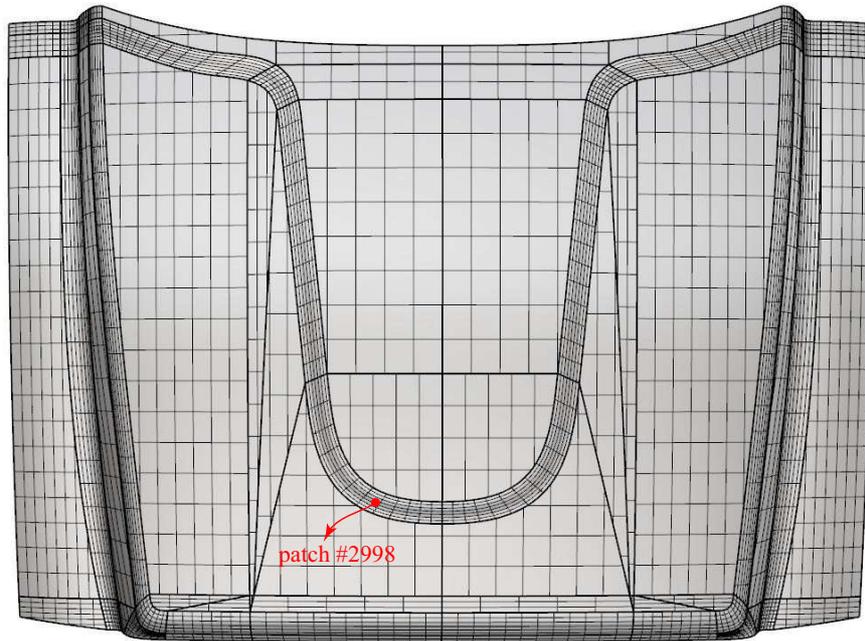


Figure 26: Isogeometric mesh of the RAM hood (76 patches, 3,896 cubic NURBS elements, 2,259 trimming segments).

where a geometry-related problem was detected that could potentially prevent successful meshing. We found that most of the necessary geometry clean-up steps could be directly performed within Abaqus via its geometry clean-up capabilities. A typical remedy applied by Abaqus is the

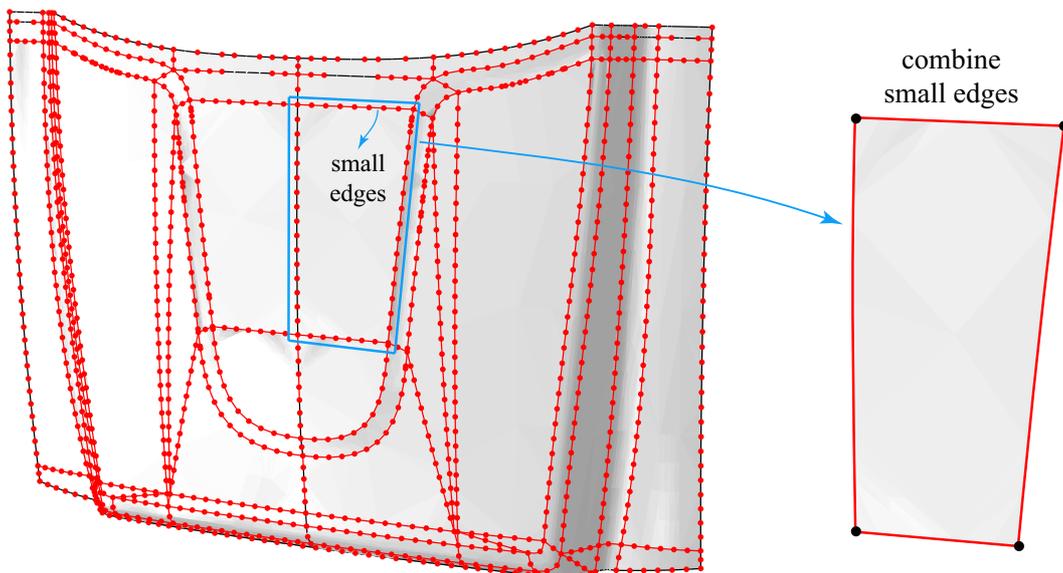


Figure 27: Imprecise geometry and corresponding geometry repair in Abaqus: series of small edges in the original trimmed model need to be merged into larger edges before a mesh can be created.

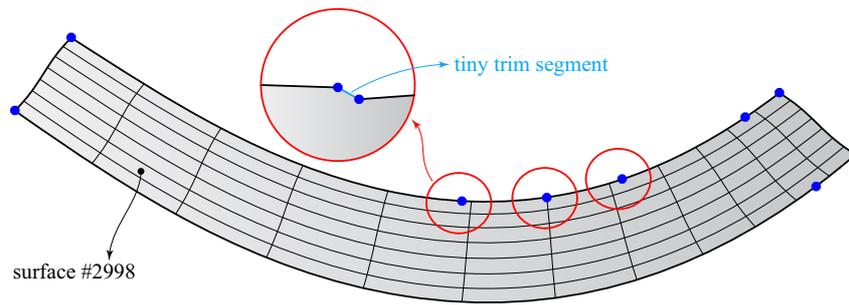


Figure 28: Geometric inaccuracies in the form of tiny trimming segments.

combination of small segments of the same edge as illustrated in Fig. 27. Some of the geometric inaccuracies, however, could not be resolved, which required manual intervention on our part. Figure 28 illustrates an example, which corresponds to the trimmed patch #2998 shown in Fig. 26. The upper and lower patch boundaries are represented by several small segments, each being defined as a trimming curve. We observe that there exist several tiny trimming/non-trimming curve segments, which obviously cause significant complications for standard mesh algorithms. We emphasize that this feature does not pose a detrimental roadblock for isogeometric trimmed shell analysis, where they are far below the mesh resolution and therefore do not have an adverse effect.

The “clean” geometric model can then be used for mesh generation in Abaqus. The resulting mesh is shown in Fig. 29. It consists of 7,464 four-node quadrilateral and three-node triangular Reissner-Mindlin shell elements with 45,546 degrees of freedom. The element type adopted in this example is S4R.

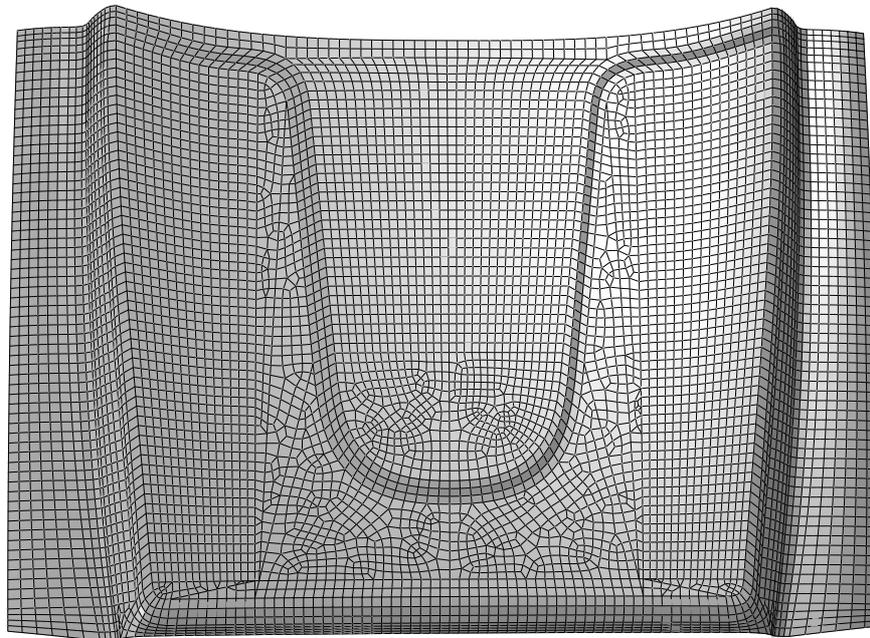


Figure 29: The meshes of the RAM hood model in ABAQUS.

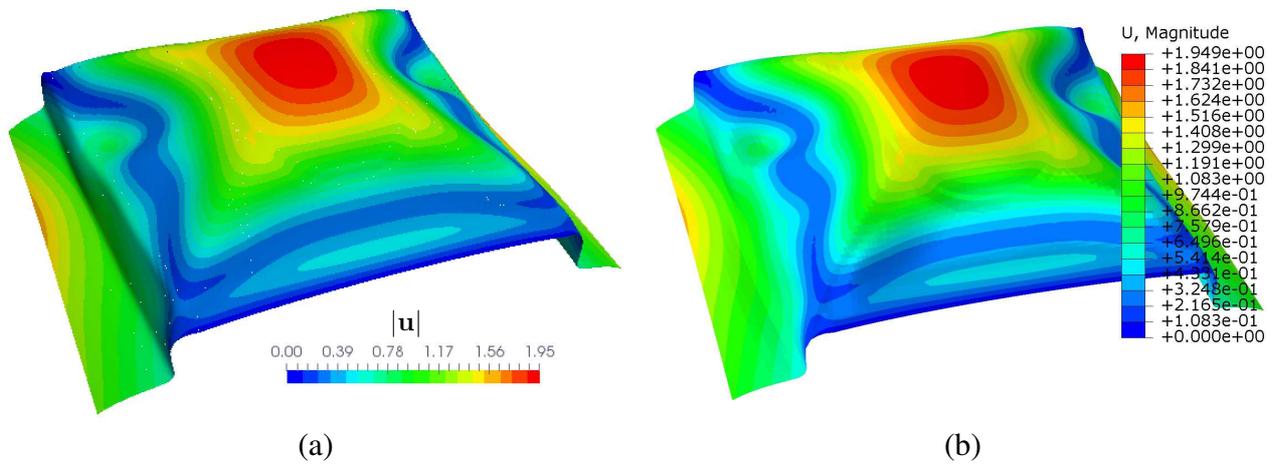


Figure 30: Total displacement of the RAM hood: (a) isogeometric analysis, (b) Abaqus.

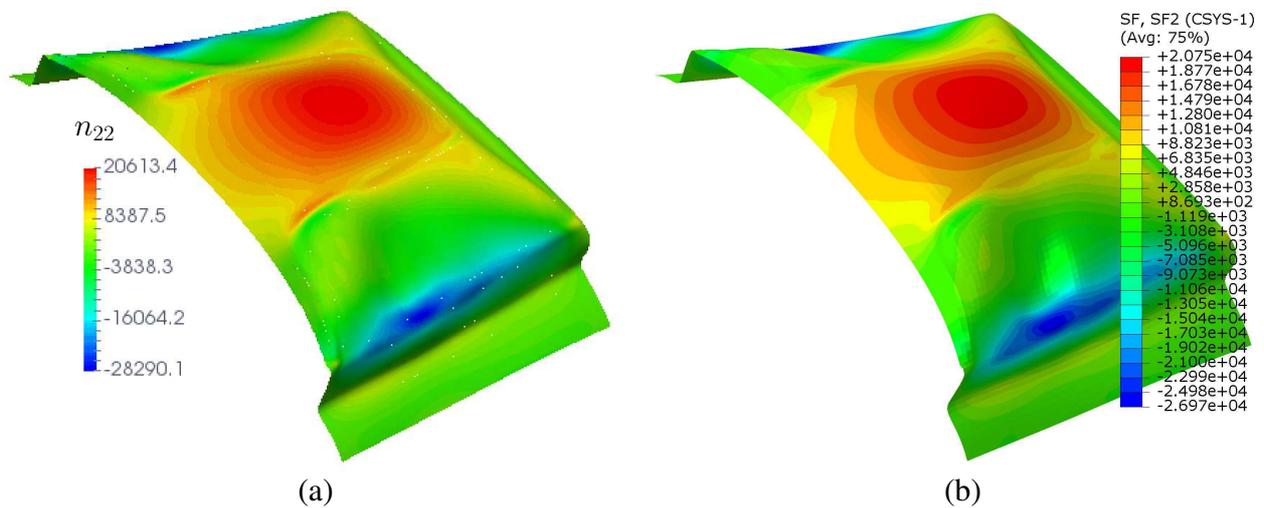


Figure 31: The stress resultant n_{22} of the RAM model: (a) isogeometric analysis, (b) Abaqus.

Figures 30, 31 and 32 plot the total displacements, the stress resultant n_{22} and the bending moment m_{22} on the deformed configuration. We note that the latter two components are evaluated with respect to global coordinates to ease the comparison between the two discretization methods. The displayed solution fields were obtained with isogeometric trimmed shell analysis on the NURBS discretization shown in Fig. 26 and Abaqus standard shell finite elements (S4R) on the nodal mesh shown in Fig. 29. We observe that both the displacement response as well as the stress and moment response confirms that isogeometric trimmed shell analysis yields results that provide the same overall accuracy as the corresponding Abaqus results. We emphasize again that the decisive advantage of isogeometric trimmed shell analysis is not to gain accuracy over standard finite elements, but to mitigate geometry clean-up and mesh healing procedures. However, as has been noted in numerous examples [9, 73, 74], isogeometric analysis in virtually every case yields results much superior to traditional FEA.

In geometric models based on trimmed CAD surfaces, trimming curves are of critical impor-

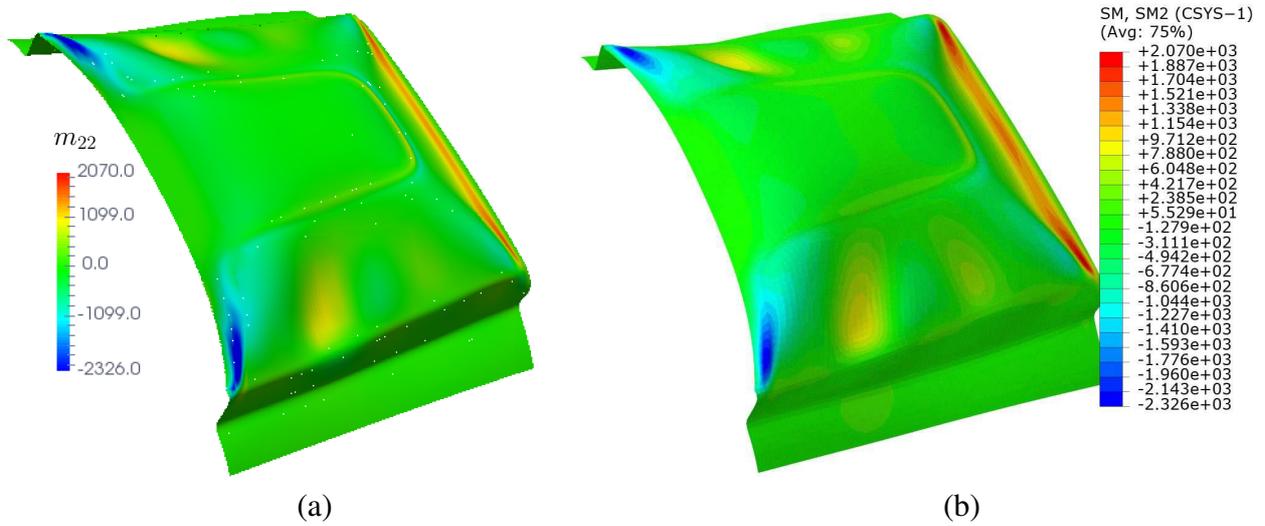


Figure 32: The bending moment m_{22} of the RAM model: (a) isogeometric analysis, (b) Abaqus.

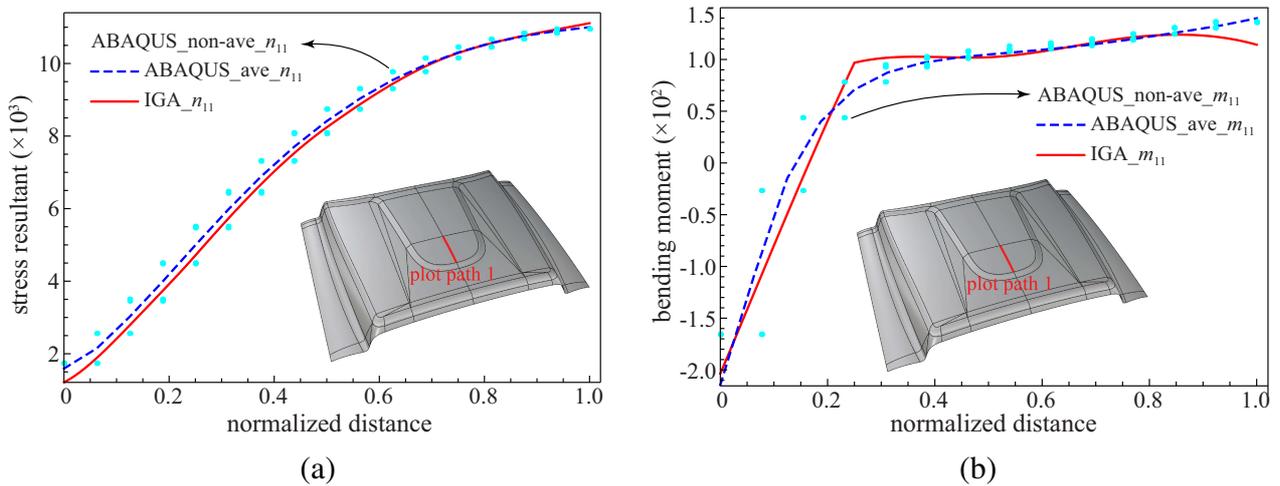


Figure 33: Derivative quantities along an untrimmed interface: (a) stress resultant n_{11} , (b) bending moment m_{11} .

tance from an engineering design point of view, because they typically coincide with regions, where high stress concentration or connections such as welding or gluing occur. Therefore, the fidelity of the isogeometric solution at and near trimming curves is of significant engineering interest. To gain a first impression of the quality of the stress solution at curves, where interface constraints are enforced in a weak sense, we plot stress resultants and bending moments along two selected interface curves. Figure 33 compares the results of isogeometric trimmed shell analysis and standard shell finite elements for an interface that conforms to NURBS element edges in the isogeometric discretization. In other words, this interface involves the geometrically nonlinear Nitsche method, but does not involve trimming. We observe that derivative quantities obtained with trimmed shell analysis are in good accordance with the Abaqus results. We note that the Abaqus non-averaged results shown in Figs. 33 and 34 are obtained from the nodal values extrapolated from the Gauss points of the element. Figure 34 compares equivalent results for a true

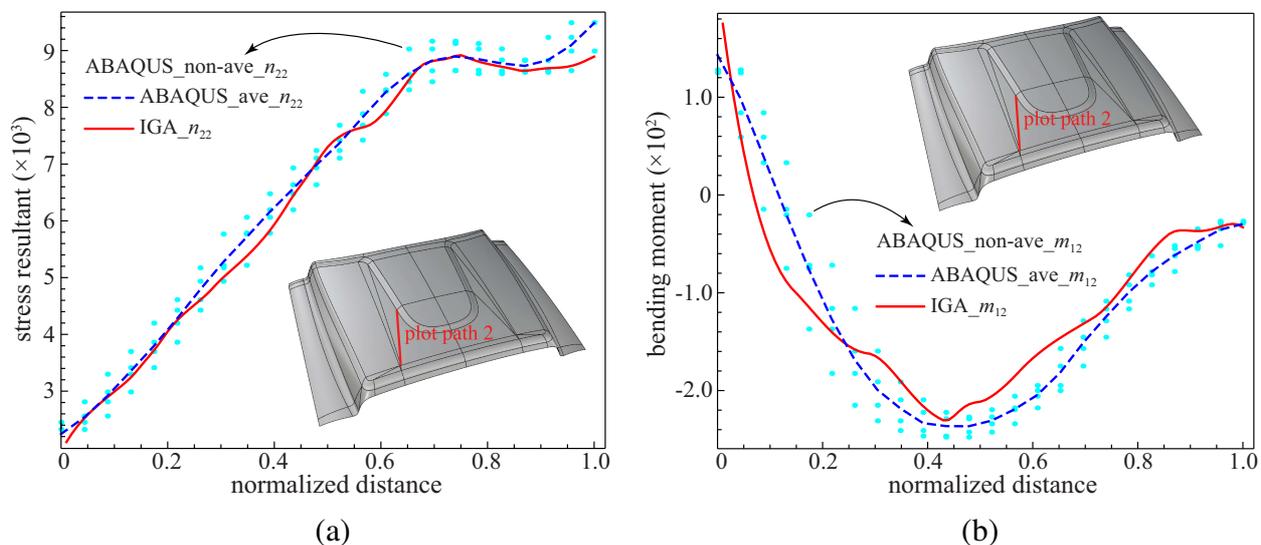


Figure 34: Derivative quantities along a trimming curve: (a) stress resultant n_{22} , (b) bending moment m_{12} .

trimming curve, which does not conform to NURBS element edges in the CAD surface. We observe that in the trimmed case, derivative quantities obtained with trimmed shell analysis are still in good accordance with the Abaqus results, although not to the same extent as for an untrimmed interface. When interpreting these results, we recall that the trimmed isogeometric shell results are computed based on a rotation-free Kirchhoff-Love formulation on a relatively coarse mesh, while the Abaqus results are obtained on a finer mesh with a Reissner-Mindlin formulation and postprocessed by projecting stress/moment results at Gauss points onto a C^0 -continuous basis.

6. Summary and conclusions

In this paper, we generalized our prior work on variationally consistent Nitsche-type methods for the weak imposition of coupling constraints to geometrically nonlinear Kirchhoff-Love shells, including the element-wise estimation of stabilization parameters. We combined the resulting formulation with a nonlinear rotation-free Kirchhoff-Love shell element and the blending function method that re-parametrizes trimmed surface elements to find geometrically accurate quadrature rules. In addition, we discussed a set of algorithms that enable the automatic interaction with CAD data structures based on the STEP exchange format, focusing on the classification of elements as full, cut and void, and the generation of quadrature points for trimmed elements and along surface-to-surface intersection curves. These efforts resulted in a fully operable framework for isogeometric trimmed shell analysis that, due to the generality of STEP for trimmed CAD surfaces, can operate immediately with any CAD system.

We illustrated the performance of isogeometric trimmed shell analysis for a series of numerical examples that involve large-deformation analysis. The examples included a slit annular plate, a hemispherical shell, and a hinged cylindrical roof. Our results confirm the accuracy of isogeometric shell analysis in comparison to benchmark results available in the literature, both in terms of the

overall nonlinear load-displacement behavior as well as in terms of local stress accuracy near and directly at trimming curves. We also demonstrated the algorithmic robustness of the variationally consistent constraint formulation. In particular, it required a slightly smaller number of Newton iterations to converge than the penalty method. To demonstrate the effectiveness of isogeometric trimmed shell analysis in a practically relevant engineering design scenario, we considered a full-scale hood of a Dodge RAM truck, parametrized by a 76-patch trimmed CAD surface that was generated by the CAD software Faircad3d. We showed that the exported STEP file could be directly analyzed with our framework, while the leading commercial software Abaqus required a number of geometry clean-up operations before successful mesh generation. We demonstrated that trimmed shell analysis based on STEP import and the geometrically nonlinear Nitsche-type formulation achieved the same overall solution accuracy in both displacements and stresses/moments as standard shell finite elements in Abaqus. We also showed that the local accuracy of stress resultants and bending moments directly at trimming curves are in good accuracy with respect to corresponding Abaqus results.

In summary, we think that isogeometric trimmed shell analysis has made significant progress towards an industry-relevant design-through-analysis technology within the last five years. Although its fundamental principles seem to be clear and fairly well-established at this point, one can identify a number of technical details that still need to be worked out. In the context of geometrically nonlinear analysis, an example is a mechanism to control the re-estimation of stabilization parameters during the Newton-Raphson algorithm due to the change in configuration, for example based on the rate of change in element-wise strain energy. Other open questions include the proof of concept of isogeometric trimmed shells in special analysis situations such as explicit dynamics. A very important aspect that is still unclear from a fundamental point of view is how to guarantee accurate stress resultants and bending moments directly at trimming curves. This is critical for the applicability in engineering design, since trimming curves typically coincide with regions of stress concentration. We therefore anticipate that the long-term success of isogeometric trimmed shell analysis in the form presented in this paper will depend on developing effective technology that can guarantee accurate stress and moment solutions at trimming curves.

Acknowledgments: D. Schillinger gratefully acknowledges support from the National Science Foundation through the NSF CAREER Award No. 1651577. Y. Guo would like to thank the National Natural Science Foundation of China (Grant no. 11602106) and Natural Science Foundation of Jiangsu Province of China (Grant no. BK20160783) for their support. Thomas J. R. Hughes was partially supported by the Office of Naval Research (Grant Nos. N00014-17-1-2119 and N00014-13-1-0500), and by the Army Research Office (Grant No. W911NF-13-1-0220). This support is gratefully acknowledged. The authors also acknowledge the Minnesota Supercomputing Institute (MSI) of the University of Minnesota for providing computing resources that have contributed to the research results reported within this paper (<https://www.msi.umn.edu/>).

Appendix A. Some relevant details from differential geometry and shell kinematics

In this appendix, we again resort to writing derivatives with respect to displacement variables in terms of the three displacements variables U_r at control point \mathbf{B}_r . For second derivatives, we introduce a second control point index s .

Appendix A.1. Derivatives of the surface's contravariant metric coefficients

The surface's covariant and contravariant metric coefficients have the following relation:

$$a^{\alpha\beta} a_{\beta\gamma} = \delta_\gamma^\alpha \quad (\text{A.1})$$

The derivatives of (A.1) with respect to the individual displacement variables U_r can be represented as:

$$a_{,\mathbf{U}_r}^{\alpha\beta} a_{\beta\gamma} + a^{\alpha\beta} a_{\beta\gamma,\mathbf{U}_r} = 0 \quad (\text{A.2})$$

from which we get the first derivatives of the contravariant coefficients with respect to the displacement variables:

$$a_{,\mathbf{U}_r}^{\alpha\beta} = -a^{\alpha\gamma} a_{\gamma\delta,\mathbf{U}_r} a^{\delta\beta} \quad (\text{A.3})$$

where

$$a_{\gamma\delta,\mathbf{U}_r} = \mathbf{a}_{\gamma,\mathbf{U}_r} \cdot \mathbf{a}_\delta + \mathbf{a}_\gamma \cdot \mathbf{a}_{\delta,\mathbf{U}_r} \quad (\text{A.4})$$

The second derivatives of the contravariant metric coefficients with respect to the displacement variables are:

$$\begin{aligned} a_{,\mathbf{U}_r\mathbf{U}_s}^{\alpha\beta} &= a^{\alpha\mu} a_{\mu\nu,\mathbf{U}_s} a^{\nu\gamma} a_{\gamma\delta,\mathbf{U}_r} a^{\delta\beta} - a^{\alpha\gamma} a_{\gamma\delta,\mathbf{U}_r\mathbf{U}_s} a^{\delta\beta} \\ &\quad + a^{\alpha\gamma} a_{\gamma\delta,\mathbf{U}_r} a^{\delta\mu} a_{\mu\nu,\mathbf{U}_s} a^{\nu\beta} \end{aligned} \quad (\text{A.5})$$

in which

$$a_{\gamma\delta,\mathbf{U}_r\mathbf{U}_s} = \mathbf{a}_{\gamma,\mathbf{U}_r\mathbf{U}_s} \cdot \mathbf{a}_\delta + \mathbf{a}_{\gamma,\mathbf{U}_r} \cdot \mathbf{a}_{\delta,\mathbf{U}_s} + \mathbf{a}_{\gamma,\mathbf{U}_s} \cdot \mathbf{a}_{\delta,\mathbf{U}_r} + \mathbf{a}_\gamma \cdot \mathbf{a}_{\delta,\mathbf{U}_r\mathbf{U}_s} \quad (\text{A.6})$$

Appendix A.2. Derivatives of the normal vector

The first derivatives of the normal vector with respect to the displacement variables U_r is written as:

$$\mathbf{a}_{3,\mathbf{U}_r} = \frac{\hat{\mathbf{a}}_{3,\mathbf{U}_r}}{\bar{\mathbf{a}}_3} - \frac{\hat{\mathbf{a}}_3 \cdot \bar{\mathbf{a}}_{3,\mathbf{U}_r}}{\bar{\mathbf{a}}_3^2} \quad (\text{A.7})$$

with:

$$\hat{\mathbf{a}}_3 = \mathbf{a}_1 \times \mathbf{a}_2 \quad (\text{A.8})$$

$$\hat{\mathbf{a}}_{3,\mathbf{U}_r} = \mathbf{a}_{1,\mathbf{U}_r} \times \mathbf{a}_2 + \mathbf{a}_1 \times \mathbf{a}_{2,\mathbf{U}_r} \quad (\text{A.9})$$

$$\bar{\mathbf{a}}_3 = |\mathbf{a}_1 \times \mathbf{a}_2| \quad (\text{A.10})$$

$$\bar{\mathbf{a}}_{3,\mathbf{U}_r} = \frac{\hat{\mathbf{a}}_3 \cdot \hat{\mathbf{a}}_{3,\mathbf{U}_r}}{\bar{\mathbf{a}}_3} \quad (\text{A.11})$$

The second derivatives of the normal vector with respect to the displacement variables can be formulated as:

$$\mathbf{a}_{3,U_r U_s} = \frac{\hat{\mathbf{a}}_{3,U_r U_s}}{\bar{\mathbf{a}}_3} - \frac{\hat{\mathbf{a}}_{3,U_r} \cdot \bar{\mathbf{a}}_{3,U_s}}{\bar{\mathbf{a}}_3^2} - \frac{\hat{\mathbf{a}}_{3,U_s} \cdot \bar{\mathbf{a}}_{3,U_r} + \hat{\mathbf{a}}_3 \cdot \bar{\mathbf{a}}_{3,U_r U_s}}{\bar{\mathbf{a}}_3^2} + \frac{2 \hat{\mathbf{a}}_3 \cdot \bar{\mathbf{a}}_{3,U_r} \bar{\mathbf{a}}_{3,U_s}}{\bar{\mathbf{a}}_3^3} \quad (\text{A.12})$$

where

$$\hat{\mathbf{a}}_{3,U_r U_s} = \mathbf{a}_{1,U_r} \times \mathbf{a}_{2,U_s} + \mathbf{a}_{1,U_s} \times \mathbf{a}_{2,U_r} \quad (\text{A.13})$$

$$\bar{\mathbf{a}}_{3,U_r U_s} = \frac{\hat{\mathbf{a}}_{3,U_s} \cdot \hat{\mathbf{a}}_{3,U_r} + \hat{\mathbf{a}}_3 \cdot \hat{\mathbf{a}}_{3,U_r U_s}}{\bar{\mathbf{a}}_3} - \frac{(\hat{\mathbf{a}}_3 \cdot \hat{\mathbf{a}}_{3,U_r}) \bar{\mathbf{a}}_{3,U_s}}{\bar{\mathbf{a}}_3^2} \quad (\text{A.14})$$

The derivatives of the normal vector \mathbf{a}_3 w.r.t the curvilinear coordinates ξ_α follow the same way as Eq. (A.7), and the derivatives of $\mathbf{a}_{3,\alpha,U_r}$ can also be obtained from the Eq. (A.12). For $\mathbf{a}_{3,\alpha,U_r U_s}$, it can be represented as:

$$\begin{aligned} \mathbf{a}_{3,\alpha,U_r U_s} &= \frac{\hat{\mathbf{a}}_{3,\alpha,U_r U_s}}{\bar{\mathbf{a}}_3} - \frac{\hat{\mathbf{a}}_{3,\alpha,U_r} \bar{\mathbf{a}}_{3,U_s}}{\bar{\mathbf{a}}_3^2} \\ &- \frac{\hat{\mathbf{a}}_{3,\alpha,U_s} \cdot \bar{\mathbf{a}}_{3,U_r} + \hat{\mathbf{a}}_{3,\alpha} \cdot \bar{\mathbf{a}}_{3,U_r U_s}}{\bar{\mathbf{a}}_3^2} + \frac{2 \hat{\mathbf{a}}_{3,\alpha} \cdot \bar{\mathbf{a}}_{3,U_r} \bar{\mathbf{a}}_{3,U_s}}{\bar{\mathbf{a}}_3^3} \\ &- \frac{\hat{\mathbf{a}}_{3,U_r U_s} \cdot \bar{\mathbf{a}}_{3,\alpha} + \hat{\mathbf{a}}_{3,U_r} \cdot \bar{\mathbf{a}}_{3,\alpha,U_s} + \hat{\mathbf{a}}_{3,U_s} \cdot \bar{\mathbf{a}}_{3,\alpha,U_r} + \hat{\mathbf{a}}_3 \cdot \bar{\mathbf{a}}_{3,\alpha,U_r U_s}}{\bar{\mathbf{a}}_3^2} \\ &+ \frac{2 (\hat{\mathbf{a}}_{3,U_r} \cdot \bar{\mathbf{a}}_{3,\alpha} + \hat{\mathbf{a}}_3 \cdot \bar{\mathbf{a}}_{3,\alpha,U_r}) \bar{\mathbf{a}}_{3,U_s}}{\bar{\mathbf{a}}_3^3} \\ &+ \frac{2 (\hat{\mathbf{a}}_{3,U_s} \cdot \bar{\mathbf{a}}_{3,\alpha} + \hat{\mathbf{a}}_3 \cdot \bar{\mathbf{a}}_{3,\alpha,U_s}) \bar{\mathbf{a}}_{3,U_r} + 2 \hat{\mathbf{a}}_3 \cdot \bar{\mathbf{a}}_{3,\alpha} \bar{\mathbf{a}}_{3,U_r U_s}}{\bar{\mathbf{a}}_3^3} \\ &- \frac{6 \hat{\mathbf{a}}_3 \cdot \bar{\mathbf{a}}_{3,\alpha} \bar{\mathbf{a}}_{3,U_r} \bar{\mathbf{a}}_{3,U_s}}{\bar{\mathbf{a}}_3^4} \end{aligned} \quad (\text{A.15})$$

where

$$\begin{aligned} \hat{\mathbf{a}}_{3,\alpha,U_r U_s} &= \mathbf{a}_{1,\alpha,U_r} \times \mathbf{a}_{2,U_s} + \mathbf{a}_{1,\alpha,U_s} \times \mathbf{a}_{2,U_r} \\ &+ \mathbf{a}_{1,U_r} \times \mathbf{a}_{2,\alpha,U_s} + \mathbf{a}_{1,U_s} \times \mathbf{a}_{2,\alpha,U_r} \end{aligned} \quad (\text{A.16})$$

and where

$$\begin{aligned} \bar{\mathbf{a}}_{3,\alpha,U_r U_s} &= \frac{\hat{\mathbf{a}}_{3,U_r U_s} \cdot \hat{\mathbf{a}}_{3,\alpha} + \hat{\mathbf{a}}_{3,U_r} \cdot \hat{\mathbf{a}}_{3,\alpha,U_s} + \hat{\mathbf{a}}_{3,U_s} \cdot \hat{\mathbf{a}}_{3,\alpha,U_r} + \hat{\mathbf{a}}_3 \cdot \hat{\mathbf{a}}_{3,\alpha,U_r U_s}}{\bar{\mathbf{a}}_3} \\ &- \frac{(\hat{\mathbf{a}}_{3,U_r} \cdot \hat{\mathbf{a}}_{3,\alpha} + \hat{\mathbf{a}}_3 \cdot \hat{\mathbf{a}}_{3,\alpha,U_r}) \bar{\mathbf{a}}_{3,U_s}}{\bar{\mathbf{a}}_3^2} \\ &- \frac{(\hat{\mathbf{a}}_{3,U_s} \cdot \hat{\mathbf{a}}_{3,\alpha} + \hat{\mathbf{a}}_3 \cdot \hat{\mathbf{a}}_{3,\alpha,U_s}) \bar{\mathbf{a}}_{3,U_r} + (\hat{\mathbf{a}}_3 \cdot \hat{\mathbf{a}}_{3,\alpha}) \bar{\mathbf{a}}_{3,U_r U_s}}{\bar{\mathbf{a}}_3^2} \\ &+ \frac{2 (\hat{\mathbf{a}}_3 \cdot \hat{\mathbf{a}}_{3,\alpha}) \bar{\mathbf{a}}_{3,U_r} \bar{\mathbf{a}}_{3,U_s}}{\bar{\mathbf{a}}_3^3} \end{aligned} \quad (\text{A.17})$$

Appendix A.3. Derivatives of the stress resultants and bending moments

According to the Eqs. (13) and (14), the derivatives of the stress resultants $N^{\alpha\beta}$ and bending moments $M^{\alpha\beta}$ can be obtained from the corresponding derivatives of the membrane strain $\varepsilon_{\alpha\beta}$ and bending strain $\kappa_{\alpha\beta}$, respectively.

The first and second derivatives of the membrane strain with respect to the displacement variables can be written as:

$$\varepsilon_{\alpha\beta, \mathbf{U}_r} = \frac{1}{2}(a_{\alpha\beta} - A_{\alpha\beta})_{, \mathbf{U}_r} = \frac{1}{2}(\mathbf{a}_{\alpha, \mathbf{U}_r} \cdot \mathbf{a}_\beta + \mathbf{a}_\alpha \cdot \mathbf{a}_{\beta, \mathbf{U}_r}) \quad (\text{A.18})$$

$$\varepsilon_{\alpha\beta, \mathbf{U}_r \mathbf{U}_s} = \frac{1}{2}(a_{\alpha\beta} - A_{\alpha\beta})_{, \mathbf{U}_r \mathbf{U}_s} = \frac{1}{2}(\mathbf{a}_{\alpha, \mathbf{U}_r} \cdot \mathbf{a}_{\beta, \mathbf{U}_s} + \mathbf{a}_{\alpha, \mathbf{U}_s} \cdot \mathbf{a}_{\beta, \mathbf{U}_r}) \quad (\text{A.19})$$

Similarly, the derivatives $\varepsilon_{\alpha\beta, \gamma, \mathbf{U}_r}$ and $\varepsilon_{\alpha\beta, \gamma, \mathbf{U}_r \mathbf{U}_s}$ in Eqs. (65)-(66) are represented, respectively, as:

$$\varepsilon_{\alpha\beta, \gamma, \mathbf{U}_r} = \frac{1}{2}(\mathbf{a}_{\alpha, \gamma, \mathbf{U}_r} \cdot \mathbf{a}_\beta + \mathbf{a}_{\alpha, \gamma} \cdot \mathbf{a}_{\beta, \mathbf{U}_r} + \mathbf{a}_{\alpha, \mathbf{U}_r} \cdot \mathbf{a}_{\beta, \gamma} + \mathbf{a}_\alpha \cdot \mathbf{a}_{\beta, \gamma, \mathbf{U}_r}) \quad (\text{A.20})$$

$$\varepsilon_{\alpha\beta, \gamma, \mathbf{U}_r \mathbf{U}_s} = \frac{1}{2}(\mathbf{a}_{\alpha, \gamma, \mathbf{U}_r} \cdot \mathbf{a}_{\beta, \mathbf{U}_s} + \mathbf{a}_{\alpha, \gamma, \mathbf{U}_s} \cdot \mathbf{a}_{\beta, \mathbf{U}_r} + \mathbf{a}_{\alpha, \mathbf{U}_r} \cdot \mathbf{a}_{\beta, \gamma, \mathbf{U}_s} + \mathbf{a}_{\alpha, \mathbf{U}_s} \cdot \mathbf{a}_{\beta, \gamma, \mathbf{U}_r}) \quad (\text{A.21})$$

The first and second derivatives of the bending strain with respect to the displacement variables are:

$$\kappa_{\alpha\beta, \mathbf{U}_r} = (B_{\alpha\beta} - b_{\alpha\beta})_{, \mathbf{U}_r} = -(\mathbf{a}_{\alpha, \beta, \mathbf{U}_r} \cdot \mathbf{a}_3 + \mathbf{a}_{\alpha, \beta} \cdot \mathbf{a}_{3, \mathbf{U}_r}) \quad (\text{A.22})$$

$$\begin{aligned} \kappa_{\alpha\beta, \mathbf{U}_r \mathbf{U}_s} &= (B_{\alpha\beta} - b_{\alpha\beta})_{, \mathbf{U}_r \mathbf{U}_s} \\ &= -(\mathbf{a}_{\alpha, \beta, \mathbf{U}_r} \cdot \mathbf{a}_{3, \mathbf{U}_s} + \mathbf{a}_{\alpha, \beta, \mathbf{U}_s} \cdot \mathbf{a}_{3, \mathbf{U}_r} + \mathbf{a}_{\alpha, \beta} \cdot \mathbf{a}_{3, \mathbf{U}_r \mathbf{U}_s}) \end{aligned} \quad (\text{A.23})$$

For the derivatives of $\kappa_{\alpha\beta, \gamma, \mathbf{U}_r}$ and $\kappa_{\alpha\beta, \gamma, \mathbf{U}_r \mathbf{U}_s}$ in Eqs. (63)-(64), they are written as:

$$\kappa_{\alpha\beta, \gamma, \mathbf{U}_r} = -(\mathbf{a}_{\alpha, \beta \gamma, \mathbf{U}_r} \cdot \mathbf{a}_3 + \mathbf{a}_{\alpha, \beta \gamma} \cdot \mathbf{a}_{3, \mathbf{U}_r} + \mathbf{a}_{\alpha, \beta, \mathbf{U}_r} \cdot \mathbf{a}_{3, \gamma} + \mathbf{a}_{\alpha, \beta} \cdot \mathbf{a}_{3, \gamma, \mathbf{U}_r}) \quad (\text{A.24})$$

$$\begin{aligned} \kappa_{\alpha\beta, \gamma, \mathbf{U}_r \mathbf{U}_s} &= -(\mathbf{a}_{\alpha, \beta \gamma, \mathbf{U}_r} \cdot \mathbf{a}_{3, \mathbf{U}_s} + \mathbf{a}_{\alpha, \beta \gamma, \mathbf{U}_s} \cdot \mathbf{a}_{3, \mathbf{U}_r} + \mathbf{a}_{\alpha, \beta \gamma} \cdot \mathbf{a}_{3, \mathbf{U}_r \mathbf{U}_s} \\ &\quad + \mathbf{a}_{\alpha, \beta, \mathbf{U}_r} \cdot \mathbf{a}_{3, \gamma, \mathbf{U}_s} + \mathbf{a}_{\alpha, \beta, \mathbf{U}_s} \cdot \mathbf{a}_{3, \gamma, \mathbf{U}_r} + \mathbf{a}_{\alpha, \beta} \cdot \mathbf{a}_{3, \gamma, \mathbf{U}_r \mathbf{U}_s}) \end{aligned} \quad (\text{A.25})$$

where the derivatives of the normal vector \mathbf{a}_3 with respect to the displacement and curvilinear coordinates are shown in Appendix A.2.

In order to apply the constitutive relations, the derivatives of the membrane strain and the bending strain are transferred to the local Cartesian coordinates first. Taking the term $\varepsilon_{\alpha\beta, \mathbf{U}_r}$ for example:

$$\bar{\varepsilon}_{\gamma\delta, \mathbf{U}_r} = \varepsilon_{\alpha\beta, \mathbf{U}_r} (\mathbf{E}_\gamma \cdot \mathbf{A}^\alpha) (\mathbf{A}^\beta \cdot \mathbf{E}_\delta) \quad (\text{A.26})$$

The multiplication of $\bar{\varepsilon}_{\gamma\delta, \mathbf{U}_r}$ with the constitutive matrix $\hat{\mathbf{C}}$ results in the derivative of the stress resultants in the local Cartesian coordinates, which is written as $\bar{N}^{\gamma\delta}$. Then, a second transformation of the resulting term from the local Cartesian base to the covariant base is performed:

$$N^{\alpha\beta}_{, \mathbf{U}_r} = \bar{N}^{\gamma\delta}_{, \mathbf{U}_r} (\mathbf{A}^\alpha \cdot \mathbf{E}_\gamma) (\mathbf{E}_\delta \cdot \mathbf{A}^\beta) \quad (\text{A.27})$$

The same procedure applies to the derivatives of the bending moments $M^{\alpha\beta}$.

Appendix B. STEP source code of the trimmed surface example

The STEP source file for the example in section 4 is constructed from the CAD software Rhinoceros [55] and exported with a setting of AP 214 Automotive Design CC2. The complete data of the file is given below:

```
ISO-10303-21;
HEADER;
/* Generated by software containing ST-Developer
 * from STEP Tools, Inc. (www.steptools.com)
 */
/* OPTION: using custom schema-name function */

FILE_DESCRIPTION(
/* description */ (''),
/* implementation_level */ ('2;1');

FILE_NAME(
/* name */ ('example'),
/* time_stamp */ ('2017-05-30T11:18:52+08:00'),
/* author */ (''),
/* organization */ (''),
/* preprocessor_version */ ('ST-DEVELOPER v15'),
/* originating_system */ (''),
/* authorisation */ (''));

FILE_SCHEMA (('AUTOMOTIVE_DESIGN'));
ENDSEC;

DATA;
#10=SHAPE_REPRESENTATION_RELATIONSHIP('', '#100', #13);
#11=PRESENTATION_LAYER_ASSIGNMENT('X2\98848BBE503C\X0\','',
(#12));
#12=SHELL_BASED_SURFACE_MODEL('shell_1', (#14));
#13=MANIFOLD_SURFACE_SHAPE_REPRESENTATION('shell_rep_0',
(#12, #102), #99);
#14=OPEN_SHELL('', (#15, #16));
#15=ADVANCED_FACE('', (#17), #80, .T.);
#16=ADVANCED_FACE('', (#18), #81, .T.);
#17=FACE_OUTER_BOUND('', #19, .T.);
#18=FACE_OUTER_BOUND('', #20, .T.);
#19=EDGE_LOOP('', (#21, #22, #23, #24));
#20=EDGE_LOOP('', (#25, #26, #27, #28));
#21=ORIENTED_EDGE('', *, *, #52, .T.);
#22=ORIENTED_EDGE('', *, *, #53, .T.);
#23=ORIENTED_EDGE('', *, *, #54, .T.);
#24=ORIENTED_EDGE('', *, *, #55, .T.);
#25=ORIENTED_EDGE('', *, *, #55, .F.);
#26=ORIENTED_EDGE('', *, *, #56, .T.);
#27=ORIENTED_EDGE('', *, *, #57, .T.);
#28=ORIENTED_EDGE('', *, *, #58, .T.);
#29=PCURVE('', #80, #37);
#30=PCURVE('', #80, #38);
#31=PCURVE('', #80, #39);
#32=PCURVE('', #80, #40);
#33=PCURVE('', #81, #41);
#34=PCURVE('', #81, #42);
#35=PCURVE('', #81, #43);
#36=PCURVE('', #81, #44);
#37=DEFINITIONAL_REPRESENTATION('', (#60), #232);
#38=DEFINITIONAL_REPRESENTATION('', (#62), #232);
#39=DEFINITIONAL_REPRESENTATION('', (#64), #232);
#40=DEFINITIONAL_REPRESENTATION('', (#66), #232);
#41=DEFINITIONAL_REPRESENTATION('', (#67), #232);
#42=DEFINITIONAL_REPRESENTATION('', (#69), #232);
#43=DEFINITIONAL_REPRESENTATION('', (#71), #232);
#44=DEFINITIONAL_REPRESENTATION('', (#73), #232);
#45=SURFACE_CURVE('', #59, (#29), .PCURVE_S1.);
#46=SURFACE_CURVE('', #61, (#30), .PCURVE_S1.);
#47=SURFACE_CURVE('', #63, (#31), .PCURVE_S1.);
#48=SURFACE_CURVE('', #65, (#32, #33), .PCURVE_S1.);
#49=SURFACE_CURVE('', #68, (#34), .PCURVE_S1.);
#50=SURFACE_CURVE('', #70, (#35), .PCURVE_S1.);
#51=SURFACE_CURVE('', #72, (#36), .PCURVE_S1.);
#52=EDGE_CURVE('', #76, #74, #45, .T.);
#53=EDGE_CURVE('', #74, #75, #46, .T.);
#54=EDGE_CURVE('', #75, #77, #47, .T.);
#55=EDGE_CURVE('', #77, #76, #48, .T.);
#56=EDGE_CURVE('', #77, #78, #49, .T.);
#57=EDGE_CURVE('', #78, #79, #50, .T.);
#58=EDGE_CURVE('', #79, #76, #51, .T.);

#59=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#169, #170, #171, #172, #173),
.UNSPECIFIED., .F., .F., (4, 1, 4), (2.71428571428571, 3.5, 7.)),
.UNSPECIFIED.);
#60=B_SPLINE_CURVE_WITH_KNOTS('', 1, (#174, #175), .UNSPECIFIED.,
.F., .F., (2, 2), (2.71428571428571, 7.)), .UNSPECIFIED.);
#61=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#176, #177, #178, #179, #180,
#181), .UNSPECIFIED., .F., .F., (4, 1, 1, 4), (-10.,
-6.66666666666667, -3.33333333333333, 0.)), .UNSPECIFIED.);
#62=B_SPLINE_CURVE_WITH_KNOTS('', 1, (#182, #183), .UNSPECIFIED.,
.F., .F., (2, 2), (-10., 0.)), .UNSPECIFIED.);
#63=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#184, #185, #186, #187, #188),
.UNSPECIFIED., .F., .F., (4, 1, 4), (-7., -3.5, -1.28571428571428),
.UNSPECIFIED.);
#64=B_SPLINE_CURVE_WITH_KNOTS('', 1, (#189, #190), .UNSPECIFIED.,
.F., .F., (2, 2), (-7., -1.28571428571428), .UNSPECIFIED.);
#65=B_SPLINE_CURVE_WITH_KNOTS('', 1, (#191, #192), .UNSPECIFIED.,
.F., .F., (2, 2), (2.02030508910442, 12.1218305346265),
.UNSPECIFIED.);
#66=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#193, #194, #195, #196, #197,
#198, #199, #200, #201, #202), .UNSPECIFIED., .F., .F., (4, 3, 3, 4),
(2.02030508910442, 5.38748023761179, 8.75465538611916,
12.1218305346265)), .UNSPECIFIED.);
#67=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#203, #204, #205, #206, #207,
#208, #209), .UNSPECIFIED., .F., .F., (4, 3, 4), (2.02030508910442,
7.07106901606129, 12.1218305346265)), .UNSPECIFIED.);
#68=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#210, #211, #212, #213, #214),
.UNSPECIFIED., .F., .F., (4, 1, 4), (-4.28571428571428, -3.5, 0.)),
.UNSPECIFIED.);
#69=B_SPLINE_CURVE_WITH_KNOTS('', 1, (#215, #216), .UNSPECIFIED.,
.F., .F., (2, 2), (-4.28571428571428, 0.)), .UNSPECIFIED.);
#70=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#217, #218, #219, #220, #221),
.UNSPECIFIED., .F., .F., (4, 1, 4), (0., 5., 10.)), .UNSPECIFIED.);
#71=B_SPLINE_CURVE_WITH_KNOTS('', 1, (#222, #223), .UNSPECIFIED.,
.F., .F., (2, 2), (0., 10.)), .UNSPECIFIED.);
#72=B_SPLINE_CURVE_WITH_KNOTS('', 3, (#224, #225, #226, #227, #228),
.UNSPECIFIED., .F., .F., (4, 1, 4), (0., 3.5, 5.71428571428571)),
.UNSPECIFIED.);
#73=B_SPLINE_CURVE_WITH_KNOTS('', 1, (#229, #230), .UNSPECIFIED.,
.F., .F., (2, 2), (0., 5.71428571428571), .UNSPECIFIED.);
#74=VERTEX_POINT('', #163);
#75=VERTEX_POINT('', #164);
#76=VERTEX_POINT('', #165);
#77=VERTEX_POINT('', #166);
#78=VERTEX_POINT('', #167);
#79=VERTEX_POINT('', #168);
#80=B_SPLINE_SURFACE_WITH_KNOTS('', 3, 3, ((#108, #109, #110, #111,
#112), (#113, #114, #115, #116, #117), (#118, #119, #120, #121, #122),
(#123, #124, #125, #126, #127), (#128, #129, #130, #131, #132), (#133,
#134, #135, #136, #137))), .UNSPECIFIED., .F., .F., .F., (4, 1, 1, 4),
(4, 1, 4), (0., 3.33333333333333, 6.66666666666667, 10.)), (0., 3.5,
7.)), .UNSPECIFIED.);
#81=B_SPLINE_SURFACE_WITH_KNOTS('', 3, 3, ((#138, #139, #140, #141,
#142), (#143, #144, #145, #146, #147), (#148, #149, #150, #151, #152),
(#153, #154, #155, #156, #157), (#158, #159, #160, #161, #162))),
.UNSPECIFIED., .F., .F., .F., (4, 1, 4), (4, 1, 4), (0., 5., 10.)), (0.,
3.5, 7.)), .UNSPECIFIED.);
#82=SHAPE_DEFINITION_REPRESENTATION(#83, #100);
#83=PRODUCT_DEFINITION_SHAPE('Document', '', #85);
#84=PRODUCT_DEFINITION_CONTEXT('3D Mechanical Parts', #89,
'design');
#85=PRODUCT_DEFINITION('A', 'First version', #86, #84);
#86=PRODUCT_DEFINITION_FORMATION_WITH_SPECIFIED_SOURCE('A',
'First version', #91, .MADE.);
#87=PRODUCT_RELATED_PRODUCT_CATEGORY('tool', 'tool', (#91));
#88=APPLICATION_PROTOCOL_DEFINITION('Draft International
Standard', 'automotive_design', 1999, #89);
#89=APPLICATION_CONTEXT(
'data for automotive mechanical design processes');
#90=PRODUCT_CONTEXT('3D Mechanical Parts', #89, 'mechanical');
#91=PRODUCT('Document', 'Document',
'Rhino converted to STEP', (#90));
#92=(
LENGTH_UNIT()
NAMED_UNIT(*)
SI_UNIT(.MILLI., .METRE.)
);
```

```

#93=(
  NAMED_UNIT(*)
  PLANE_ANGLE_UNIT()
  SI_UNIT($, .RADIAN.)
);
#94=DIMENSIONAL_EXPONENTS(0.,0.,0.,0.,0.,0.,0.);
#95=PLANE_ANGLE_MEASURE_WITH_UNIT
(PLANE_ANGLE_MEASURE(0.01745329252),#93);
#96=(
  CONVERSION_BASED_UNIT('DEGREES',#95)
  NAMED_UNIT(#94)
  PLANE_ANGLE_UNIT()
);
#97=(
  NAMED_UNIT(*)
  SI_UNIT($, .STERADIAN.)
  SOLID_ANGLE_UNIT()
);
#98=UNCERTAINTY_MEASURE_WITH_UNIT(LENGTH_MEASURE(0.001),#92,
'DISTANCE_ACCURACY_VALUE',
'Maximum model space distance between geometric entities
at asserted connectivities');
#99=(
  GEOMETRIC_REPRESENTATION_CONTEXT(3)
  GLOBAL_UNCERTAINTY_ASSIGNED_CONTEXT((#98))
  GLOBAL_UNIT_ASSIGNED_CONTEXT((#97,#96,#92))
  REPRESENTATION_CONTEXT('ID1','3D')
);
#100=SHAPE_REPRESENTATION('Document',(#101,#102),#99);
#101=AXIS2_PLACEMENT_3D('',#107,#103,#104);
#102=AXIS2_PLACEMENT_3D('',#231,#105,#106);
#103=DIRECTION('',(0.,0.,1.));
#104=DIRECTION('',(1.,0.,0.));
#105=DIRECTION('',(0.,0.,1.));
#106=DIRECTION('',(1.,0.,0.));
#107=CARTESIAN_POINT('',(0.,0.,0.));
#108=CARTESIAN_POINT('',(3.,0.,0.));
#109=CARTESIAN_POINT('',(4.166666666666666,0.,0.));
#110=CARTESIAN_POINT('',(6.500000000000001,0.,0.));
#111=CARTESIAN_POINT('',(8.833333333333333,0.,0.));
#112=CARTESIAN_POINT('',(10.,0.,0.));
#113=CARTESIAN_POINT('',(3.,1.111111111111111,0.));
#114=CARTESIAN_POINT('',(4.166666666666666,1.111111111111111,0.));
#115=CARTESIAN_POINT('',(6.500000000000001,1.111111111111111,0.));
#116=CARTESIAN_POINT('',(8.833333333333333,1.111111111111111,0.));
#117=CARTESIAN_POINT('',(10.,1.111111111111111,0.));
#118=CARTESIAN_POINT('',(3.,3.333333333333333,0.));
#119=CARTESIAN_POINT('',(4.166666666666667,3.333333333333333,0.));
#120=CARTESIAN_POINT('',(6.5,3.333333333333333,0.));
#121=CARTESIAN_POINT('',(8.833333333333333,3.333333333333333,0.));
#122=CARTESIAN_POINT('',(10.,3.333333333333333,0.));
#123=CARTESIAN_POINT('',(3.,6.666666666666667,0.));
#124=CARTESIAN_POINT('',(4.166666666666666,6.666666666666667,0.));
#125=CARTESIAN_POINT('',(6.500000000000001,6.666666666666667,0.));
#126=CARTESIAN_POINT('',(8.833333333333333,6.666666666666667,0.));
#127=CARTESIAN_POINT('',(10.,6.666666666666667,0.));
#128=CARTESIAN_POINT('',(3.,8.888888888888889,0.));
#129=CARTESIAN_POINT('',(4.166666666666666,8.888888888888889,0.));
#130=CARTESIAN_POINT('',(6.500000000000001,8.888888888888889,0.));
#131=CARTESIAN_POINT('',(8.833333333333333,8.888888888888889,0.));
#132=CARTESIAN_POINT('',(10.,8.888888888888889,0.));
#133=CARTESIAN_POINT('',(3.,10.,0.));
#134=CARTESIAN_POINT('',(4.166666666666666,9.999999999999999,0.));
#135=CARTESIAN_POINT('',(6.500000000000001,10.,0.));
#136=CARTESIAN_POINT('',(8.833333333333333,10.,0.));
#137=CARTESIAN_POINT('',(10.,10.,0.));
#138=CARTESIAN_POINT('',(0.,0.,0.));
#139=CARTESIAN_POINT('',(1.166666666666667,0.,0.));
#140=CARTESIAN_POINT('',(3.5,0.,0.));
#141=CARTESIAN_POINT('',(5.833333333333333,0.,0.));
#142=CARTESIAN_POINT('',(7.,0.,0.));
#143=CARTESIAN_POINT('',(0.,1.666666666666667,0.));
#144=CARTESIAN_POINT('',(1.166666666666667,1.666666666666667,0.));
#145=CARTESIAN_POINT('',(3.5,1.666666666666667,0.));
#146=CARTESIAN_POINT('',(5.833333333333333,1.666666666666667,0.));
#147=CARTESIAN_POINT('',(7.,1.666666666666667,0.));
#148=CARTESIAN_POINT('',(0.,5.,0.));
#149=CARTESIAN_POINT('',(1.166666666666667,5.,0.));
#150=CARTESIAN_POINT('',(3.5,5.000000000000001,0.));
#151=CARTESIAN_POINT('',(5.833333333333333,5.,0.));
#152=CARTESIAN_POINT('',(7.,5.,0.));
#153=CARTESIAN_POINT('',(0.,8.333333333333333,0.));
#154=CARTESIAN_POINT('',(1.166666666666667,8.333333333333333,0.));
#155=CARTESIAN_POINT('',(3.5,8.333333333333333,0.));

#156=CARTESIAN_POINT('',(5.833333333333333,8.333333333333333,0.));
#157=CARTESIAN_POINT('',(7.,8.333333333333333,0.));
#158=CARTESIAN_POINT('',(0.,10.,0.));
#159=CARTESIAN_POINT('',(1.166666666666667,9.999999999999999,0.));
#160=CARTESIAN_POINT('',(3.5,10.,0.));
#161=CARTESIAN_POINT('',(5.833333333333333,10.,0.));
#162=CARTESIAN_POINT('',(7.,10.,0.));
#163=CARTESIAN_POINT('',(10.,10.,0.));
#164=CARTESIAN_POINT('',(10.,0.,0.));
#165=CARTESIAN_POINT('',(5.71428571428571,10.,0.));
#166=CARTESIAN_POINT('',(4.28571428571428,0.,0.));
#167=CARTESIAN_POINT('',(0.,0.,0.));
#168=CARTESIAN_POINT('',(0.,10.,0.));
#169=CARTESIAN_POINT('',(5.71428571428571,10.,0.));
#170=CARTESIAN_POINT('',(5.97619047619047,10.,0.));
#171=CARTESIAN_POINT('',(7.40476190476191,10.,0.));
#172=CARTESIAN_POINT('',(8.833333333333333,10.,0.));
#173=CARTESIAN_POINT('',(10.,10.,0.));
#174=CARTESIAN_POINT('',(10.,2.71428571428571));
#175=CARTESIAN_POINT('',(10.,7.));
#176=CARTESIAN_POINT('',(10.,10.,0.));
#177=CARTESIAN_POINT('',(10.,8.888888888888889,0.));
#178=CARTESIAN_POINT('',(10.,6.666666666666667,0.));
#179=CARTESIAN_POINT('',(10.,3.333333333333333,0.));
#180=CARTESIAN_POINT('',(10.,1.111111111111111,0.));
#181=CARTESIAN_POINT('',(10.,0.,0.));
#182=CARTESIAN_POINT('',(10.,7.));
#183=CARTESIAN_POINT('',(0.,7.));
#184=CARTESIAN_POINT('',(10.,0.,0.));
#185=CARTESIAN_POINT('',(8.833333333333333,0.,0.));
#186=CARTESIAN_POINT('',(6.92857142857143,0.,0.));
#187=CARTESIAN_POINT('',(5.02380952380952,0.,0.));
#188=CARTESIAN_POINT('',(4.28571428571428,0.,0.));
#189=CARTESIAN_POINT('',(0.,7.));
#190=CARTESIAN_POINT('',(0.,1.28571428571428));
#191=CARTESIAN_POINT('',(4.28571428571428,0.,0.));
#192=CARTESIAN_POINT('',(5.71428571428571,10.,0.));
#193=CARTESIAN_POINT('',(0.,1.28571428571428));
#194=CARTESIAN_POINT('',(1.111111111111111,1.444444444444444));
#195=CARTESIAN_POINT('',(2.222222222222222,1.6031746031746));
#196=CARTESIAN_POINT('',(3.333333333333333,1.76190476190476));
#197=CARTESIAN_POINT('',(4.444444444444444,1.92063492063492));
#198=CARTESIAN_POINT('',(5.555555555555555,2.07936507936508));
#199=CARTESIAN_POINT('',(6.666666666666667,2.23809523809524));
#200=CARTESIAN_POINT('',(7.777777777777778,2.3968253968254));
#201=CARTESIAN_POINT('',(8.888888888888889,2.555555555555555));
#202=CARTESIAN_POINT('',(10.,2.71428571428571));
#203=CARTESIAN_POINT('',(0.,4.28571428571428));
#204=CARTESIAN_POINT('',(1.166666666666667,4.52380952380953));
#205=CARTESIAN_POINT('',(3.333333333333333,4.76190476190476));
#206=CARTESIAN_POINT('',(5.,5.));
#207=CARTESIAN_POINT('',(6.666666666666667,5.23809523809524));
#208=CARTESIAN_POINT('',(8.333333333333333,5.476190476190476));
#209=CARTESIAN_POINT('',(10.,5.71428571428571));
#210=CARTESIAN_POINT('',(4.28571428571428,0.,0.));
#211=CARTESIAN_POINT('',(4.02380952380952,0.,0.));
#212=CARTESIAN_POINT('',(2.5952380952381,0.,0.));
#213=CARTESIAN_POINT('',(1.166666666666667,0.,0.));
#214=CARTESIAN_POINT('',(0.,0.,0.));
#215=CARTESIAN_POINT('',(0.,4.28571428571428));
#216=CARTESIAN_POINT('',(0.,0.,0.));
#217=CARTESIAN_POINT('',(0.,0.,0.));
#218=CARTESIAN_POINT('',(0.,1.666666666666667,0.));
#219=CARTESIAN_POINT('',(0.,5.,0.));
#220=CARTESIAN_POINT('',(0.,8.333333333333333,0.));
#221=CARTESIAN_POINT('',(0.,10.,0.));
#222=CARTESIAN_POINT('',(0.,0.,0.));
#223=CARTESIAN_POINT('',(10.,0.));
#224=CARTESIAN_POINT('',(0.,10.,0.));
#225=CARTESIAN_POINT('',(1.166666666666667,9.999999999999999,0.));
#226=CARTESIAN_POINT('',(3.07142857142857,10.,0.));
#227=CARTESIAN_POINT('',(4.97619047619047,10.,0.));
#228=CARTESIAN_POINT('',(5.71428571428571,10.,0.));
#229=CARTESIAN_POINT('',(10.,0.));
#230=CARTESIAN_POINT('',(10.,5.71428571428571));
#231=CARTESIAN_POINT('',(0.,0.,0.));
#232=(
  GEOMETRIC_REPRESENTATION_CONTEXT(2)
  PARAMETRIC_REPRESENTATION_CONTEXT()
  REPRESENTATION_CONTEXT('pspace','')
);
ENDSEC;
END-ISO-10303-21;

```

References

- [1] T. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer Methods in Applied Mechanics and Engineering* 194 (2005) 4135–4195.
- [2] J. A. Cottrell, T. J. R. Hughes, Y. Bazilevs, *Isogeometric analysis: Towards Integration of CAD and FEM*, John Wiley & Sons, 2009.
- [3] B. Marussig, T. J. R. Hughes, A review of trimming in isogeometric analysis: challenges, data exchange and simulation aspects, *Archives of Computational Methods in Engineering* doi:10.1007/s11831-017-9220-9 (2017).
- [4] M. Breitenberger, CAD-integrated design and analysis of shell structures, Dissertation, Technische Universität München, 2016.
- [5] M. Breitenberger, A. Apostolatos, B. Philipp, R. Wüchner, K.-U. Bletzinger, Analysis in computer aided design: Nonlinear isogeometric B-Rep analysis of shell structures, *Computer Methods in Applied Mechanics and Engineering* 284 (2015) 401–457.
- [6] Y. Guo, M. Ruess, Nitsche’s method for a coupling of isogeometric thin shells and blended shell structures, *Computer Methods in Applied Mechanics and Engineering* 284 (2015) 881–905.
- [7] Y. Guo, M. Ruess, D. Schillinger, A parameter-free variational coupling approach for trimmed isogeometric thin shells, *Computational Mechanics* 59 (2017) 693–715.
- [8] J. Kiendl, K.-U. Bletzinger, J. Linhard, R. Wüchner, Isogeometric shell analysis with Kirchhoff-Love elements, *Computer Methods in Applied Mechanics and Engineering* 198(49-52) (2009) 3902–3914.
- [9] D. Benson, Y. Bazilevs, M. Hsu, T. Hughes, Isogeometric shell analysis: the Reissner-Mindlin shell, *Computer Methods in Applied Mechanics and Engineering* 199 (2010) 276–289.
- [10] D. Benson, Y. Bazilevs, M. Hsu, T. Hughes, A large deformation, rotation-free, isogeometric shell, *Computer Methods in Applied Mechanics and Engineering* 200 (2011) 1367–1378.
- [11] S. Hosseini, J. Remmers, C. Verhoosel, R. Borst, An isogeometric solid-like shell element for nonlinear analysis, *International Journal for Numerical Methods in Engineering* 95 (2013) 238–256.
- [12] R. Echter, B. Oesterle, M. Bischoff, A hierarchic family of isogeometric shell finite elements, *Computer Methods in Applied Mechanics and Engineering* 254 (2013) 170–180.
- [13] J. Kiendl, F. Auricchio, L. B. da Veiga, C. Lovadina, A. Reali, Isogeometric collocation methods for the Reissner-Mindlin plate problem, *Computer Methods in Applied Mechanics and Engineering* 284 (2015) 489–507.
- [14] X. Deng, A. Korobenko, J. Yan, Y. Bazilevs, Isogeometric analysis of continuum damage in rotation-free composite shells, *Computer Methods in Applied Mechanics and Engineering* 284 (2015) 349–372.
- [15] R. Bouclier, T. Elguedj, A. Combescure, An isogeometric locking-free NURBS-based solid-shell element for geometrically nonlinear analysis, *International Journal for Numerical Methods in Engineering* 101 (2015) 774–808.
- [16] W. Dornisch, S. Klinkel, B. Simeon, Isogeometric Reissner-Mindlin shell analysis with exactly calculated director vectors, *Computer Methods in Applied Mechanics and Engineering* 253 (2013) 491–504.
- [17] W. Dornisch, R. Müller, S. Klinkel, An efficient and robust rotational formulation for isogeometric Reissner-Mindlin shell elements, *Computer Methods in Applied Mechanics and Engineering* 303 (2016) 1–34.
- [18] B. Oesterle, E. Ramm, M. Bischoff, A shear deformable, rotation-free isogeometric shell formulation, *Computer Methods in Applied Mechanics and Engineering* 307 (2016) 235–255.
- [19] T. X. Duong, F. Roohbakhshan, R. A. Sauer, A new rotation-free isogeometric thin shell formulation and a corresponding continuity constraint for patch boundaries, *Computer Methods in Applied Mechanics and Engineering* 316 (2017) 43–83.
- [20] A. Hansbo, P. Hansbo, A finite element method for the simulation of strong and weak discontinuities in solid mechanics, *Computer Methods in Applied Mechanics and Engineering* 193 (2004) 3523–3540.
- [21] A. Embar, J. Dolbow, I. Harari, Imposing Dirichlet boundary conditions with Nitsche’s method and spline-based finite elements, *International Journal for Numerical Methods in Engineering* 83(7) (2010) 877–898.
- [22] M. Ruess, D. Schillinger, Y. Bazilevs, V. Varduhn, E. Rank, Weakly enforced essential boundary conditions for NURBS-embedded and trimmed NURBS geometries on the basis of the finite cell method, *International Journal for Numerical Methods in Engineering* 95 (2013) 811–846.
- [23] I. Harari, E. Grosu, A unified approach for embedded boundary conditions for fourth-order elliptic problems, *International Journal for Numerical Methods in Engineering* 104 (2015) 655–675.

- [24] W. Jiang, C. Annavarapu, J. Dolbow, I. Harari, A robust Nitsche's formulation for interface problems with spline-based finite elements, *International Journal for Numerical Methods in Engineering* 104 (2015) 676–696.
- [25] D. Schillinger, I. Harari, M.-C. Hsu, D. Kamensky, K. Stoter, Y. Yu, Z. Ying, The nonsymmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements, *Computer Methods in Applied Mechanics and Engineering* 309 (2016) 625–652.
- [26] E. Burman, P. Hansbo, Deriving robust unfitted finite element methods from augmented Lagrangian formulations, *arXiv preprint arXiv:1702.08340* (2017).
- [27] B. Müller, F. Kummer, M. Oberlack, Highly accurate surface and volume integration on implicit domains by means of moment-fitting, *International Journal for Numerical Methods in Engineering* 96 (2013) 512–528.
- [28] D. Schillinger, M. Ruess, The finite cell method: A review in the context of higher-order structural analysis of cad and image-based geometric models, *Archives of Computational Methods in Engineering* 22 (2015) 391–455.
- [29] A. Nagy, D. Benson, On the numerical integration of trimmed isogeometric elements, *Computer Methods in Applied Mechanics and Engineering* 284 (2015) 165–185.
- [30] T.-P. Fries, S. Omerovic, Higher-order accurate integration of implicit geometries, *International Journal for Numerical Methods in Engineering* 106 (2016) 323–371.
- [31] L. Kudela, N. Zander, S. Kollmannsberger, E. Rank, Smart octrees: Accurately integrating discontinuous functions in 3D, *Computer Methods in Applied Mechanics and Engineering* (2016) doi:10.1016/j.cma.2016.04.006.
- [32] C. Lehrenfeld, High order unfitted finite element methods on level set domains using isoparametric mappings, *Computer Methods in Applied Mechanics and Engineering* 300 (2016) 716–733.
- [33] A. Stavrev, L. Nguyen, R. Shen, V. Varduhn, M. Behr, S. Elgeti, D. Schillinger, Geometrically accurate, efficient, and flexible quadrature techniques for the tetrahedral finite cell method, *Computer Methods in Applied Mechanics and Engineering* 310 (2016) 646–673.
- [34] Y. Guo, M. Ruess, Weak Dirichlet boundary conditions for trimmed thin isogeometric shells, *Computers and Mathematics with Applications* 70 (2015) 1425–1440.
- [35] B. Philipp, M. Breitenberger, I. D'Auria, R. Wüchner, K.-U. Bletzinger, Integrated design and analysis of structural membranes using the isogeometric B-Rep analysis, *Computer Methods in Applied Mechanics and Engineering* 303 (2016) 312–340.
- [36] A. Bauer, M. Breitenberger, B. Philipp, R. Wüchner, K.-U. Bletzinger, Embedded structural entities in NURBS-based isogeometric analysis, *Computer Methods in Applied Mechanics and Engineering* 325 (2017) 198–218.
- [37] Y. Bazilevs, T. Hughes, Weak imposition of Dirichlet boundary conditions in fluid mechanics, *Computers & Fluids* 36 (2007) 12–26.
- [38] Y. Bazilevs, M. Hsu, M. Scott, Isogeometric fluid-structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines, *Computer Methods in Applied Mechanics and Engineering* 249–252 (2012) 28–41.
- [39] Y. Zhao, D. Schillinger, B.-X. Xu, Variational boundary conditions based on the Nitsche method for fitted and unfitted isogeometric discretizations of the mechanically coupled Cahn–Hilliard equation, *Journal of Computational Physics* 340 (2017) 177–199.
- [40] L. Kudela, N. Zander, T. Bog, S. Kollmannsberger, E. Rank, Efficient and accurate numerical quadrature for immersed boundary methods, *Advanced Modeling and Simulation in Engineering Sciences* 2 (2015) 1–22.
- [41] M. Bischoff, W. Wall, K. Bletzinger, E. Ramm, Models and finite elements for thin-walled structures, in: E. Stein, R. de Borst, T. Hughes (Eds.), *Encyclopedia of Computational Mechanics*, John Wiley & Sons, Ltd., New York, 2004, pp. 59–137.
- [42] J. Reddy, *Mechanics of laminated composite plates and shells*, CRC Press LLC, Boca Raton, 2004.
- [43] P. Ciarlet, An introduction to differential geometry with applications to elasticity, *Journal of Elasticity* 78(1-3) (2005) 1–215.
- [44] W. J. Gordon, C. A. Hall, Transfinite element methods: blending function interpolation over arbitrary curved element domains, *Numerische Mathematik* 21 (1973) 109–129.
- [45] A. Stavrev, The role of higher-order geometry approximation and accurate quadrature in NURBS based immersed boundary method, Master's thesis, Technische Universität München, München, 2012.
- [46] L. Kudela, Highly Accurate Subcell Integration in the Context of The Finite Cell Method, Master's thesis, Technische Universität München, München, 2013.

- [47] J. Parvizian, A. Düster, E. Rank, Finite cell method - h- and p-extension for embedded domain problems in solid mechanics, *Computational Mechanics* 41 (2007) 121–133.
- [48] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, E. Rank, Small and large deformation analysis with the p- and B-spline versions of the finite cell method, *Computational Mechanics* 50 (2012) 445–478.
- [49] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, *Abhandlung aus dem Mathematischen Seminar der Universität Hamburg* 36 (1970) 9–15.
- [50] 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.
- [51] W. Pietraszkiewicz, Lagrangian description and incremental formulation in the non-linear theory of thin shells, *International Journal of Non-Linear Mechanics* 19 (1983) 115–140.
- [52] S. Opoka, W. Pietraszkiewicz, Intrinsic equations for non-linear deformation and stability of thin elastic shells, *International Journal Solids and Structures* 41 (2004) 3275–3292.
- [53] ISO 10303-1:1994, Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles, <https://www.iso.org/>, 1994.
- [54] M. J. Pratt, Introduction to ISO 10303 - The STEP standard for product data exchange, *Journal of Computing and Information Science in Engineering* 1 (2001) 1–4.
- [55] flexiCAD, Rhinoceros — NURBS modeling for Windows, <http://www.de.rhino3d.com/>, 2011.
- [56] Open CASCADE Technology, Maunul — 7.1.0. beta, <https://www.opencascade.com/>, 2017.
- [57] H.-J. Kim, Y.-D. Seo, S.-K. Youn, Isogeometric analysis for trimmed CAD surfaces, *Computer Methods in Applied Mechanics and Engineering* 198 (2009) 2982–2995.
- [58] V. Varduhn, M.-C. Hsu, M. Ruess, D. Schillinger, The tetrahedral finite cell method: Higher-order immersogeometric analysis on adaptive non-boundary-fitted meshes, *International Journal for Numerical Methods in Engineering* 107 (2016) 1054–1079.
- [59] O. Marco, R. Sevilla, Y. Zhang, J. Ródenas, M. Tur, Exact 3D boundary representation in finite element analysis based on Cartesian grids independent of the geometry, *International Journal for Numerical Methods in Engineering* 103 (2015) 445–468.
- [60] N. Buechter, E. Ramm, Shell theory versus degeneration – a comparison in large rotation finite element analysis, *International Journal for Numerical Methods in Engineering* 34 (1992) 39–59.
- [61] Y. Başar, Y. Ding, Finite-rotation shell elements for the analysis of finite-rotation shell problems, *International Journal for Numerical Methods in Engineering* 34 (1992) 165–169.
- [62] K. Sze, X. Liu, S. Lo, Popular benchmark problems for geometric nonlinear analysis of shells, *Finite Elements in Analysis and Design* 40 (2004) 1551–1569.
- [63] C. Felippa, Introduction to finite element methods, Course notes from UC Boulder, available online at <http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/Home.html> (Accessed on 10/10/2017).
- [64] J. Simo, D. Fox, M. Rifai, On a stress resultant geometrically exact shell model. Part III: Computational aspects of the nonlinear theory, *Computer Methods in Applied Mechanics and Engineering* 79 (1990) 21–70.
- [65] G. Horrigmoe, P. Bergan, Nonlinear analysis of free-form shells by flat finite elements, *Computer Methods in Applied Mechanics and Engineering* 16 (1978) 11–35.
- [66] T. Hughes, W.-K. Liu, Nonlinear finite element analysis of shells: Part I. Three-dimensional shells, *Computer Methods in Applied Mechanics and Engineering* 26 (1981) 331–362.
- [67] M. A. Crisfield, A fast incremental/iterative solution procedure that handles snap-through, *Computers & Structures* 13 (1981) 55–62.
- [68] R. MacNeal, R. Harder, A proposed standard set of problems to test finite element accuracy, *Finite Elements in Analysis and Design* 1 (1985) 3–20.
- [69] J. Bonet, R. Wood, *Nonlinear Continuum Mechanics for Finite Element Analysis*, Cambridge University Press, 2008.
- [70] E. de Souza Neto, D. Perić, D. Owen, *Computational Methods for Plasticity: Theory and Applications*, Wiley, 2008.
- [71] P. Wriggers, *Nonlinear finite element methods*, Springer, 2008.
- [72] Hibbit, Karlsson, Sorensen, *ABAQUS/Standard Analysis User’s Manual*, Hibbit, Karlsson, Sorensen Inc., USA,

2007.

- [73] D. Benson, S. Hartmann, Y. Bazilevs, M. Hsu, T. Hughes, Blended isogeometric shells, *Computer Methods in Applied Mechanics and Engineering* 255 (2013) 133–146.
- [74] S. Morganti, F. Auricchio, D. Benson, F. Gambarin, S. Hartmann, T. Hughes, A. Reali, Patient-specific isogeometric structural analysis of aortic valve closure, *Computer Methods in Applied Mechanics and Engineering* 284 (2015) 508–520.