



Kirchhoff–Love shell theory based on tangential differential calculus

D. Schöllhammer¹ · T. P. Fries¹

Received: 28 May 2018 / Accepted: 14 November 2018 / Published online: 29 November 2018
© The Author(s) 2018

Abstract

The Kirchhoff–Love shell theory is recasted in the frame of the tangential differential calculus (TDC) where differential operators on surfaces are formulated based on global, three-dimensional coordinates. As a consequence, there is no need for a parametrization of the shell geometry implying curvilinear surface coordinates as used in the classical shell theory. Therefore, the proposed TDC-based formulation also applies to shell geometries which are zero-isosurfaces as in the level-set method where no parametrization is available in general. For the discretization, the TDC-based formulation may be used based on surface meshes implying element-wise parametrizations. Then, the results are equivalent to those obtained based on the classical theory. However, it may also be used in recent finite element approaches as the TraceFEM and CutFEM where shape functions are generated on a background mesh without any need for a parametrization. Numerical results presented herein are achieved with isogeometric analysis for classical and new benchmark tests. Higher-order convergence rates in the residual errors are achieved when the physical fields are sufficiently smooth.

Keywords Shells · Tangential differential calculus · TDC · Isogeometric analysis · IGA · Manifolds

1 Introduction

The mechanical modeling of shells leads to partial differential equations (PDEs) on manifolds where the manifolds are curved surfaces in the three-dimensional space. An overview in classical shell theory is given, e.g., in [4,9,32,44,45] or in the textbooks [1,5,41,49]. When modeling physical phenomena on curved surfaces, definitions for geometric quantities (normal vectors, curvatures, etc.) and differential surface operators (gradients, divergence, etc.) are key ingredients. These quantities may be either defined based on two-dimensional, curvilinear *local* coordinates living on the manifold or on *global* coordinates of the surrounding, three-dimensional space.

In the first case, the curved surface is parametrized by two parameters, i.e., there is a given map from the two-dimensional parameter space to the three-dimensional

physical space, see Fig. 1a. For the definition of geometrical quantities and surface operators, co- and contra-variant base vectors and Christoffel-symbols naturally occur. It is important to note that a parametrization of a surface is not unique, hence, there are infinitely many maps which result in the same curved surface. Obviously, the physical modeling must be independent of a concrete parametrization, which suggests the existence of a parametrization-free formulation.

In the second case, the geometric quantities and surface operators are based on global coordinates as done in the tangential differential calculus (TDC) [15,25,28]. Then, a model may also be defined even if a parametrization of a curved surface does not exist, for example, when it is a zero-isosurface of a scalar function in three dimensions following the level-set method [21,22,39,43]. When the physical modeling is based on the TDC, i.e., on global coordinates, it is applicable to surfaces which are parametrized or not. In this sense, the TDC-based approach is more general than approaches based on local coordinates. Models based on the TDC are found in various applications, see [16–18,22] for scalar problems such as heat flow and [20,31] for flow problems on manifolds. In the context of structure mechanics, this approach is used in [29] for curved beams, in [25,26,28] for membranes, and in [27] for flat shells embedded in \mathbb{R}^3 .

✉ D. Schöllhammer
schoellhammer@tugraz.at
http://www.ifb.tugraz.at

T. P. Fries
fries@tugraz.at
http://www.ifb.tugraz.at

¹ Institute of Structural Analysis, Graz University of Technology, Lessingstr. 25/II, 8010 Graz, Austria

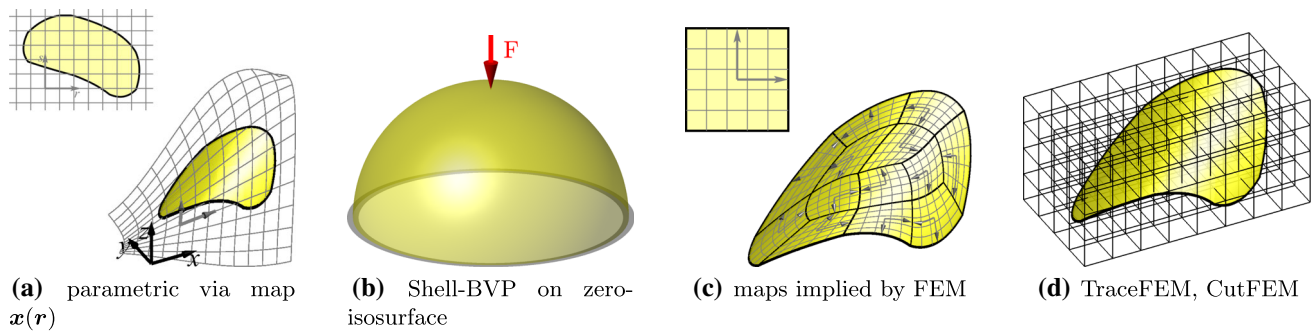


Fig. 1 **a** In classical shell mechanics, the middle surface is defined by a parametrization, i.e., a map $\mathbf{x}(r)$. **b** The cupola is given by the zero-isosurface of $\phi(\mathbf{x})$ and the mechanical response to the force F is sought. **c** The surface mesh implies element-wise, approximate parametriza-

tions even if the initial geometry is defined by level-sets. **d** For implicitly defined shells in the context of TraceFEM and CutFEM, no parametrization is needed at all

Herein, we apply the TDC for the reformulation of the classical Kirchhoff–Love shell theory which is typically formulated based on a given parametrization. Based on the TDC, it is possible to also formulate the boundary value problem (BVP) for shell geometries where no parametrization is given as for the example in Fig. 1b: the cupola with radius r is given by the zero-isosurface of $\phi(\mathbf{x}) = \|\mathbf{x}\| - r$ with $\mathbf{x} \in [-r, r]^2 \times [0, r]$ and the mechanical response to the force F is sought. As mentioned before, the TDC-based formulation is also valid when a parametrization is available; it is then equivalent to the classical formulation based on local coordinates.

Other attempts to parametrization-free formulations of the Kirchhoff–Love shell theory are found, e.g., in [11–14] with a mathematical focus and in [33,47,50] from an engineering perspective, however, only with focus on displacements. Herein, the Kirchhoff–Love shell theory is recasted in the frame of the TDC including all relevant mechanical aspects. For the first time, the parametrization-free strong form of the Kirchhoff–Love shell is given and taken as the starting point to derive the weak form. Then, boundary terms for the relevant boundary conditions of Kirchhoff–Love shell theory are naturally achieved. Furthermore, mechanical quantities such as moments, normal and shear forces are defined based on global coordinates and it is shown how (parametrization-)invariant quantities such as principal moments are computed. Finally, the strong form of Kirchhoff–Love shells is also found highly useful to define residual errors in the numerical results. Of course, evaluating this error in the strong form requires up to forth-order derivatives on the surface, which is implementationally quite some effort. The advantage, however, is that one may then confirm higher-order convergence rates in the corresponding error norm for suitable shell test cases. This is, otherwise, very difficult as exact solutions for shells are hardly available and

classical benchmark tests typically give only selected scalar quantities, often with moderate accuracy.

For the numerical solution of shells, i.e., the approximation of the shell BVP based on numerical methods, we distinguish two fundamentally different approaches. The first is a classical finite element analysis based on a surface mesh, labelled Surface FEM herein [16,18,20,22]. Once a surface mesh is generated, it implies element-wise parametrizations for the shell geometry, see Fig. 1c, no matter whether the underlying (analytic) geometry was parametrized or implied by level sets. In this case, classical shell theory based on parametrizations is suitable at least for the discretized geometry. The proposed TDC-based formulation is suitable as well which shall be seen in the numerical results. The other numerical approach is to use a three-dimensional background mesh into which the curved shell surface is embedded, cf. Fig. 1d. Then, the shape functions of the (three-dimensional) background elements are only evaluated on the shell surface and no parametrization (and surface mesh) is needed to furnish basis functions for the approximation. For these methods, e.g., labelled CutFEM [6–8,19] or TraceFEM [23,37,38,42], applied to the case of shell mechanics, it is no longer possible to rely on classical parametrization-based formulations of the shell mechanics, however, the proposed TDC-based formulation is still applicable.

For the numerical results presented herein, the continuous weak form of the BVP is discretized with the Surface FEM [16,18,20,22] using NURBS as trial and test functions as proposed by Hughes et al. [10,30] due to the continuity requirements of Kirchhoff–Love shells. The boundary conditions are weakly enforced via Lagrange multipliers [51]. The situation is similar to [2,32,35,36], however, based on the proposed view point, the implementation is quite different. In particular, when PDEs on manifolds from other application fields than shell mechanics are also of interest (e.g., when transport problems [16–18] or flow problems [20,31]

on curved surfaces are considered), there is a unified and elegant way to handle this by computing surface gradients applied to finite element shape functions which simplifies the situation considerably. In that sense one may shift significant parts of the implementation needed for shells to the underlying finite element technology and recycle this in other situations where PDEs on surfaces are considered.

We summarize the advantages of the TDC-based formulation of Kirchhoff–Love shells: (1) the definition of the BVP does not need a parametrization of the surface (though it can also handle the classical situation where a parametrization is given), (2) the TDC-based formulation is also suitable for very recent finite element technologies such as CutFEM and TraceFEM (though the typical approach based on the Surface FEM or IGA is also possible and demonstrated herein), (3) the implementation is advantageous in finite element (FE) codes where other PDEs on manifolds are considered as well due to the split of FE technology and application. From a didactic point of view, it may also be advantageous that troubles with curvilinear coordinates (co- and contra-variance, Christoffel-symbols) are avoided in the TDC-based approach where surface operators and geometric quantities are expressed in tensor notation.

The outline of the paper is as follows: In Sect. 2, important surface quantities are defined, and an introduction to the tangential differential calculus (TDC) is given. In Sect. 3, the classical linear Kirchhoff–Love shell equations under static loading are recast in terms of the TDC. Stress resultants such as membrane forces, bending moments, transverse shear forces and corner forces are defined. In Sect. 4, implementational aspects are considered. The element stiffness matrix and the resulting system of linear equations are shown. The implementation of boundary conditions based on Lagrange multipliers is outlined. Finally, in Sect. 5, numerical results are presented. The first example is a flat shell embedded in \mathbb{R}^3 , where an analytical solution is available. The second and third example are parts of popular benchmarks as proposed in [2]. In the last example, a more general geometry without analytical solution or reference displacement is considered. The error is measured in the strong form of the equilibrium in order to verify the proposed approach and higher-order convergence rates are achieved.

2 Preliminaries

Shells are geometrical objects, where one dimension is significantly smaller compared to the other two dimensions. In this case, the shell can be reduced to a surface Γ embedded in the physical space \mathbb{R}^3 . In particular, the surface is a manifold of codimension 1. Let the surface be possibly curved, sufficiently smooth, orientable, connected and bounded by $\partial\Gamma$. There are two alternatives for defining the shell geometry.

One is through a parametrization, i.e., a (bijective) mapping

$$\mathbf{x}(\mathbf{r}) : \hat{\Omega} \rightarrow \Gamma \tag{1}$$

from the parameter space $\hat{\Omega} \subset \mathbb{R}^2$ to the real domain $\Gamma \subset \mathbb{R}^3$. The other approach is based on the level-set method. Then, a level-set function $\phi(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}$ with $\mathbf{x} \in \Omega \subset \mathbb{R}^3$ exists and the shell is implicitly given by

$$\Gamma = \{ \mathbf{x} : \phi(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \Omega \} . \tag{2}$$

Additional level-set functions may restrict the zero-surface to the desired, bounded shell as described in [22]. In Fig. 2a, b the two different approaches are schematically shown.

The definition of the normal vector depends on whether the shell geometry is based on a parametrization or not. In the first case (cf. Fig. 2a), the shell geometry results from a map $\mathbf{x}(\mathbf{r})$. Then, the normal vector \mathbf{n}_Γ of the shell surface is determined by a cross-product of the columns of the Jacobi matrix $\mathbf{J}(\mathbf{r}) = \partial\mathbf{x}/\partial\mathbf{r}$. The resulting geometric quantities, surface operators, and models in this case are parametrization-based.

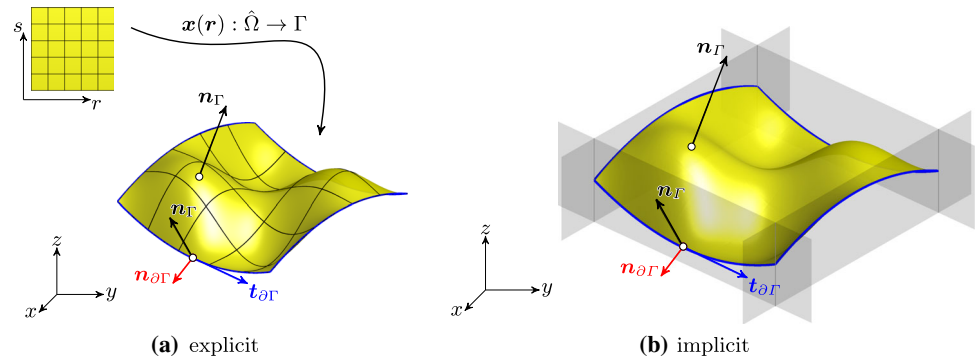
In the case where the shell geometry is implied by the zero-surface of a level-set function $\phi(\mathbf{x})$ (cf. Fig. 2b) and no parametrization is available, the normal vector may be determined by $\mathbf{n}_\Gamma = \nabla\phi/\|\nabla\phi\|$. All resulting quantities including the BVP of the Kirchhoff–Love shell are parametrization-free in this case. Of course, when in the wake of discretizing the BVP, the Surface FEM is used for the approximation, then a surface mesh of the shell geometry is needed and the surface elements do imply a parametrization again. It was already mentioned above, that other numerical methods such as the TraceFEM and CutFEM do not rely on a surface mesh. In this case, the continuous and discrete BVP for the shell are truly parametrization-free.

In addition to the normal vector on the surface, along the boundary $\partial\Gamma$ there is an associated tangential vector $\mathbf{t}_{\partial\Gamma} \in \mathbb{R}^3$ pointing in the direction of $\partial\Gamma$ and a co-normal vector $\mathbf{n}_{\partial\Gamma} = \mathbf{n}_\Gamma \times \mathbf{t}_{\partial\Gamma} \in \mathbb{R}^3$ pointing “outwards” and being perpendicular to the boundary yet in the tangent plane of the surface Γ . For the proof of equivalence of both cases we refer to, e.g., [18].

2.1 Tangential differential calculus

The TDC provides a framework to define differential operators avoiding the use of classical differential geometric methods based on local coordinate systems and Christoffel symbols. In the following, an overview of the operators and relations in the frame of the TDC are presented. For simplicity, we restrict ourselves to the case of surfaces embedded in the three dimensional space. However, the shown relations and definitions may be adopted to other situations

Fig. 2 Examples of bounded surfaces Γ embedded in the physical space \mathbb{R}^3 : **a** explicitly defined surface with a map $\mathbf{x}(\mathbf{r})$, **b** implicitly defined surface with a master level-function $\phi(\mathbf{x}) = 0$ (yellow) and slave level-set functions ψ_i for the boundary definition (gray)



accordingly (e.g., curved lines embedded in 2D or 3D). An introduction from a more mathematical point of view is given in [15,25,31].

Orthogonal projection operator P

The orthogonal projection operator or normal projector $\mathbf{P} \in \mathbb{R}^{3 \times 3}$ is defined as

$$\mathbf{P} = \mathbb{I} - \mathbf{n}_\Gamma \otimes \mathbf{n}_\Gamma . \tag{3}$$

The operator \otimes is the dyadic product of two vectors. The normal projector \mathbf{P} projects a vector \mathbf{v} onto the tangent space $T_P \Gamma$ of the surface. Note that \mathbf{P} is idempotent ($\mathbf{P} \cdot \mathbf{P} = \mathbf{P}$), symmetric ($\mathbf{P} = \mathbf{P}^T$) and obviously in the tangent space $T_P \Gamma$ of the surface, i.e., $\mathbf{P} \cdot \mathbf{n}_\Gamma = \mathbf{n}_\Gamma^T \cdot \mathbf{P} = \mathbf{0}$.

The projection of a vector field $\mathbf{v} : \Gamma \rightarrow \mathbb{R}^3$ onto the tangent plane is defined by

$$\mathbf{v}_t = \mathbf{P} \cdot \mathbf{v} \in T_P \Gamma \tag{4}$$

where \mathbf{v}_t is tangential, i.e. $\mathbf{v}_t \cdot \mathbf{n}_\Gamma = 0$. The double projection of a second-order tensor function $\mathbf{A}(\mathbf{x}) : \Gamma \rightarrow \mathbb{R}^{3 \times 3}$ leads to an in-plane tensor and is defined as

$$\mathbf{A}_t = \mathbf{P} \cdot \mathbf{A} \cdot \mathbf{P} \in T_P \Gamma , \tag{5}$$

with the properties $\mathbf{A}_t = \mathbf{P} \cdot \mathbf{A}_t \cdot \mathbf{P}$ and $\mathbf{A}_t \cdot \mathbf{n}_\Gamma = \mathbf{n}_\Gamma^T \cdot \mathbf{A}_t = \mathbf{0}$.

Tangential gradient of scalar functions

The tangential gradient ∇_Γ of a scalar function $u : \Gamma \rightarrow \mathbb{R}$ on the manifold is defined as

$$\nabla_\Gamma u(\mathbf{x}) = \mathbf{P}(\mathbf{x}) \cdot \nabla \tilde{u}(\mathbf{x}) , \quad \nabla_\Gamma u(\mathbf{x}) \in \mathbb{R}^{3 \times 1} , \quad \mathbf{x} \in \Gamma \tag{6}$$

where ∇ is the standard gradient operator in the physical space and \tilde{u} is a smooth extension of u in a neighbourhood \mathcal{U} of the manifold Γ . Alternatively, \tilde{u} is given as a function in global coordinates $\tilde{u}(\mathbf{x}) : \mathbb{R}^3 \rightarrow \mathbb{R}$ and only evaluated at the manifold $\tilde{u}|_\Gamma = u$.

For parametrized surfaces defined by the map $\mathbf{x}(\mathbf{r})$, and a given scalar function $u(\mathbf{r}) : \hat{\Omega} \rightarrow \mathbb{R}$, the tangential gradient can be determined without explicitly computing an extension \tilde{u} using

$$\nabla_\Gamma u(\mathbf{x}(\mathbf{r})) = \mathbf{J}(\mathbf{r}) \cdot \mathbf{G}(\mathbf{r})^{-1} \cdot \nabla_r u(\mathbf{r}) , \tag{7}$$

with $\mathbf{J}(\mathbf{r}) = \partial \mathbf{x} / \partial \mathbf{r} \in \mathbb{R}^{3 \times 2}$ being the Jacobi matrix, $\mathbf{G} = \mathbf{J}^T \cdot \mathbf{J}$ is the metric tensor or the first fundamental form and the operator ∇_r is the gradient with respect to the reference coordinates. The components of the tangential gradient are denoted by

$$\nabla_\Gamma u = \begin{bmatrix} \partial_x^\Gamma u \\ \partial_y^\Gamma u \\ \partial_z^\Gamma u \end{bmatrix} , \tag{8}$$

representing first-order partial tangential derivatives. An important property of $\nabla_\Gamma u$ is that the tangential gradient of a scalar-valued function is in the tangent space of the surface $\nabla_\Gamma u \in T_P \Gamma$, i.e., $\nabla_\Gamma u \cdot \mathbf{n}_\Gamma = 0$. When using the Surface FEM to solve BVPs on surfaces, one may use Eq. (7) to compute tangential gradients of the shape functions. If, on the other hand, TraceFEM or CutFEM is used, one may use Eq. (6).

Tangential gradient of vector-valued functions

Consider a vector-valued function $\mathbf{v}(\mathbf{x}) : \Gamma \rightarrow \mathbb{R}^3$ and apply to each component of \mathbf{v} the tangential gradient for scalars. This leads to the *directional* gradient of \mathbf{v} defined as

$$\nabla_\Gamma^{\text{dir}} \mathbf{v}(\mathbf{x}) = \nabla_\Gamma^{\text{dir}} \begin{bmatrix} u(\mathbf{x}) \\ \mathbf{v}(\mathbf{x}) \\ w(\mathbf{x}) \end{bmatrix} = \begin{bmatrix} \partial_x^\Gamma u & \partial_y^\Gamma u & \partial_z^\Gamma u \\ \partial_x^\Gamma v & \partial_y^\Gamma v & \partial_z^\Gamma v \\ \partial_x^\Gamma w & \partial_y^\Gamma w & \partial_z^\Gamma w \end{bmatrix} . \tag{9}$$

Note that the directional gradient is *not* in the tangent space of the surface, in general. A projection of the directional gradient to the tangent space leads to the *covariant* gradient

of \mathbf{v} and is defined as

$$\nabla_{\Gamma}^{\text{cov}} \mathbf{v} = \mathbf{P} \cdot \nabla_{\Gamma}^{\text{dir}} \mathbf{v}, \tag{10}$$

which is an in-plane tensor, i.e., $\nabla_{\Gamma}^{\text{cov}} \mathbf{v} \in T_{\mathcal{P}} \Gamma$. The covariant gradient often appears in the modelling of physical phenomena on manifolds, i.e., in the governing equations. In contrast the directional gradient appears naturally in product rules or divergence theorems on manifolds.

In the following, partial surface derivatives of scalar functions are denoted as $\partial_{x_i}^{\Gamma} u$ or $u_{,i}^{\Gamma}$ with $i = 1, 2, 3$. Partial surface derivatives of vector or tensor components are denoted as $v_{i,j}^{\text{dir}}$ for directional and $v_{i,j}^{\text{cov}}$ for covariant derivatives with $i, j = 1, 2, 3$.

Tangential gradient of tensor functions

For a second-order tensor function $\mathbf{A}(\mathbf{x}) : \Gamma \rightarrow \mathbb{R}^{3 \times 3}$, the partial directional gradient with respect to x_i is defined as

$$\nabla_{\Gamma,i}^{\text{dir}} \mathbf{A} = \frac{\partial \mathbf{A}}{\partial x_i} = \begin{bmatrix} \partial_{x_i}^{\Gamma} A_{11} & \partial_{x_i}^{\Gamma} A_{12} & \partial_{x_i}^{\Gamma} A_{13} \\ \partial_{x_i}^{\Gamma} A_{21} & \partial_{x_i}^{\Gamma} A_{22} & \partial_{x_i}^{\Gamma} A_{23} \\ \partial_{x_i}^{\Gamma} A_{31} & \partial_{x_i}^{\Gamma} A_{32} & \partial_{x_i}^{\Gamma} A_{33} \end{bmatrix}, \tag{11}$$

with $i = 1, 2, 3$. The directional gradient of the tensor function is then defined as

$$\nabla_{\Gamma}^{\text{dir}} \mathbf{A} = \left(\nabla_{\Gamma,1}^{\text{dir}} \mathbf{A} \quad \nabla_{\Gamma,2}^{\text{dir}} \mathbf{A} \quad \nabla_{\Gamma,3}^{\text{dir}} \mathbf{A} \right). \tag{12}$$

The covariant partial derivative is determined by projecting the partial directional derivative onto the tangent space

$$\nabla_{\Gamma,i}^{\text{cov}} \mathbf{A} = \mathbf{P} \cdot \nabla_{\Gamma,i}^{\text{dir}} \mathbf{A} \cdot \mathbf{P}. \tag{13}$$

Second-order tangential derivatives

Next, second-order derivatives of scalar functions are considered. The directional second order gradient of a scalar function u is defined by

$$\begin{aligned} \{\mathbf{H}\mathbf{e}^{\text{dir}}\}_{ij}(u(\mathbf{x})) &= \partial_{x_j}^{\Gamma, \text{dir}} \left(\partial_{x_i}^{\Gamma} u(\mathbf{x}) \right) = u_{,ji}^{\text{dir}} \\ &= \begin{bmatrix} \partial_{xx}^{\Gamma} u & \partial_{yx}^{\Gamma} u & \partial_{zx}^{\Gamma} u \\ \partial_{xy}^{\Gamma} u & \partial_{yy}^{\Gamma} u & \partial_{zy}^{\Gamma} u \\ \partial_{xz}^{\Gamma} u & \partial_{yz}^{\Gamma} u & \partial_{zz}^{\Gamma} u \end{bmatrix} = \nabla_{\Gamma}^{\text{dir}} (\nabla_{\Gamma} u(\mathbf{x})) \end{aligned} \tag{14}$$

where $\mathbf{H}\mathbf{e}^{\text{dir}}$ is the tangential Hessian matrix which is not symmetric in the case of curved manifolds [15], i.e., $u_{,ij}^{\text{dir}} \neq u_{,ji}^{\text{dir}}$. For the case of parametrized surfaces and a given scalar function in the reference space, the tangential Hessian matrix can be determined by

$$\begin{aligned} \mathbf{H}\mathbf{e}^{\text{dir}}(u) &= \nabla_{\Gamma}^{\text{dir}} (\mathbf{Q} \cdot \nabla_r u) \\ &= [\mathbf{Q}_{,r} \cdot \nabla_r u \quad \mathbf{Q}_{,s} \cdot \nabla_r u] \cdot \mathbf{Q}^{\text{T}} \\ &\quad + \mathbf{Q} \cdot \nabla_r (\nabla_r u) \cdot \mathbf{Q}^{\text{T}} \end{aligned} \tag{15}$$

where $\mathbf{Q} = \mathbf{J} \cdot \mathbf{G}^{-1}$, and $\mathbf{Q}_{,r_i}$ denotes the partial tangential derivative of \mathbf{Q} with respect to r_i . The covariant counterpart is

$$\mathbf{H}\mathbf{e}^{\text{cov}}(u) = \nabla_{\Gamma}^{\text{cov}} (\nabla_{\Gamma} u) = \mathbf{P} \cdot \nabla_{\Gamma}^{\text{dir}} (\nabla_{\Gamma} u) = \mathbf{P} \cdot \mathbf{H}\mathbf{e}^{\text{dir}}(u). \tag{16}$$

In contrast to $\mathbf{H}\mathbf{e}^{\text{dir}}$, $\mathbf{H}\mathbf{e}^{\text{cov}}$ is symmetric and an in-plane tensor [48]. In the special case of flat surfaces embedded in \mathbb{R}^3 the directional and covariant Hessian matrix are equal.

Tangential divergence operators

The divergence operator of a vector-valued function $\mathbf{v}(\mathbf{x}) : \Gamma \rightarrow \mathbb{R}^3$ is given as

$$\text{div}_{\Gamma} \mathbf{v}(\mathbf{x}) = \text{tr} \left(\nabla_{\Gamma}^{\text{dir}} \mathbf{v}(\mathbf{x}) \right) = \text{tr} \left(\nabla_{\Gamma}^{\text{cov}} \mathbf{v}(\mathbf{x}) \right), \tag{17}$$

and the divergence of a matrix or tensor function $\mathbf{A}(\mathbf{x}) : \Gamma \rightarrow \mathbb{R}^{3 \times 3}$, is

$$\text{div}_{\Gamma} \mathbf{A}(\mathbf{x}) = \begin{bmatrix} \text{div}_{\Gamma} [A_{11}, A_{12}, A_{13}] \\ \text{div}_{\Gamma} [A_{21}, A_{22}, A_{23}] \\ \text{div}_{\Gamma} [A_{31}, A_{32}, A_{33}] \end{bmatrix}. \tag{18}$$

Note that $\text{div}_{\Gamma} \mathbf{A}$ is, in general, not a tangential vector. It would only be tangential if the surface is flat and \mathbf{A} is an in-plane tensor.

Weingarten map and curvature

The Weingarten map as introduced in [15,31] is defined as

$$\mathbf{H} = \nabla_{\Gamma}^{\text{dir}} \mathbf{n}_{\Gamma} = \nabla_{\Gamma}^{\text{cov}} \mathbf{n}_{\Gamma} \tag{19}$$

and is related to the second fundamental form in differential geometry. The Weingarten map is a symmetric, in-plane tensor and its two non-zero eigenvalues are associated with the principal curvatures

$$\kappa_{1,2} = -\text{eig}(\mathbf{H}). \tag{20}$$

The minus in Eq. (20) is due to fact that the Weingarten map is defined with the “outward” unit normal vector instead of the “inward” unit normal vector, which leads to positive curvatures of a sphere. The third eigenvalue is zero, because \mathbf{H}

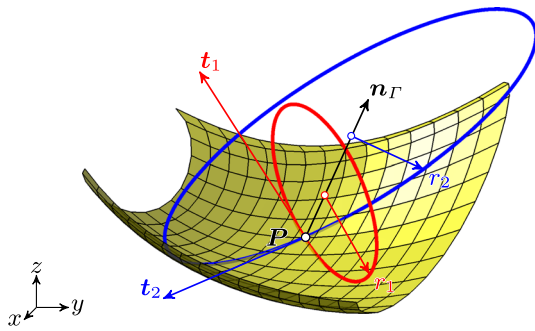


Fig. 3 Osculating circles (blue, red) and eigenvectors (t_1, t_2, n_Γ) of \mathbf{H} at point P on a surface embedded in \mathbb{R}^3 . (Color figure online)

is an in-plane tensor. The corresponding eigenvectors t_1, t_2 and n_Γ are perpendicular as \mathbf{H} is symmetric. In Fig. 3, the osculating circles with the radii $r_i = 1/\kappa_i$ and the eigenvectors at a point P are shown.

The Gauß curvature is defined as the product of the principal curvatures $K = \prod_{i=1}^2 \kappa_i$ and the mean curvature is introduced as $\varkappa = \kappa_1 + \kappa_2 = \text{tr}(\mathbf{H})$.

Divergence theorems in terms of tangential operators

The divergence theorem or Green’s formula for a scalar function $f \in C^1(\Gamma)$ and a vector valued function $\mathbf{v} \in C^1(\Gamma)^3$ are defined as in [13,15]

$$\int_\Gamma f \cdot \text{div}_\Gamma \mathbf{v} \, d\Gamma = - \int_\Gamma \nabla_\Gamma f \cdot \mathbf{v} \, d\Gamma + \int_\Gamma \varkappa f (\mathbf{v} \cdot \mathbf{n}_\Gamma) \, d\Gamma + \int_{\partial\Gamma} f \mathbf{v} \cdot \mathbf{n}_{\partial\Gamma} \, ds. \tag{21}$$

The term with the mean curvature \varkappa is vanishing if the vector \mathbf{v} is tangential, then $\mathbf{v} \cdot \mathbf{n}_\Gamma = 0$. In extension to Eq. (21), Green’s formula for second order tensor functions $\mathbf{A} \in C^1(\Gamma)^{3 \times 3}$, is

$$\int_\Gamma \mathbf{v} \cdot \text{div}_\Gamma \mathbf{A} \, d\Gamma = - \int_\Gamma \nabla_\Gamma^{\text{dir}} \mathbf{v} : \mathbf{A} \, d\Gamma + \int_\Gamma \varkappa \mathbf{v} \cdot (\mathbf{A} \cdot \mathbf{n}_\Gamma) \, d\Gamma + \int_{\partial\Gamma} \mathbf{v} \cdot (\mathbf{A} \cdot \mathbf{n}_{\partial\Gamma}) \, ds \tag{22}$$

where $\nabla_\Gamma^{\text{dir}} \mathbf{v} : \mathbf{A} = \text{tr}(\nabla_\Gamma^{\text{dir}} \mathbf{v} \cdot \mathbf{A}^\top)$. In the case of in-plane tensors, e.g., $\mathbf{A}_t = \mathbf{P} \cdot \mathbf{A}_t \cdot \mathbf{P}$, the term with the mean curvature \varkappa vanishes due to $\mathbf{A}_t \cdot \mathbf{n}_\Gamma = \mathbf{0}$ and we also have $\nabla_\Gamma^{\text{dir}} \mathbf{v} : \mathbf{A}_t = \nabla_\Gamma^{\text{cov}} \mathbf{v} : \mathbf{A}_t$.

3 The shell equations

In this section, we derive the linear Kirchhoff–Love shell theory in the frame of tangential operators based on a global Cartesian coordinate system. We restrict ourselves to infinitesimal deformations, which means that the reference and spatial configuration are indistinguishable. Furthermore, a linear elastic material governed by Hooke’s law is assumed. As usual in the Kirchhoff–Love shell theory, the transverse shear strains and the change of curvature in the material law are neglected, which restricts the model to thin shells ($t\kappa_{\max} \ll 1$).

With these assumptions, an analytical pre-integration with respect to the thickness leads to stress resultants such as normal forces and bending moments. The equilibrium in strong form is then expressed in terms of the stress resultants. Finally, the transverse shear forces may be identified via equilibrium considerations.

3.1 Kinematics

The middle surface Γ of the shell is a sufficiently smooth manifold embedded in the physical space \mathbb{R}^3 . A point on the middle surface is denoted as $\mathbf{x}_\Gamma \in \Gamma \subset \mathbb{R}^3$ and may be obtained explicitly or implicitly, see Sect. 2. With the unit-normal vector \mathbf{n}_Γ a point in the domain of the shell Ω of thickness t is defined by

$$\mathbf{x} = \mathbf{x}_\Gamma + \zeta \mathbf{n}_\Gamma \tag{23}$$

with ζ being the thickness parameter and $|\zeta| \leq t/2$. Alternatively, if the middle surface is defined implicitly with a signed distance function $\phi(\mathbf{x})$ the domain of the shell Ω is defined by

$$\Omega = \left\{ \mathbf{x} \in \mathbb{R}^3 : |\phi(\mathbf{x})| \leq \frac{t}{2} \right\}. \tag{24}$$

In this case the middle surface Γ is the zero-isosurface of $\phi(\mathbf{x})$, see Eq. (2). The displacement field \mathbf{u}_Ω of a point $P(\mathbf{x}_\Gamma, \zeta)$ in the shell continuum Ω takes the form

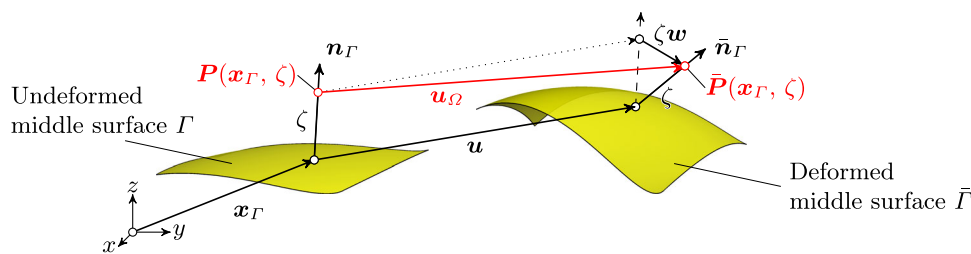
$$\mathbf{u}_\Omega(\mathbf{x}_\Gamma, \zeta) = \mathbf{u}(\mathbf{x}_\Gamma) + \zeta \mathbf{w}(\mathbf{x}_\Gamma) \tag{25}$$

with $\mathbf{u}(\mathbf{x}_\Gamma) = [u, v, w]^\top$ being the displacement field of the middle surface and $\mathbf{w}(\mathbf{x}_\Gamma)$ being the difference vector, as illustrated in Fig. 4.

Without transverse shear strains, the difference vector \mathbf{w} expressed in terms of TDC is defined as in [13]

$$\mathbf{w}(\mathbf{x}_\Gamma) = - \left[\nabla_\Gamma^{\text{dir}} \mathbf{u} + (\nabla_\Gamma^{\text{dir}} \mathbf{u})^\top \right] \cdot \mathbf{n}_\Gamma = \mathbf{H} \cdot \mathbf{u} - \nabla_\Gamma(\mathbf{u} \cdot \mathbf{n}_\Gamma). \tag{26}$$

Fig. 4 Displacements u_Ω , u and w of the shell



As readily seen in the equation above, the difference vector w is tangential. Alternatively, the difference vector w may also be re-written in terms of partial tangential derivatives of u and the normal vector n_Γ

$$w(x_\Gamma) = \mathbf{H} \cdot u - \nabla_\Gamma(u \cdot n_\Gamma) = - \begin{bmatrix} u_{,x}^{\text{dir}} \cdot n_\Gamma \\ u_{,y}^{\text{dir}} \cdot n_\Gamma \\ u_{,z}^{\text{dir}} \cdot n_\Gamma \end{bmatrix} \quad (27)$$

Consequently, the displacement field of the shell continuum is only a function of the middle surface displacement u , the unit normal vector n_Γ and the thickness parameter ζ .

The linearised, in-plane strain tensor ϵ_Γ is defined by the symmetric part of the directional gradient of the displacement field u_Ω , projected with \mathbf{P} [26]

$$\begin{aligned} \epsilon_\Gamma(x_\Gamma, \zeta) &= \mathbf{P} \cdot \frac{1}{2} \left[\nabla_\Gamma^{\text{dir}} u_\Omega + (\nabla_\Gamma^{\text{dir}} u_\Omega)^\top \right] \cdot \mathbf{P} \\ &= \mathbf{P} \cdot \epsilon_\Gamma^{\text{dir}} \cdot \mathbf{P} \\ &= \frac{1}{2} \left[\nabla_\Gamma^{\text{cov}} u_\Omega + (\nabla_\Gamma^{\text{cov}} u_\Omega)^\top \right]. \end{aligned} \quad (28)$$

Finally, the whole strain tensor may be split into a membrane and bending part, as usual in the classical theory

$$\epsilon_\Gamma = \epsilon_{\Gamma,M}(u) + \zeta \epsilon_{\Gamma,B}(w), \quad (29)$$

with

$$\begin{aligned} \epsilon_{\Gamma,M} &= \frac{1}{2} (\nabla_\Gamma^{\text{cov}} u + (\nabla_\Gamma^{\text{cov}} u)^\top), \\ \epsilon_{\Gamma,B} &= - \begin{bmatrix} u_{,xx}^{\text{cov}} \cdot n_\Gamma & u_{,yx}^{\text{cov}} \cdot n_\Gamma & u_{,zx}^{\text{cov}} \cdot n_\Gamma \\ \text{sym} & u_{,yy}^{\text{cov}} \cdot n_\Gamma & u_{,zy}^{\text{cov}} \cdot n_\Gamma \\ & & u_{,zz}^{\text{cov}} \cdot n_\Gamma \end{bmatrix}. \end{aligned}$$

Note that in the linearised bending strain tensor $\epsilon_{\Gamma,B}$, the term $(\nabla_\Gamma^{\text{dir}} u)^\top \cdot \mathbf{H}$ is neglected as in classical theory [45, Remark 2.2] or [49]. The resulting membrane and bending strain in Eq. (29) are equivalent compared to the classical theory, e.g., [1]. In the case of flat shell structures as considered in [27] the membrane strain is only a function of the tangential displacement $u_t = \mathbf{P} \cdot u$ and the bending strain only depends on the normal displacement $u_n = u \cdot n_\Gamma$, which simplifies the whole kinematic significantly. Moreover, the normal vector n_Γ is then constant and the difference vector simplifies to $w(x_\Gamma) = -\nabla_\Gamma u_n$.

3.2 Constitutive equation

As already mentioned above, the shell is assumed to be linear elastic and, as usual for thin structures, plane stress is presumed. The in-plane stress tensor σ_Γ is defined as

$$\sigma_\Gamma(x_\Gamma, \zeta) = \mathbf{P} \cdot [2\mu \epsilon_\Gamma + \lambda \text{tr}(\epsilon_\Gamma) \mathbb{I}] \cdot \mathbf{P} \quad (30)$$

$$= \mathbf{P} \cdot [2\mu \epsilon_\Gamma^{\text{dir}} + \lambda \text{tr}(\epsilon_\Gamma^{\text{dir}}) \mathbb{I}] \cdot \mathbf{P} \quad (31)$$

where $\mu = \frac{E}{2(1+\nu)}$ and $\lambda = \frac{E\nu}{(1-\nu^2)}$ are the Lamé constants and $\epsilon_\Gamma^{\text{dir}}$ is the directional strain tensor from Eq. (28). With this identity the in-plane stress tensor can be computed only with the directional strain tensor

$$\epsilon_\Gamma^{\text{dir}} = \epsilon_{\Gamma,M}^{\text{dir}}(u) + \zeta \epsilon_{\Gamma,B}^{\text{dir}}(w),$$

with

$$\begin{aligned} \epsilon_{\Gamma,M}^{\text{dir}} &= \frac{1}{2} (\nabla_\Gamma^{\text{dir}} u + (\nabla_\Gamma^{\text{dir}} u)^\top), \\ \epsilon_{\Gamma,B}^{\text{dir}} &= - \begin{bmatrix} u_{,xx}^{\text{dir}} \cdot n_\Gamma & \frac{1}{2} (u_{,yx}^{\text{dir}} + u_{,xy}^{\text{dir}}) \cdot n_\Gamma & \frac{1}{2} (u_{,zx}^{\text{dir}} + u_{,xz}^{\text{dir}}) \cdot n_\Gamma \\ \text{sym} & u_{,yy}^{\text{dir}} \cdot n_\Gamma & \frac{1}{2} (u_{,zy}^{\text{dir}} + u_{,yz}^{\text{dir}}) \cdot n_\Gamma \\ & & u_{,zz}^{\text{dir}} \cdot n_\Gamma \end{bmatrix}, \end{aligned}$$

which is from an implementational point of view an advantage, because covariant derivatives are not needed explicitly. In comparison to the classical theory, the in-plane stress tensor expressed in terms of TDC does not require the computation of the metric coefficients in the material law. Therefore, the resulting stress tensor does not hinge on a parametrization of the middle surface and shell analysis on implicitly defined surfaces is enabled.

3.2.1 Stress resultants

The stress tensor is only a function of the middle surface displacement vector u , the difference vector $w(u)$ and the thickness parameter ζ . This enables an analytical pre-integration with respect to the thickness and stress resultants can be identified. The following quantities are equivalent to the stress resultants in the classical theory [1,45], but they are expressed in terms of the TDC using a global Cartesian coordinate system.

The symmetric moment tensor \mathbf{m}_Γ is defined as

$$\begin{aligned} \mathbf{m}_\Gamma &= \int_{-t/2}^{t/2} \zeta \boldsymbol{\sigma}_\Gamma(\mathbf{u}, \zeta) \, d\zeta = \frac{t^3}{12} \boldsymbol{\sigma}_\Gamma(\boldsymbol{\varepsilon}_{\Gamma,B}) \\ &= \mathbf{P} \cdot \mathbf{m}_\Gamma^{\text{dir}} \cdot \mathbf{P}, \end{aligned} \tag{32}$$

results in the components

$$\begin{aligned} \left[\mathbf{m}_\Gamma^{\text{dir}} \right]_{11} &= -D_B (u_{,xx}^{\text{dir}} + \nu u_{,yy}^{\text{dir}} + \nu u_{,zz}^{\text{dir}}) \cdot \mathbf{n}_\Gamma, \\ \left[\mathbf{m}_\Gamma^{\text{dir}} \right]_{22} &= -D_B (u_{,yy}^{\text{dir}} + \nu u_{,xx}^{\text{dir}} + \nu u_{,zz}^{\text{dir}}) \cdot \mathbf{n}_\Gamma, \\ \left[\mathbf{m}_\Gamma^{\text{dir}} \right]_{33} &= -D_B (u_{,zz}^{\text{dir}} + \nu u_{,xx}^{\text{dir}} + \nu u_{,yy}^{\text{dir}}) \cdot \mathbf{n}_\Gamma, \\ \left[\mathbf{m}_\Gamma^{\text{dir}} \right]_{12} &= -D_B \frac{1-\nu}{2} (u_{,yx}^{\text{dir}} + u_{,xy}^{\text{dir}}) \cdot \mathbf{n}_\Gamma, \\ \left[\mathbf{m}_\Gamma^{\text{dir}} \right]_{13} &= -D_B \frac{1-\nu}{2} (u_{,zx}^{\text{dir}} + u_{,xz}^{\text{dir}}) \cdot \mathbf{n}_\Gamma, \\ \left[\mathbf{m}_\Gamma^{\text{dir}} \right]_{23} &= -D_B \frac{1-\nu}{2} (u_{,zy}^{\text{dir}} + u_{,yz}^{\text{dir}}) \cdot \mathbf{n}_\Gamma, \end{aligned}$$

where $D_B = \frac{Et^3}{12(1-\nu^2)}$ is the flexural rigidity of the shell. The moment tensor \mathbf{m}_Γ is symmetric and an in-plane tensor. Therefore, one of the three eigenvalues is zero and the two non-zero eigenvalues of \mathbf{m}_Γ are the principal bending moments m_1 and m_2 . The principal moments are in agreement with the eigenvalues of the moment tensor in the classical setting, see [1]. For the effective normal force tensor $\tilde{\mathbf{n}}_\Gamma$ we have

$$\begin{aligned} \tilde{\mathbf{n}}_\Gamma &= \int_{-t/2}^{t/2} \boldsymbol{\sigma}_\Gamma(\mathbf{u}, \zeta) \, d\zeta = t \boldsymbol{\sigma}_\Gamma(\boldsymbol{\varepsilon}_{\Gamma,M}) \\ &= \mathbf{P} \cdot \mathbf{n}_\Gamma^{\text{dir}} \cdot \mathbf{P}, \end{aligned} \tag{33}$$

with the components

$$\begin{aligned} \left[\mathbf{n}_\Gamma^{\text{dir}} \right]_{11} &= D_M [u_{,x}^{\text{dir}} + \nu(v_{,y}^{\text{dir}} + w_{,z}^{\text{dir}})], \\ \left[\mathbf{n}_\Gamma^{\text{dir}} \right]_{22} &= D_M [v_{,y}^{\text{dir}} + \nu(u_{,x}^{\text{dir}} + w_{,z}^{\text{dir}})], \\ \left[\mathbf{n}_\Gamma^{\text{dir}} \right]_{33} &= D_M [w_{,z}^{\text{dir}} + \nu(u_{,x}^{\text{dir}} + v_{,y}^{\text{dir}})], \\ \left[\mathbf{n}_\Gamma^{\text{dir}} \right]_{12} &= D_M \left[\frac{1-\nu}{2} (u_{,y}^{\text{dir}} + v_{,x}^{\text{dir}}) \right], \\ \left[\mathbf{n}_\Gamma^{\text{dir}} \right]_{13} &= D_M \left[\frac{1-\nu}{2} (u_{,z}^{\text{dir}} + w_{,x}^{\text{dir}}) \right], \\ \left[\mathbf{n}_\Gamma^{\text{dir}} \right]_{23} &= D_M \left[\frac{1-\nu}{2} (v_{,z}^{\text{dir}} + w_{,y}^{\text{dir}}) \right], \end{aligned}$$

where $D_M = \frac{Et}{1-\nu^2}$. Similar to the moment tensor, the two non-zero eigenvalues of $\tilde{\mathbf{n}}_\Gamma$ are in agreement with the effective normal force tensor expressed in local coordinates. Note

that for curved shells this tensor is *not* the physical normal force tensor. This tensor only appears in the variational formulation, see Sect. 4. The physical normal force tensor $\mathbf{n}_\Gamma^{\text{real}}$ is defined by

$$\mathbf{n}_\Gamma^{\text{real}} = \tilde{\mathbf{n}}_\Gamma + \mathbf{H} \cdot \mathbf{m}_\Gamma \tag{34}$$

and is, in general, not symmetric and also has one zero eigenvalue. The occurrence of the zero eigenvalues in \mathbf{m}_Γ , $\tilde{\mathbf{n}}_\Gamma$ and $\mathbf{n}_\Gamma^{\text{real}}$ is due to fact that these tensors are in-plane tensors, i.e. $\mathbf{m}_\Gamma \cdot \mathbf{n}_\Gamma = \mathbf{n}_\Gamma^T \cdot \mathbf{m}_\Gamma = \mathbf{0}$. The normal vector \mathbf{n}_Γ is the corresponding eigenvector to the zero eigenvalue and the other two eigenvectors are tangential.

3.3 Equilibrium

Based on the stress resultants from above, one obtains the equilibrium for a curved shell in strong form as

$$\text{div}_\Gamma \mathbf{n}_\Gamma^{\text{real}} + \mathbf{n}_\Gamma \text{div}_\Gamma (\mathbf{P} \cdot \text{div}_\Gamma \mathbf{m}_\Gamma) + \mathbf{H} \cdot \text{div}_\Gamma \mathbf{m}_\Gamma = -\mathbf{f},$$

which converts to

$$\begin{aligned} \text{div}_\Gamma \tilde{\mathbf{n}}_\Gamma + \mathbf{n}_\Gamma \text{div}_\Gamma (\mathbf{P} \cdot \text{div}_\Gamma \mathbf{m}_\Gamma) + 2\mathbf{H} \cdot \text{div}_\Gamma \mathbf{m}_\Gamma \\ + [\partial x_i^T \mathbf{H}]_{jk} [\mathbf{m}_\Gamma]_{ki} = -\mathbf{f}, \end{aligned} \tag{35}$$

with \mathbf{f} being the load vector per area on the middle surface Γ . A summation over the indices $i, k = 1, 2, 3$ has to be performed. The obtained equilibrium does not rely on a parametrization of the middle surface but is, otherwise, equivalent to the equilibrium in local coordinates [1,49]. From this point of view, the reformulation of the linear Kirchhoff–Love shell equations in terms of the TDC may be seen as a generalization, because the requirement of a parametrized middle surface is circumvented. With boundary conditions, as shown in detail in Sect. 3.3.2, the complete fourth-order boundary value problem (BVP) is defined.

Based on the equilibrium in Eq. (35), the transverse shear force vector \mathbf{q} is defined as

$$\mathbf{q} = \mathbf{P} \cdot \text{div}_\Gamma \mathbf{m}_\Gamma. \tag{36}$$

Note that in the special case of *flat* Kirchhoff–Love structures embedded in \mathbb{R}^3 the divergence of an in-plane tensor is a tangential vector, as already mentioned in Sect. 2.1. Therefore, the definition of the transverse shear force vector in [27] is in agreement with the obtained transverse shear force vector herein.

3.3.1 Equilibrium in weak form

The equilibrium in strong form is converted to a weak form by multiplying Eq. (35) with a suitable test function \mathbf{v} and integrating over the domain, leading to

$$\begin{aligned}
 & - \int_{\Gamma} \mathbf{v} \cdot \{ \operatorname{div}_{\Gamma} \tilde{\mathbf{n}}_{\Gamma} + \mathbf{n}_{\Gamma} \operatorname{div}_{\Gamma} (\mathbf{P} \cdot \operatorname{div}_{\Gamma} \mathbf{m}_{\Gamma}) + 2\mathbf{H} \cdot \operatorname{div}_{\Gamma} \mathbf{m}_{\Gamma} \\
 & + [\partial x_i^{\Gamma} \mathbf{H}]_{jk} [\mathbf{m}_{\Gamma}]_{ki} \} d\Gamma = \int_{\Gamma} \mathbf{v} \cdot \mathbf{f} d\Gamma. \tag{37}
 \end{aligned}$$

With Green’s formula from Sect. 2.1, we introduce the continuous weak form of the equilibrium:

Find $\mathbf{u} \in \mathcal{V} : \Gamma \rightarrow \mathbb{R}^3$ such that

$$a(\mathbf{u}, \mathbf{v}) = \langle \mathbf{F}, \mathbf{v} \rangle \quad \forall \mathbf{v} \in \mathcal{V}_0, \tag{38}$$

with

$$\begin{aligned}
 a(\mathbf{u}, \mathbf{v}) &= \int_{\Gamma} \nabla_{\Gamma}^{\operatorname{dir}} \mathbf{v} : \tilde{\mathbf{n}}_{\Gamma} - \boldsymbol{\varepsilon}_{\Gamma, \text{B}}^{\operatorname{dir}}(\mathbf{v}) : \mathbf{m}_{\Gamma} d\Gamma, \\
 \langle \mathbf{F}, \mathbf{v} \rangle &= \int_{\Gamma} \mathbf{f} \cdot \mathbf{v} d\Gamma - \int_{\partial \Gamma_{\text{N}}} \nabla_{\Gamma}^{\operatorname{dir}}(\mathbf{v} \cdot \mathbf{n}_{\Gamma}) \cdot (\mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) \\
 & - 2(\mathbf{H} \cdot \mathbf{v}) \cdot (\mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) - \mathbf{v} \cdot (\tilde{\mathbf{n}}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) \\
 & - (\mathbf{v} \cdot \mathbf{n}_{\Gamma}) (\mathbf{P} \cdot \operatorname{div}_{\Gamma} \mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) ds.
 \end{aligned}$$

The corresponding function spaces are

$$\mathcal{V} = \{ \mathbf{u} : \Gamma \rightarrow \mathbb{R}^3 \mid \mathbf{u} \in \mathcal{H}^1(\Gamma)^3 : \mathbf{u}_{,ji} \cdot \mathbf{n}_{\Gamma} \in \mathcal{L}^2(\Gamma)^3 \} \tag{39}$$

$$\mathcal{V}_0 = \{ \mathbf{v} \in \mathcal{V}(\Gamma) : \mathbf{v}|_{\partial \Gamma_{\text{D}}} = \mathbf{0} \} \tag{40}$$

where $\partial \Gamma_{\text{D}}$ is the Dirichlet boundary and $\partial \Gamma_{\text{N}}$ is the Neumann boundary. The advantage of this procedure is that the boundary terms naturally occur and directly allow to consider for mechanically meaningful boundary conditions.

3.3.2 Boundary conditions

As well known in the classical Kirchhoff–Love shell theory, special attention needs to be paid to the boundary conditions. In the following, the boundary terms of the weak form in Eq. (38) are rearranged in order to derive the effective boundary forces.

Using Eqs. (34) and (26), we have

$$\begin{aligned}
 & - \int_{\partial \Gamma_{\text{N}}} \nabla_{\Gamma}^{\operatorname{dir}}(\mathbf{v} \cdot \mathbf{n}_{\Gamma}) \cdot (\mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) - 2(\mathbf{H} \cdot \mathbf{v}) \cdot (\mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) \\
 & - \mathbf{v} \cdot (\tilde{\mathbf{n}}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) - (\mathbf{v} \cdot \mathbf{n}_{\Gamma}) (\mathbf{P} \cdot \operatorname{div}_{\Gamma} \mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) ds \\
 & = \int_{\partial \Gamma_{\text{N}}} \mathbf{v} \cdot (\mathbf{n}_{\Gamma}^{\operatorname{real}} \cdot \mathbf{n}_{\partial \Gamma}) + \mathbf{w}(\mathbf{v}) \cdot (\mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) \\
 & + (\mathbf{v} \cdot \mathbf{n}_{\Gamma}) \cdot (\mathbf{P} \cdot \operatorname{div}_{\Gamma} \mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) ds. \tag{41}
 \end{aligned}$$

As already mentioned above, the difference vector \mathbf{w} is a tangential vector. Consequently, the difference vector at the boundary may be expressed in terms of the tangential vectors $\mathbf{t}_{\partial \Gamma}$ and $\mathbf{n}_{\partial \Gamma}$

$$\begin{aligned}
 \mathbf{w}(\mathbf{v}) &= \underbrace{[\mathbf{H} \cdot \mathbf{v} - \nabla_{\Gamma}(\mathbf{v} \cdot \mathbf{n}_{\Gamma})] \cdot \mathbf{n}_{\partial \Gamma}}_{\boldsymbol{\omega}_{\mathbf{t}_{\partial \Gamma}}} \mathbf{t}_{\partial \Gamma} \\
 & + \underbrace{[\mathbf{H} \cdot \mathbf{v} - \nabla_{\Gamma}(\mathbf{v} \cdot \mathbf{n}_{\Gamma})] \cdot \mathbf{t}_{\partial \Gamma}}_{\boldsymbol{\omega}_{\mathbf{n}_{\partial \Gamma}}} \mathbf{n}_{\partial \Gamma} \tag{42}
 \end{aligned}$$

where $\boldsymbol{\omega}_{\mathbf{t}_{\partial \Gamma}} = \boldsymbol{\omega}_{\mathbf{t}_{\partial \Gamma}} \mathbf{t}_{\partial \Gamma}$ may be interpreted as rotation along the boundary and $\boldsymbol{\omega}_{\mathbf{n}_{\partial \Gamma}} = \boldsymbol{\omega}_{\mathbf{n}_{\partial \Gamma}} \mathbf{n}_{\partial \Gamma}$ is the rotation in co-normal direction, when the test function \mathbf{v} is interpreted as a displacement, see Fig. 5a. Analogously to the difference vector, the expressions $\mathbf{n}_{\Gamma}^{\operatorname{real}} \cdot \mathbf{n}_{\partial \Gamma}$ and $\mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}$ in Eq. (41) are decomposed in a similar manner

$$\begin{aligned}
 \mathbf{n}_{\Gamma}^{\operatorname{real}} \cdot \mathbf{n}_{\partial \Gamma} &= \underbrace{(\mathbf{n}_{\Gamma}^{\operatorname{real}} \cdot \mathbf{n}_{\partial \Gamma}) \cdot \mathbf{t}_{\partial \Gamma}}_{p_{\mathbf{t}_{\partial \Gamma}}} \mathbf{t}_{\partial \Gamma} \\
 & + \underbrace{(\mathbf{n}_{\Gamma}^{\operatorname{real}} \cdot \mathbf{n}_{\partial \Gamma}) \cdot \mathbf{n}_{\partial \Gamma}}_{p_{\mathbf{n}_{\partial \Gamma}}} \mathbf{n}_{\partial \Gamma}, \tag{43}
 \end{aligned}$$

$$\begin{aligned}
 \mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma} &= \underbrace{(\mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) \cdot \mathbf{n}_{\partial \Gamma}}_{m_{\mathbf{t}_{\partial \Gamma}}} \mathbf{t}_{\partial \Gamma} \\
 & + \underbrace{(\mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}) \cdot \mathbf{t}_{\partial \Gamma}}_{m_{\mathbf{n}_{\partial \Gamma}}} \mathbf{n}_{\partial \Gamma}. \tag{44}
 \end{aligned}$$

Next, the term $p_{\mathbf{n}_{\Gamma}} = \mathbf{P} \cdot \operatorname{div}_{\Gamma} \mathbf{m}_{\Gamma} \cdot \mathbf{n}_{\partial \Gamma}$ represents the resultant force in normal direction. In Fig. 5b the forces and bending moments along a curved boundary are illustrated.

Inserting these expressions in Eq. (41), the integral along the Neumann boundary simplifies to

$$\begin{aligned}
 & \int_{\partial \Gamma_{\text{N}}} \mathbf{v} \cdot (p_{\mathbf{t}_{\partial \Gamma}} \mathbf{t}_{\partial \Gamma} + p_{\mathbf{n}_{\partial \Gamma}} \mathbf{n}_{\partial \Gamma} + p_{\mathbf{n}_{\Gamma}} \mathbf{n}_{\Gamma}) \\
 & + \boldsymbol{\omega}_{\mathbf{t}_{\partial \Gamma}} m_{\mathbf{t}_{\partial \Gamma}} + \boldsymbol{\omega}_{\mathbf{n}_{\partial \Gamma}} m_{\mathbf{n}_{\partial \Gamma}} ds. \tag{45}
 \end{aligned}$$

As discussed in detail, e.g., in [1], the rotation in co-normal direction $\boldsymbol{\omega}_{\mathbf{n}_{\partial \Gamma}}$ is already prescribed with $\mathbf{v}|_{\partial \Gamma}$. Therefore, the term $\boldsymbol{\omega}_{\mathbf{n}_{\partial \Gamma}} m_{\mathbf{n}_{\partial \Gamma}}$ is expanded and with integration by parts we obtain

$$\begin{aligned}
 \int_{\partial \Gamma_{\text{N}}} \boldsymbol{\omega}_{\mathbf{n}_{\partial \Gamma}} m_{\mathbf{n}_{\partial \Gamma}} ds &= \int_{\partial \Gamma_{\text{N}}} -\nabla_{\Gamma}(\mathbf{v} \cdot \mathbf{n}_{\Gamma}) \cdot \mathbf{t}_{\partial \Gamma} m_{\mathbf{n}_{\partial \Gamma}} \\
 & + \mathbf{H} \cdot \mathbf{v} \cdot \mathbf{t}_{\partial \Gamma} m_{\mathbf{n}_{\partial \Gamma}} ds \\
 & = \int_{\partial \Gamma_{\text{N}}} (\mathbf{v} \cdot \mathbf{n}_{\Gamma}) \cdot (\nabla_{\Gamma} m_{\mathbf{n}_{\partial \Gamma}} \cdot \mathbf{t}_{\partial \Gamma}) \\
 & + \mathbf{H} \cdot \mathbf{v} \cdot \mathbf{t}_{\partial \Gamma} m_{\mathbf{n}_{\partial \Gamma}} ds \\
 & - (\mathbf{v} \cdot \mathbf{n}_{\Gamma}) m_{\mathbf{n}_{\partial \Gamma}} \Big|_{+C}^{-C} \tag{46}
 \end{aligned}$$

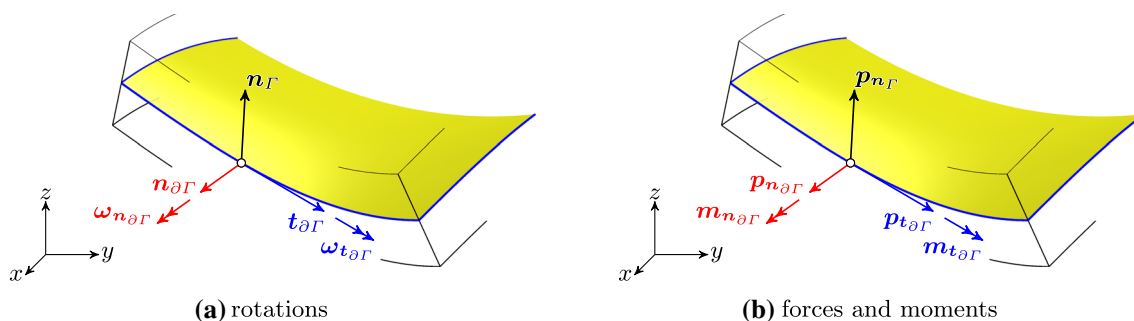


Fig. 5 Decomposition of the difference vector w , in-plane normal forces $\mathbf{n}_\Gamma^{\text{real}} \cdot \mathbf{n}_{\partial\Gamma}$ and bending moments $\mathbf{m}_\Gamma \cdot \mathbf{n}_{\partial\Gamma}$ along the boundary $\partial\Gamma$ in terms of $\mathbf{t}_{\partial\Gamma}$ and $\mathbf{n}_{\partial\Gamma}$: **a** rotations at the boundary, **b** normal force tensor and bending moments at the boundary

where $+C$ and $-C$ are points close at a corner C . The new boundary term are the Kirchhoff forces or corner forces. Note that if the boundary of the shell is smooth, the corner forces vanish. Finally, the integral over the Neumann boundary in Eq. (38) is expressed in terms of the well-known effective boundary forces and the bending moment along the boundary

$$\int_{\partial\Gamma_N} \mathbf{v} \cdot (\tilde{p}_{\mathbf{t}_{\partial\Gamma}} \mathbf{t}_{\partial\Gamma} + \tilde{p}_{\mathbf{n}_{\partial\Gamma}} \mathbf{n}_{\partial\Gamma} + \tilde{p}_{\mathbf{n}_\Gamma} \mathbf{n}_\Gamma) + \omega_{\mathbf{t}_{\partial\Gamma}} m_{\mathbf{t}_{\partial\Gamma}} ds - (\mathbf{v} \cdot \mathbf{n}_\Gamma) m_{\mathbf{n}_{\partial\Gamma}} \Big|_{+C}^{-C} \tag{47}$$

with:

$$\tilde{p}_{\mathbf{t}_{\partial\Gamma}} = p_{\mathbf{t}_{\partial\Gamma}} + (\mathbf{H} \cdot \mathbf{t}_{\partial\Gamma}) \cdot \mathbf{t}_{\partial\Gamma} m_{\mathbf{n}_{\partial\Gamma}}, \tag{48}$$

$$\tilde{p}_{\mathbf{n}_{\partial\Gamma}} = p_{\mathbf{n}_{\partial\Gamma}} + (\mathbf{H} \cdot \mathbf{t}_{\partial\Gamma}) \cdot \mathbf{n}_{\partial\Gamma} m_{\mathbf{n}_{\partial\Gamma}}, \tag{49}$$

$$\tilde{p}_{\mathbf{n}_\Gamma} = p_{\mathbf{n}_\Gamma} + \nabla_\Gamma m_{\mathbf{n}_{\partial\Gamma}} \cdot \mathbf{t}_{\partial\Gamma}. \tag{50}$$

The obtained effective boundary forces and moments are in agreement with the given quantities in local coordinates [1,49]. The prescribeable boundary conditions are the conjugated displacements and rotations to the effective forces and moments at the boundary

$$\begin{aligned} \tilde{p}_{\mathbf{t}_{\partial\Gamma}} &\iff \mathbf{u} \cdot \mathbf{t}_{\partial\Gamma} = u_{\mathbf{t}_{\partial\Gamma}}, \\ \tilde{p}_{\mathbf{n}_{\partial\Gamma}} &\iff \mathbf{u} \cdot \mathbf{n}_{\partial\Gamma} = u_{\mathbf{n}_{\partial\Gamma}}, \\ \tilde{p}_{\mathbf{n}_\Gamma} &\iff \mathbf{u} \cdot \mathbf{n}_\Gamma = u_{\mathbf{n}_\Gamma}, \\ m_{\mathbf{t}_{\partial\Gamma}} &\iff \omega_{\mathbf{t}_{\partial\Gamma}} = [\mathbf{H} \cdot \mathbf{u} - \nabla_\Gamma(\mathbf{u} \cdot \mathbf{n}_\Gamma)] \cdot \mathbf{n}_{\partial\Gamma}, \\ &= - \left[(\nabla_\Gamma^{\text{dir}} \mathbf{u})^\top \cdot \mathbf{n}_\Gamma \right] \cdot \mathbf{n}_{\partial\Gamma}. \end{aligned}$$

In Table 1, common support types are given. Other boundary conditions (e.g., membrane support, etc.) may be derived, with the quantities above, accordingly.

4 Implementational aspects

The continuous weak form is discretized using isogeometric analysis as proposed by Hughes et al. [10,30]. The NURBS

patch T is the middle surface of the shell and the elements τ_i ($i = 1, \dots, n_{\text{Elem}}$) are defined by the knot spans of the patch. The mesh is then defined by the union of the elements $\Gamma = \bigcup_{\tau \in T} \tau$.

There is a fixed set of local basis functions $\{N_i^k(\mathbf{r})\}$ of order k with $i = 1, \dots, n_k$ being the number of control points and the displacements $\{\hat{u}_i, \hat{v}_i, \hat{w}_i\}$ stored at the control points i are the degrees of freedom. Using the isoparametric concept, the shape functions $N_i^k(\mathbf{r})$ are NURBS of order k . The surface derivatives of the shape functions are computed as defined in Sect. 2, similar as in the Surface FEM [16,18,20,22] using NURBS instead of Lagrange polynomials as ansatz and test functions. The shape functions of order $k \geq 2$ are in the function space \mathcal{V} , see Eq. (39). In fact, the used shape functions are in the Sobolev space $\mathcal{H}^k(\Gamma)^3 \subset \mathcal{V}$ iff $k \geq 2$.

The resulting element stiffness matrix \mathbf{K}_{Elem} is a 3×3 block matrix and is divided into a membrane and bending part

$$\mathbf{K}_{\text{Elem}} = \mathbf{K}_{\text{Elem,M}} + \mathbf{K}_{\text{Elem,B}}. \tag{51}$$

The membrane part is defined by

$$\begin{aligned} \mathbf{K}_{\text{Elem,M}} &= t \int_\Gamma P_{ib} \cdot [\hat{\mathbf{K}}]_{bj} d\Gamma \\ [\hat{\mathbf{K}}]_{kj} &= \mu(\delta_{kj} N_{,a}^T \cdot N_{,a}^T + N_{,j}^T \cdot N_{,k}^T) + \lambda N_{,k}^T \cdot N_{,j}^T, \end{aligned} \tag{52}$$

summation over a and b . The matrix $\hat{\mathbf{K}}$ is determined by directional first-order derivatives of the shape functions N . One may recognize that the structure of the matrix $\hat{\mathbf{K}}$ is similar to the stiffness matrix of 3D linear elasticity problems. For the bending part we have

$$[\mathbf{K}_{\text{Elem,B}}]_{ij} = D_B \int_\Gamma n_i n_j \tilde{\mathbf{K}} d\Gamma \tag{54}$$

$$\tilde{\mathbf{K}} = (1 - \nu) N_{,ab}^{\text{cov}} \cdot N_{,ab}^{\text{cov}\top} + \nu N_{,cc}^{\text{cov}} \cdot N_{,dd}^{\text{cov}\top}. \tag{55}$$

Table 1 Set of common boundary conditions

Clamped edge	$u_{t_{\partial\Gamma}} = 0$	$u_{n_{\partial\Gamma}} = 0$	$u_{n_{\Gamma}} = 0$	$\omega_{t_{\partial\Gamma}} = 0$
Simply supported edge	$u_{t_{\partial\Gamma}} = 0$	$u_{n_{\partial\Gamma}} = 0$	$u_{n_{\Gamma}} = 0$	$m_{t_{\partial\Gamma}} = 0$
Symmetry support	$\tilde{p}_{t_{\partial\Gamma}} = 0$	$u_{n_{\partial\Gamma}} = 0$	$\tilde{p}_{n_{\Gamma}} = 0$	$\omega_{t_{\partial\Gamma}} = 0$
Free edge	$\tilde{p}_{t_{\partial\Gamma}} = 0$	$\tilde{p}_{n_{\partial\Gamma}} = 0$	$\tilde{p}_{n_{\Gamma}} = 0$	$m_{t_{\partial\Gamma}} = 0$

A summation over a, b on the one hand and c, d on the other has to be performed. The first term of $\tilde{\mathbf{K}}$ is the contraction of the covariant Hessian matrix $\mathbf{H}\mathbf{e}_F^{\text{cov}}$ and the second term may be identified as the Bi-Laplace operator. Note that for the Bi-Laplace operator also directional derivatives may be used, due to the fact that the trace of second order derivatives is invariant, although the components differ. This suggests a further rearrangement of the contraction of the covariant Hessian matrix in order to use only directional derivatives, which is preferred from an implementational point of view. The equivalent expression of $\tilde{\mathbf{K}}$ using only second-order directional derivatives is

$$\tilde{\mathbf{K}} = (1 - \nu) P_{ea} N_{,ab}^{\text{dir}} \cdot N_{,be}^{\text{dir}\text{T}} + \nu N_{,cc}^{\text{dir}} \cdot N_{,dd}^{\text{dir}\text{T}}, \tag{56}$$

with summation over a, b, e and c, d as above. When the shell is given through a parametrization, the resulting element stiffness matrix in the classical theory, e.g., [9] is equivalent to the element stiffness matrix from above, but in the classical setting the computation may be found more cumbersome due to fact that the local basis vectors and the metric tensor in co- and contra-variant form has to be computed. In contrast, herein, the surface gradients and second-order derivatives are first applied to the shape functions (NURBS or classical finite element functions) to obtain $N_{,i}^{\Gamma}$, $N_{,ij}^{\text{dir}}$ and $N_{,ij}^{\text{cov}}$, which is independent of the application.

In this sense, a significant part of the complexity of implementing shells is shifted to finite element technology and may be recycled for any kind of boundary value problems on curved surfaces in \mathbb{R}^3 . Examples are transport problems [16–18] or flow problems [20,31] on curved surfaces. We expect that future implementations in finite element software will provide frameworks for solving PDEs on manifolds and, based, e.g., on this work will also apply to shells. In order to emphasize the differences in the implementation, example Matlab®-codes for the proposed TDC-based formulation and the classical parametrization-based formulation are given in Sect. A, clearly highlighting the differences.

The boundary conditions are weakly enforced by Lagrange multipliers [51]. The shape functions of the Lagrange multipliers are NURBS of the same order than the shape functions of the displacements. Therefore, the shape functions of the Lagrange multiplier i is defined as

$$\{N_{iL}^k(\mathbf{r})\} = \{N_i^k(\mathbf{r})|_{\partial\Gamma_D}\}. \tag{57}$$

For the test cases shown in Sect. 5, bounded condition numbers and unique solutions are observed. The usual assembly yields a linear system of equations in the form

$$\begin{bmatrix} \mathbf{K} & \mathbf{C} \\ \mathbf{C}^{\text{T}} & \mathbf{0} \end{bmatrix} \cdot \begin{bmatrix} \hat{\mathbf{u}} \\ \hat{\boldsymbol{\lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix}, \tag{58}$$

with $[\hat{\mathbf{u}}, \hat{\boldsymbol{\lambda}}]^{\text{T}} = [\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{w}}, \hat{\boldsymbol{\lambda}}]$ being the sought displacements of the control points and Lagrange multipliers. With the shape functions of the Lagrange multipliers N_L , the constraint matrix \mathbf{C} for simply supported edges is defined by

$$\mathbf{C} = \int_{\partial\Gamma_D} \begin{bmatrix} N \cdot N_L^{\text{T}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & N \cdot N_L^{\text{T}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & N \cdot N_L^{\text{T}} \end{bmatrix} ds, \tag{59}$$

for clamped edges

$$\mathbf{C} = \int_{\partial\Gamma_D} \begin{bmatrix} N \cdot N_L^{\text{T}} & \mathbf{0} & \mathbf{0} & (n_x n_{\partial\Gamma_i} N_{,i}^{\Gamma}) \cdot N_L^{\text{T}} \\ \mathbf{0} & N \cdot N_L^{\text{T}} & \mathbf{0} & (n_y n_{\partial\Gamma_i} N_{,i}^{\Gamma}) \cdot N_L^{\text{T}} \\ \mathbf{0} & \mathbf{0} & N \cdot N_L^{\text{T}} & (n_z n_{\partial\Gamma_i} N_{,i}^{\Gamma}) \cdot N_L^{\text{T}} \end{bmatrix} ds, \tag{60}$$

and for symmetry supports

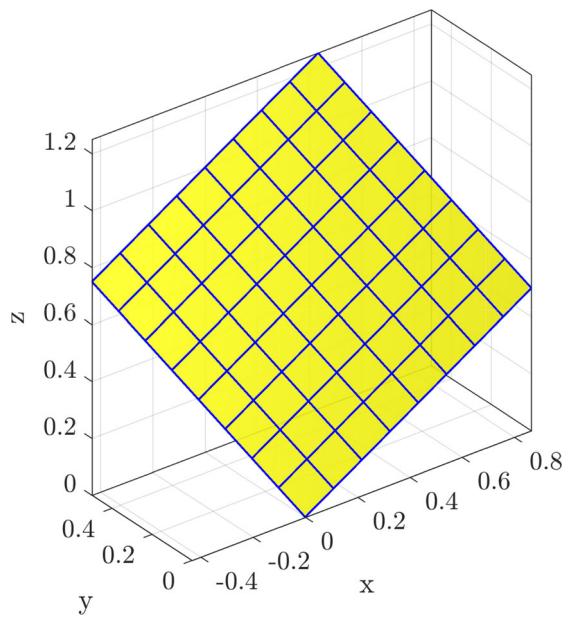
$$\mathbf{C} = \int_{\partial\Gamma_D} \begin{bmatrix} n_{\partial\Gamma_x} N \cdot N_L^{\text{T}} & (n_x n_{\partial\Gamma_i} N_{,i}^{\Gamma}) \cdot N_L^{\text{T}} \\ n_{\partial\Gamma_y} N \cdot N_L^{\text{T}} & (n_y n_{\partial\Gamma_i} N_{,i}^{\Gamma}) \cdot N_L^{\text{T}} \\ n_{\partial\Gamma_z} N \cdot N_L^{\text{T}} & (n_z n_{\partial\Gamma_i} N_{,i}^{\Gamma}) \cdot N_L^{\text{T}} \end{bmatrix} ds. \tag{61}$$

Note that all constraint matrices have three block-rows referring to the unknowns $\hat{\mathbf{u}}, \hat{\mathbf{v}}, \hat{\mathbf{w}}$.

5 Numerical results

The numerical results are achieved using NURBS functions for the geometry and shape function definition, following the methodology of isogeometric analysis (IGA) [3,24,30,32,34]. The definition of NURBS is omitted here for brevity but is found at numerous references in the frame of IGA, e.g., [10,40].

The obtained shell equations are carefully verified and compared to the classical approach with different test cases. As already mentioned above the proposed approach leads to an equivalent stiffness matrix for arbitrary curved and non-curved shells. Consequently, the same convergence



Geometry: Quadrilateral flat shell
 $L = 1$
 $t = 0.01$
 $\mathbf{n}_T = [-1/4, -\sqrt{3}/2, \sqrt{3}/4]^T$

Material parameters: $E = 10\,000$
 $\nu = 0.3$

Load: \mathbf{f}_t and f_n

Support: Simple support on all edges

Fig. 6 Definition of flat shell problem

properties as shown, e.g., in [9,32] are expected. In the following, the results of the convergence analyses of a flat shell embedded in \mathbb{R}^3 , the Scordelis-Lo roof, and the pinched cylinder test (part of the shell obstacle course proposed by Belytschko et al. [2]) are shown. Furthermore, a new test case with a challenging geometry is proposed which features smooth solutions enabling higher-order convergence rates. These rates are confirmed in the residual error as no analytic solution exists, see Sect. 5.4. Other examples (e.g., pinched hemispherical shell, shells of revolution, etc.) have been considered but are omitted here for brevity.

In the convergence studies, NURBS patches with different orders and numbers of knot spans in each direction are employed. This is equivalent to meshes with higher-order elements and $n = \{2, 4, 8, 16, 32\}$ elements per side are used. The orders are varied as $p = \{2, 3, 4, 5, 6\}$.

5.1 Flat shell embedded in \mathbb{R}^3

Following a similar rationale as in [27], as a first test case, we consider a simple quadrilateral, flat shell with the normal vector $\mathbf{n}_T = [-1/4, -\sqrt{3}/2, \sqrt{3}/4]^T$ in \mathbb{R}^3 , see Fig. 6. The shell is simply supported at all edges. For verification, the

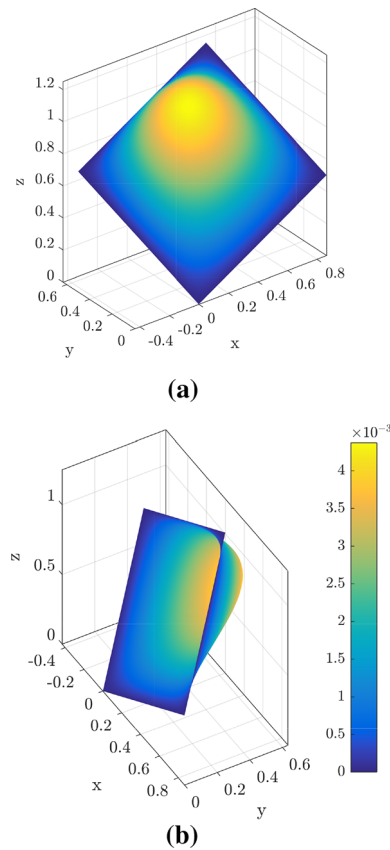


Fig. 7 Displacement \mathbf{u} of arbitrarily orientated flat shell, scaled by two orders of magnitude **a** front view, **b** rotated view

load vector \mathbf{f} is split into tangential \mathbf{f}_t and normal f_n loads. The tangential loads are obtained with the method of manufactured solution for a given displacement field $\mathbf{u}_t(\mathbf{x}) = [[1, 1]^T \cdot 1/4 \cdot \sin(\pi r) \sin(\pi s)] \circ \chi^{-1}$. In normal direction, a sinusoidal load $f_n(\mathbf{x}) = [-D_B \sin(\pi r) \sin(\pi s)] \circ \chi^{-1}$ is applied to the shell. Herein, χ is an affine mapping function (rigid-body rotation) from the horizontal parameter space to the real domain. An analytic solution for the normal displacements is easily obtained with $u_n(\mathbf{x}) = [-(\sin(\pi r) \sin(\pi s))/(4\pi^4)] \circ \chi^{-1}$, [46]. The shell is defined with $L = 1$ and the thickness is set to $t = 0.01$. The material parameters are: Young’s modulus $E = 10000$ and the Poisson’s ratio $\nu = 0.3$.

In Fig. 7, the solution of the shell is illustrated. The displacements are scaled by two orders of magnitude. The colours on the deformed surface indicate the Euclidean norm of the displacement field $\|\mathbf{u}\|$.

The results of the convergence analysis are shown in Fig. 8. The curves are plotted as a function of the element size $1/n$ (which is rather a characteristic length of the knot spans). The dotted lines indicate the theoretical optimal order of convergence. In Fig. 8a, the relative L_2 -error of the primal variable (displacements) is shown. Optimal higher-order convergence rates $\mathcal{O}(p + 1)$ are achieved. In the figures Fig. 8b–d, the

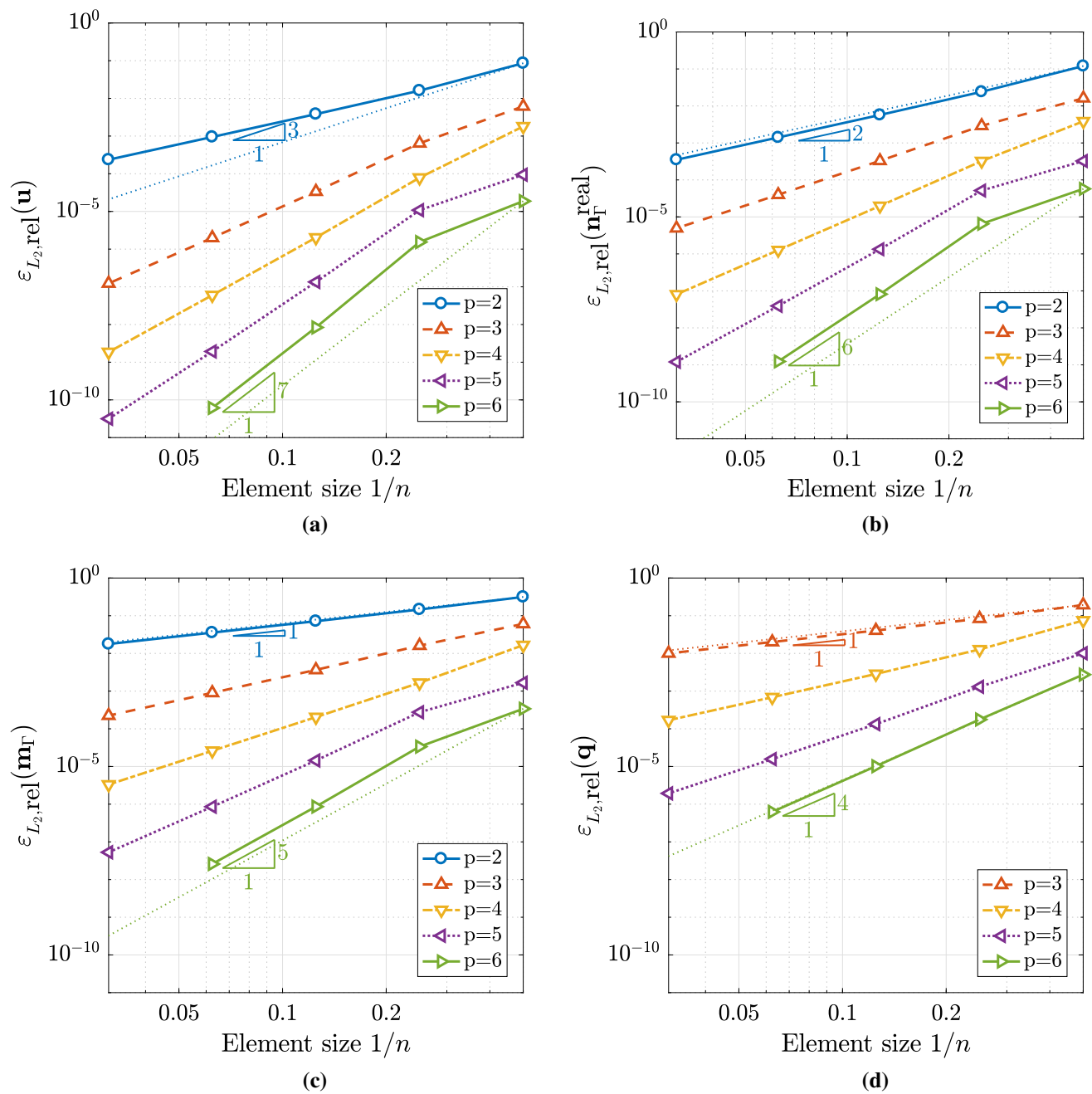
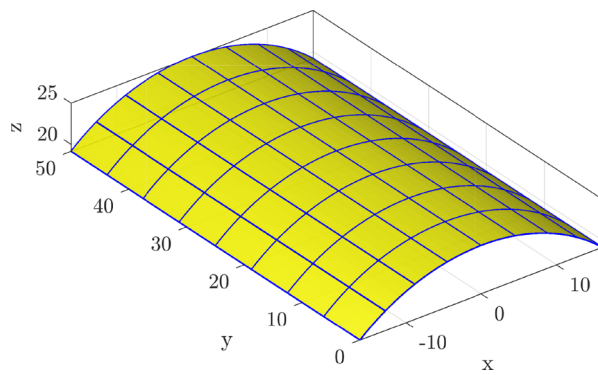


Fig. 8 Convergence results for the rotated flat shell. **a** Relative L_2 -norm of displacements \mathbf{u} . **b** Relative L_2 -norm of normal forces $\mathbf{n}_\Gamma^{\text{real}}$. **c** Relative L_2 -norm of bending moments \mathbf{m}_Γ . **d** Relative L_2 -norm of transverse shear forces \mathbf{q}

relative L_2 -errors of the normal forces (membrane forces), bending moments and transverse shear forces are plotted. For all stress resultants the theoretical optimal orders of convergence are achieved. It is clear that the same results were obtained if the results are computed for the purely two-dimensional case as, e.g., in [9].

5.2 Scordelis-Lo roof

The Scordelis-Lo roof is a cylindrical shell and is supported with two rigid diaphragms at the ends. The shell is loaded by gravity forces, see Fig. 9. The cylinder is defined with $L = 50$, $R = 25$ and the angle subtended by the roof is $\phi = 80^\circ$. The thickness of the shell is set to $t = 0.25$. The material parameters are: Young’s modulus $E = 4.32 \times 10^8$ and the Poisson’s ratio $\nu = 0.0$. In contrast to the first exam-



Geometry: Cylindrical shell
 $L = 50$
 $R = 25$
 $\phi = 80^\circ$
 $t = 0.25$

Material parameters: $E = 4.32 \times 10^8$
 $\nu = 0.0$

Load: Gravity load $\mathbf{f} = [0, 0, -90]^\top$

Support: Rigid diaphragms at its ends

Fig. 9 Definition of Scordelis-Lo roof problem

ple, the maximum vertical displacement $u_{z,\max}$ is compared with the reference solution $u_{z,\max,\text{Ref}} = 0.3024$ as given in reference [2].

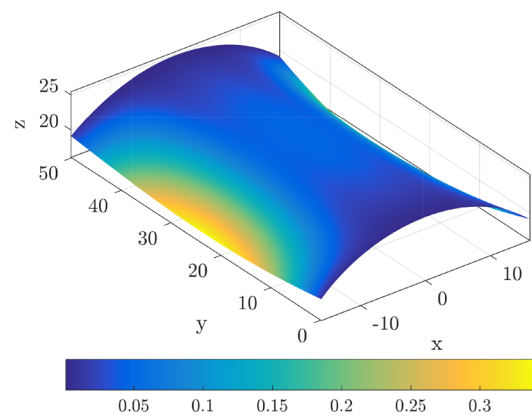
In Fig. 10a, the numerical solution of the Scordelis-Lo roof is illustrated. The displacements are magnified by one order of magnitude.

In Fig. 10b, the convergence of the maximum displacement $u_{z,\max}$ is plotted up to polynomial order of $p = 6$ as a function of the element (knot span) size. It is clearly seen that the expected results are achieved, with increasing accuracy for higher-order NURBS. Due to the lack of a more accurate reference solutions, it is not useful to show these results in a double-logarithmic diagram as usual for error plots. The style of presentation follows those of many other references such as, e.g., in [2,9,32].

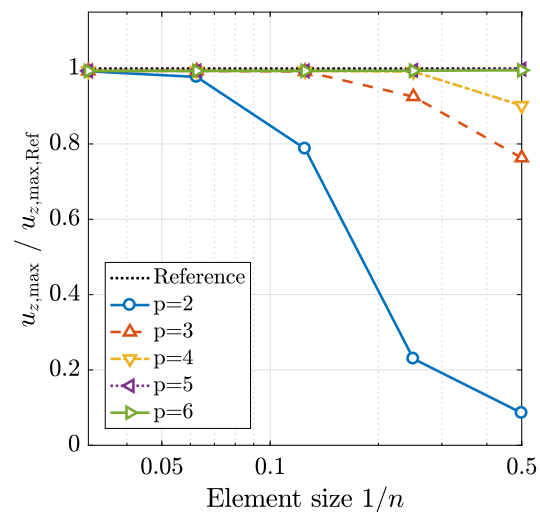
5.3 Pinched cylinder

The next test case is a cylindrical shell pinched with two diametrically opposite unit loads located within the middle of the shell, see Fig. 11. The cylinder is defined with $L = 600$, $R = 300$. The thickness is set to $t = 3$. The material properties are: Young's modulus $E = 3 \times 10^6$ and the Poisson's ratio $\nu = 0.3$. The reference displacement at the loading points are $u_{\text{Ref}} = 1.82488 \times 10^{-5}$ as given in reference [2]. Due to symmetry only one eighth of the geometry is modelled.

In Fig. 12a, the numerical solution of the pinched cylinder is illustrated with scaled displacements by a factor of 5×10^6 .



(a) displacement u



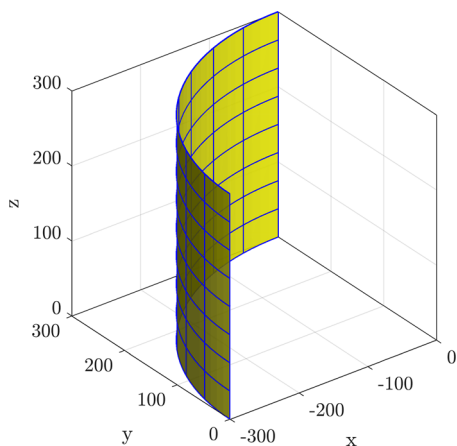
(b) convergence

Fig. 10 **a** Displacement field of the Lo-Scordelis roof scaled by one order of magnitude, **b** normalized convergence of reference displacement $u_{z,\max,\text{Ref}} = 0.3024$

As in the example before, in Fig. 12b, the convergence to a normalized reference displacement as a function of the element size is plotted. The results converge with the expected behaviour as shown in [9,32]. It is noted that due to the singularity in some mechanical quantities due to the single force, higher-order convergence rates are not possible here. However, the improvement for increasing the order of the NURBS is still seen in the figure. An additional grading of the elements in order to better resolve the singularity would have further improved the situation but is omitted here.

5.4 Flower shaped shell

As a last example, a more complex geometry is considered, which enables smooth mechanical fields and thereby enables higher-order convergence rates. The geometry of the middle surface is given with



Geometry: Cylinder (one eighth of cylinder modeled)
 $L/2 = 300$
 $R = 300$
 $\phi = 90^\circ$
 $t = 3$

Material parameters: $E = 3 \times 10^6$
 $\nu = 0.3$

Load: Single unit forces

Support: Rigid diaphragms at the top and symmetry boundary conditions

Fig. 11 Definition of the pinched cylinder problem

$$\mathbf{x}_\Gamma(r, s) = \begin{bmatrix} (A - C) \cos(\theta) \\ (A - C) \sin(\theta) \\ 1 - s^2 \end{bmatrix} \tag{62}$$

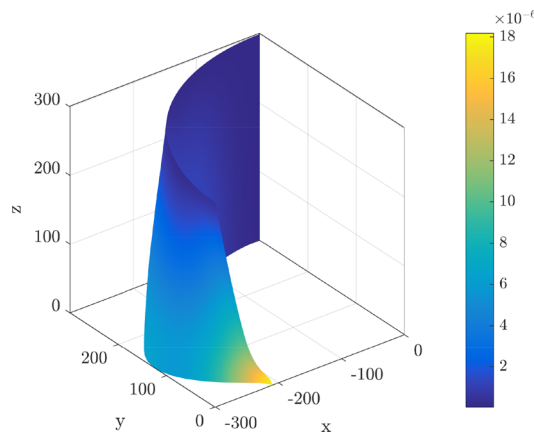
with:

$$\begin{aligned} r, s &\in [-1, 1], A = 2.3, B = 0.8 \\ \theta(r) &= \pi(r + 1) \\ C(r, s) &= s[B + 0.3 \cos(6\theta)] \end{aligned} \tag{63}$$

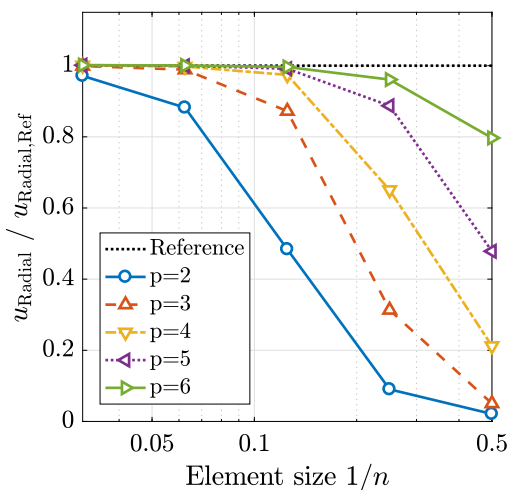
and illustrated in Fig. 13. Below the figure, the boundary conditions and material parameters are defined. The middle surface of the shell features varying principal curvatures and curved boundaries.

The curved boundaries are clamped and the corresponding conditions (from Table 1) have to be properly enforced. An analytical solution or reference displacement is not available. Therefore, the error is measured in the strong form of the equilibrium from Eq. (35) and may be called residual error. In particular, the residual error is the summed element-wise relative L_2 -error

$$\begin{aligned} \varepsilon_{\text{rel, residual}} &= \sum_{i=1}^{n_{\text{Elem}}} \varepsilon_{L_2, \text{rel}, \tau_i} \\ \varepsilon_{L_2, \text{rel}, \tau}^2 &= \frac{\int_\Gamma \left\{ \text{div}_\Gamma \tilde{\mathbf{n}}_\Gamma + \mathbf{n}_\Gamma \text{div}_\Gamma (\mathbf{P} \cdot \text{div}_\Gamma \mathbf{m}_\Gamma) + 2\mathbf{H} \cdot \text{div}_\Gamma \mathbf{m}_\Gamma + [\partial x_i^f \mathbf{H}]_{jk} [\mathbf{m}_\Gamma]_{ki} + \mathbf{f} \right\}^2 d\Gamma}{\int_\Gamma \mathbf{f}^2 d\Gamma} \end{aligned} \tag{64}$$



(a) displacements u

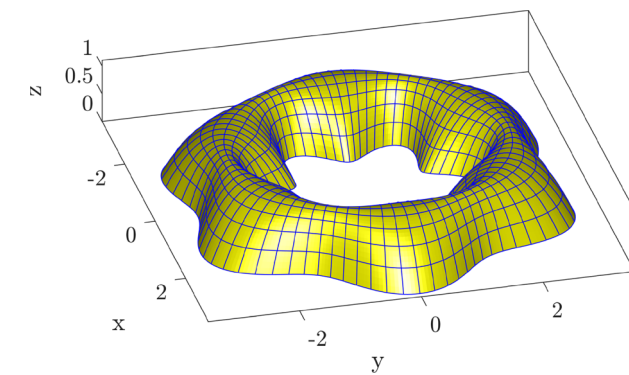


(b) convergence

Fig. 12 Pinched cylinder: **a** displacement u of one eighth of the geometry (scaled by a factor of 5×10^6), **b** normalized convergence of reference displacement $u_{\text{Radial, Ref}} = 1.82488 \times 10^{-5}$ at loading points

The computation of the residual error requires the evaluation of fourth-order surface derivatives. It is noteworthy that the implementation of these higher-order derivatives is not without efforts. For example, recall that mixed directional surface derivatives are not symmetric. That is, there are $3^4 = 81$ partial fourth-order derivatives. Nevertheless, if the displacement field is smooth enough this error measure is a suitable quantity for the convergence analysis.

In Fig. 14a, the deformed shell is illustrated. The displacement field is scaled by one order of magnitude. In Fig. 14b, the results of the convergence analysis are plotted. Due to the fact that fourth-order derivatives need to be computed, at least fourth-order shape functions are required. The theoretical optimal order of convergence is $\mathcal{O}(p - 3)$ if the solution is smooth enough. One may observe that higher-order convergence rates are achieved, however, rounding-off errors and the conditioning may slightly influence the convergence. Nevertheless, the results are excellent also given



Geometry: Flower shaped shell
see Eq. 63
 $t = 0.1$ m

Material parameters: $E = 1 \times 10^5$ kN m $^{-2}$
 $\nu = 0.3$

Load: $\mathbf{f}(\mathbf{x}_T) = [1, 2, -10]^T$ kN m $^{-2}$

Support: Clamped edges at inner and outer boundary

Fig. 13 Definition of flower shaped shell problem

the fact that higher-order accurate results for shells (given in double-logarithmic error plots) are the exception.

The stored elastic energy at the finest level with a polynomial order $p = 8$, which may be seen as an overkill solution, is $\epsilon = 1.7635958 \pm 1 \times 10^{-7}$ kN m. This stored elastic energy may be used for future benchmark tests, without the need to implement fourth-order derivatives on manifolds.

6 Conclusions and outlook

The linear Kirchhoff–Love shell theory is reformulated in terms of the TDC using a global Cartesian coordinate system and tensor notation. The resulting model equations apply to shell geometries which are parametrized or not. For example, a parametrization may not be available when shell geometries are implied by the level-set method. Because the TDC-based formulation holds in both cases, it may be seen as a generalization to the classical shell theory which is based on parametrizations and curvilinear coordinates.

The TDC-based strong form is used as the starting point to consistently obtain the weak form including all boundary terms well-known in the Kirchhoff–Love theory. Mechanical stress-resultants such as moments, normal and shear forces are defined in global coordinates. Furthermore, the strong form may be used in the numerical results to compute residual errors and thus enable convergence analyses even without the knowledge of exact solutions which, for shells, are scarce.

For the discretization, the Surface FEM is used with NURBS as trial and test functions. That is, an isogeometric approach is chosen due to continuity requirements. In this case, the presence of a surface mesh (i.e., a NURBS patch),

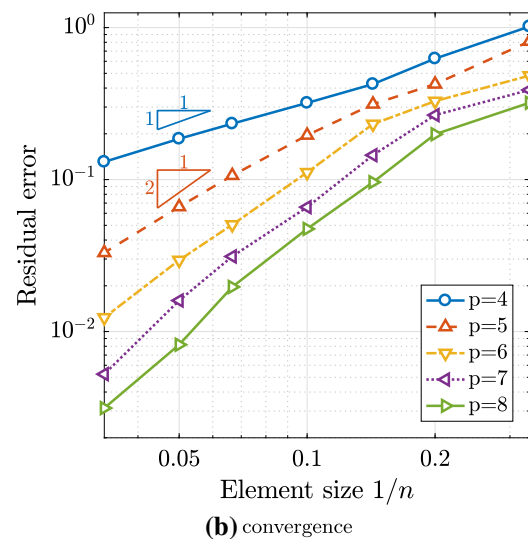
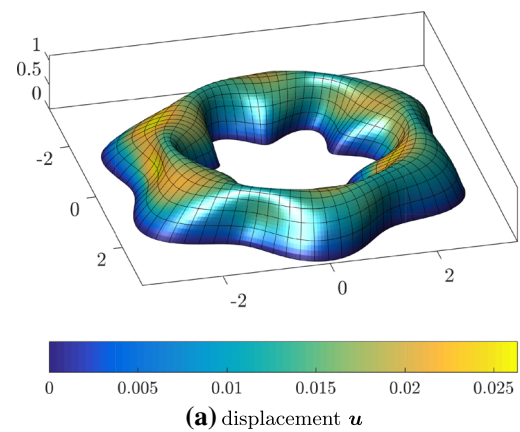


Fig. 14 Flower shaped shell: **a** displacement u of flower shaped shell (scaled by one order of magnitude), **b** residual error $\epsilon_{\text{rel, residual}}$

implies a parametrization and although the involved equations and the resulting implementations vary significantly, it is seen that the classical, parametrization-based and the proposed TDC-based formulation are equivalent. For a generic finite element framework enabling various implementations for PDEs on manifolds (in addition to only shells), the TDC-based approach is beneficial, because surface gradients of shape functions may be computed beforehand and are independent of the application.

The numerical results confirm higher-order convergence rates. As mentioned, based on the residual errors, a framework for the verification of complex test cases is presented. There is a large potential in the parametrization-free reformulation of shell models, because the obtained PDEs may be discretized with new finite element techniques such as Trace-FEM or CutFEM based on implicitly defined surfaces. In this case, neither the problem statement nor the discretization is based on a parametrization.


```

function [ElemMat, ElemRhs] = GetElementContributionShell(...
    ShapeFcts, ipReal, Parameters, NormalVectors)

% Compute element stiffness matrix.
%
% Input variables:
% ShapeFcts is a data structure storing shape functions and their surface
% derivatives as members. Each member is an (n x m)-matrix of n shape
% functions evaluated at m integration points. Existing members:
% -ShapeFcts.NodeNum: Number n of shape functions in this element/knot span.
% -ShapeFcts.f: shape functions (e.g., NURBS or classical FE functions).
% -ShapeFcts.fx, .fy, .fz: Surface gradient applied to each shape function.
% -ShapeFcts.fxx, .fyy, .fzz, .fxy, .fyz: *Covariant* 2nd derivatives
% of the shape functions.
% ipReal stores the integration points and weights in members:
% -ipReal.nQ: Number of integration points.
% -ipReal.xx, ipReal.yy, ipReal.zz: Coordinates of integration points.
% -ipReal.wv: Integration weights.
% Parameters stores material parameters.
% NormalVectors is a matrix with 3 columns storing the normal vector
% components at each integration point.
%
% Output variables:
% ElemMat: Element matrix.
% ElemRhs: Element right hand side.

nn = ShapeFcts.NodeNum;
ElemMat = zeros(3*nn, 3*nn); % Size of element matrix.
ElemRhs = zeros(3*nn, 1); % Size of element right hand side.

% Material parameters.
DD = Parameters.DD;
nu = Parameters.nu;
mu = Parameters.mu;
lambda = Parameters.lambda;
tt = Parameters.tt;

% Loading in x-, y-, z-direction evaluated at integration points.
[fx, fy, fz] = EvaluateLoad(ipReal.xx, ipReal.yy, ipReal.zz);

% Loop over integration points.
for i = 1 : ipReal.nQ

% Read out shape functions and derivatives at current integration point.
N = ShapeFcts.f(:, i);
Nx = ShapeFcts.fx(:, i); Ny = ShapeFcts.fy(:, i); Nz = ShapeFcts.fz(:, i);
Nxx = ShapeFcts.fxx(:, i); Nyy = ShapeFcts.fyy(:, i); Nzz = ShapeFcts.fzz(:, i);
Nxy = ShapeFcts.fxy(:, i); Nxz = ShapeFcts.fxz(:, i); Nyz = ShapeFcts.fyz(:, i);

% Define projectors, (3x3)-matrices:
Q = NormalVectors(i, :) * NormalVectors(i, :);
P = eye(3,3) - Q;

% Membrane stiffness, contribution at integration point.
Mat11 = (lambda+2*mu) * Nx*Nx' + mu * Ny*Ny' + mu * Nz*Nz';
Mat12 = lambda * Nx*Ny' + mu * Ny*Nx';
Mat13 = lambda * Nx*Nz' + mu * Nz*Nx';

Mat21 = lambda * Ny*Nx' + mu * Nx*Ny';
Mat22 = (lambda+2*mu) * Ny*Ny' + mu * Nx*Nx' + mu * Nz*Nz';
Mat23 = lambda * Ny*Nz' + mu * Nz*Ny';

Mat31 = lambda * Nz*Nx' + mu * Nx*Nz';
Mat32 = lambda * Nz*Ny' + mu * Ny*Nz';
Mat33 = (lambda+2*mu) * Nz*Nz' + mu * Nx*Nx' + mu * Ny*Ny';

KKmemb11 = P(1,1)*Mat11 + P(1,2)*Mat21 + P(1,3)*Mat31;
KKmemb12 = P(1,1)*Mat12 + P(1,2)*Mat22 + P(1,3)*Mat32;
KKmemb13 = P(1,1)*Mat13 + P(1,2)*Mat23 + P(1,3)*Mat33;

KKmemb21 = P(2,1)*Mat11 + P(2,2)*Mat21 + P(2,3)*Mat31;
KKmemb22 = P(2,1)*Mat12 + P(2,2)*Mat22 + P(2,3)*Mat32;
KKmemb23 = P(2,1)*Mat13 + P(2,2)*Mat23 + P(2,3)*Mat33;

KKmemb31 = P(3,1)*Mat11 + P(3,2)*Mat21 + P(3,3)*Mat31;
KKmemb32 = P(3,1)*Mat12 + P(3,2)*Mat22 + P(3,3)*Mat32;
KKmemb33 = P(3,1)*Mat13 + P(3,2)*Mat23 + P(3,3)*Mat33;

KKmemb = tt * [...
    KKmemb11 KKmemb12 KKmemb13;
    KKmemb21 KKmemb22 KKmemb23;
    KKmemb31 KKmemb32 KKmemb33];

% Bending stiffness, contribution at integration point.
% ...covariant Hesse contraction (with covariant derivatives)
K1 = Nxx*Nxx' + Nyy*Nyy' + Nzz*Nzz' + 2*(Nxy*Nxy' + Nxz*Nxz' + Nyz*Nyz');
% ...Bi-Laplace-Beltrami
K2 = (Nxx+Nyy+Nzz) * (Nxx+Nyy+Nzz)';
MatW = (1-nu) * K1 + nu * K2;

KKbend = DD * [...
    MatW*Q(1,1) MatW*Q(1,2) MatW*Q(1,3);
    MatW*Q(2,1) MatW*Q(2,2) MatW*Q(2,3);
    MatW*Q(3,1) MatW*Q(3,2) MatW*Q(3,3)];

% Right hand side, contribution at integration point.
Rhs1 = N * fx(i);
Rhs2 = N * fy(i);
Rhs3 = N * fz(i);
Rhs = [Rhs1; Rhs2; Rhs3];

% Add contribution at integration point to element matrix and rhs.
ElemMat = ElemMat + ipReal.wv(i) * (KKmemb + Kkbend);
ElemRhs = ElemRhs + ipReal.wv(i) * Rhs;

end

function [fx, fy, fz] = EvaluateLoad(xx, yy, zz)

nn = length(xx);
fx = zeros(nn, 1);
fy = zeros(nn, 1);
fz = zeros(nn, 1);

end

```

Code 1: Element contribution in terms of TDC

Acknowledgements Open access funding provided by Graz University of Technology.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

A Element stiffness matrix

In order to clarify the implementation, we give the Matlab®-code of the routine which evaluates the element contribution to the matrix and right hand side.

Using TDC, the input contains the shape function data and normal vectors evaluated at the integration points plus the material parameters, see Code 1. Note that the first- and

second-order surface derivatives are included in the shape function data, i.e., $\nabla_{\Gamma} N_i(x_j)$ and $\mathbf{H}^{\text{cov}}(N_i(x_j))$ where $i = 1, \dots, n$ refers to the n shape functions in the current element (knot span) and x_j with $j = 1, \dots, m$ to integration points. The computation of these quantities is part of the standard finite element technology provided by the implementation and is independent of the application to shells.

When curvilinear coordinates are used, the input contains shape function data, integration points, material parameters plus the coordinates of the control points of the corresponding element, see Code 2. Note that the included derivatives of the shape functions are the derivatives w.r.t. reference coordinates, i.e., $\nabla_{\mathbf{r}} N(\mathbf{r})$. As a first step, the covariant base vectors and the metric tensor in co- and contra-variant form need to be calculated at each integration point and in the next step, the element stiffness matrix and the element load vector are computed.

```

function [ElemMat, ElemRhs] = GetElementMatrixCurvilinear(ShapeFctsRef, ipRef, ...
Parameters, CoordRealElem)

% Compute element stiffness matrix based on curvilinear coordinates
% following Kiendl et al., Comp. Methods Appl. Mech. Engrg., 198, 39023914, 2009.
%
% Input variables:
% ShapeFctsRef is a data structure storing reference shape functions and
% derivatives as members. Each member is an (n x m)-matrix of n shape
% functions evaluated at m integration points. Existing members:
% -ShapeFctsRef.NodeNum: Number n of shape functions in this element/knot span.
% -ShapeFctsRef.f: shape functions (e.g., NURBS or classical FE functions).
% -ShapeFctsRef.fx, .fy: 1st derivatives of the shape functions.
% -ShapeFctsRef.fxx, .fxy, .fyy: 2nd derivatives of the shape functions.
% ipRef stores the integration points and weights in the reference element:
% -ipRef.nQ: Number of integration points.
% -ipRef.xx, ipRef.yy: Coordinates of integration points.
% -ipRef.w: Integration weights.
% Parameters stores material parameters.
% CoordRealElem stores the 3D coordinates of the real curved element.
% -CoordRealElem.xx, .yy, .zz: (x,y,z)-coordinates
%
% Output variables:
% ElemMat: Element matrix.
% ElemRhs: Element right hand side.

% Initialization.
nn = ShapeFctsRef.NodeNum;
ElemMat = zeros(3*nn, 3*nn); % Size of element matrix.
ElemRhs = zeros(3*nn, 1); % Size of element right hand side.

% Integration points in real curved shell element.
ipReal.nQ = ipRef.nQ;
ipReal.xx = ShapeFctsRef.f.' * CoordRealElem.xx;
ipReal.yy = ShapeFctsRef.f.' * CoordRealElem.yy;
ipReal.zz = ShapeFctsRef.f.' * CoordRealElem.zz;

% Loading in x-, y-, z-direction evaluated at integration points.
[fx, fy, fz] = EvaluateLoad(ipReal.xx, ipReal.yy, ipReal.zz);

% Material parameters.
EE = Parameters.EE;
nu = Parameters.nu;
tt = Parameters.tt;
Dmemb = EE*tt/(1-nu^2);
Dbend = EE*tt^3/12/(1-nu^2);

% Loop over integration points.
for i = 1 : ipRef.nQ

% Reference shape functions at current integration point.
N = ShapeFctsRef.f(:, i);
Nr = ShapeFctsRef.fx(:, i); Ns = ShapeFctsRef.fy(:, i);
Nrr = ShapeFctsRef.fxx(:, i); Nrs = ShapeFctsRef.fxy(:, i); Nss = ShapeFctsRef.fyy(:, i);

% Get covariant basis.
A1Cov = [Nr'*CoordRealElem.xx; Ns'*CoordRealElem.yy; Ns'*CoordRealElem.zz];
A2Cov = [Ns'*CoordRealElem.xx; Ns'*CoordRealElem.yy; Ns'*CoordRealElem.zz];

% Get partial derivatives of covariant basis vectors.
A1Cov_1 = [Nrr'*CoordRealElem.xx; Nrr'*CoordRealElem.yy; Nrr'*CoordRealElem.zz];
A1Cov_2 = [Nrs'*CoordRealElem.xx; Nrs'*CoordRealElem.yy; Nrs'*CoordRealElem.zz]; %=A2Cov_1
A2Cov_2 = [Nss'*CoordRealElem.xx; Nss'*CoordRealElem.yy; Nss'*CoordRealElem.zz];

% Normal vector.
A3tilde = cross(A1Cov, A2Cov);
A3bar = norm(A3tilde);
A3 = A3tilde / A3bar;

% Get covariant metric tensor.
A11Cov = dot(A1Cov, A1Cov); A12Cov = dot(A1Cov, A2Cov); A22Cov = dot(A2Cov, A2Cov);
AACov = [A11Cov A12Cov; A12Cov A22Cov];

% Get contravariant metric tensor.
AACont = inv(AACov);
A11Cont = AACont(1,1);
A12Cont = AACont(1,2);
A22Cont = AACont(2,2);

% Evaluate constitutive matrix.
c1111 = A11Cont^2;
c1122 = nu * A11Cont * A22Cont + (1-nu) * A12Cont^2;
c1112 = A11Cont * A12Cont;
c2222 = A22Cont^2;
c2212 = A12Cont * A22Cont;
c1212 = 1/2 * ((1+nu) * A12Cont^2 + (1-nu) * A11Cont * A22Cont);

H = [c1111 c1122 c1112;
c1122 c2222 c2212;
c1112 c2212 c1212];

Hmemb = Dmemb * H;
Hbend = Dbend * H;

% Compute membrane stiffness.
m11 = [Nr' * A1Cov(1), Nr' * A1Cov(2), Nr' * A1Cov(3)];
m22 = [Ns' * A2Cov(1), Ns' * A2Cov(2), Ns' * A2Cov(3)];
m12 = [...
0.5 * (Nr' * A2Cov(1) + Ns' * A1Cov(1)), ...
0.5 * (Nr' * A2Cov(2) + Ns' * A1Cov(2)), ...
0.5 * (Nr' * A2Cov(3) + Ns' * A1Cov(3))];
M = [m11; m22; 2*m12];
KKmemb = M' * Hmemb * M;

% Compute bending stiffness.
b11Q = Nrr' - 1/A3bar*(dot(cross(A2Cov, A3), A1Cov_1)*Nr' ...
+ dot(cross(A3, A1Cov), A1Cov_1)*Ns');
b22Q = Nss' - 1/A3bar*(dot(cross(A2Cov, A3), A2Cov_2)*Nr' ...
+ dot(cross(A3, A1Cov), A2Cov_2)*Ns');
b12Q = Nrs' - 1/A3bar*(dot(cross(A2Cov, A3), A1Cov_2)*Nr' ...
+ dot(cross(A3, A1Cov), A1Cov_2)*Ns');
b11 = [A3(1) * b11Q, A3(2) * b11Q, A3(3) * b11Q];
b22 = [A3(1) * b22Q, A3(2) * b22Q, A3(3) * b22Q];
b12 = [A3(1) * b12Q, A3(2) * b12Q, A3(3) * b12Q];
B = [b11; b22; 2*b12];
KKbend = B' * Hbend * B;

% Right hand side, contribution at integration point.
Rhs = [N*fx(i); N*fy(i); N*fz(i)];

% Add contribution at integration point to element matrix and rhs.
ElemMat = ElemMat + ipRef.w(i)*A3bar * (KKmemb + Kkbend);
ElemRhs = ElemRhs + ipRef.w(i)*A3bar * Rhs;

end

function [fx, fy, fz] = EvaluateLoad(xx, yy, zz)

nn = length(xx);
fx = zeros(nn, 1);
fy = zeros(nn, 1);
fz = zeros(nn, 1);

end

```

Code 2: Element contribution in terms of local coordinates

References

1. Başar Y, Krätzig WB (1985) *Mechanik der Flächentragwerke*. Vieweg+Teubner Verlag, Braunschweig
2. Belytschko T, Stolarski H, Liu WK, Carpenter N, Ong JSJ (1985) Stress projection for membrane and shear locking in shell finite elements. *Comput Methods Appl Mech Eng* 51:221–258
3. Benson DJ, Bazilevs Y, Hsu MC, Hughes TJR (2010) Isogeometric shell analysis: the Reissner–Mindlin shell. *Comput Methods Appl Mech Eng* 199:276–289
4. Bischoff M, Bletzinger KU, Wall WA, Ramm E (2004) Models and finite elements for thin-walled structures, chapter 3. In: *Encyclopedia of computational mechanics*. Wiley, Chichester
5. Blaauwendraad J, Hoefakker JH (2014) *Structural shell analysis. Solid mechanics and its applications*, vol 200. Springer, Berlin
6. Burman E, Claus S, Hansbo P, Larson MG, Massing A (2015) CutFEM: discretizing geometry and partial differential equations. *Int J Numer Methods Eng* 104:472–501
7. Burman E, Elfverson D, Hansbo P, Larson MG, Larsson K (2018) Shape optimization using the cut finite element method. *Comput Methods Appl Mech Eng* 328:242–261
8. Cenanovic M, Hansbo P, Larson MG (2016) Cut finite element modeling of linear membranes. *Comput Methods Appl Mech Eng* 310:98–111
9. Cirak F, Ortiz M, Schröder P (2000) Subdivision surfaces: a new paradigm for thin-shell finite-element analysis. *Int J Numer Methods Eng* 47(12):2039–2072
10. Cottrell JA, Hughes TJR, Bazilevs Y (2009) *Isogeometric analysis: toward integration of CAD and FEA*. Wiley, Chichester
11. Delfour MC, Zolésio JP (1994) Shape analysis via oriented distance functions. *J Funct Anal* 123:129–201
12. Delfour MC, Zolésio JP (1995) A boundary differential equation for thin shells. *J Differ Equ* 119:426–449
13. Delfour MC, Zolésio JP (1996) Tangential differential equations for dynamical thin shallow shells. *J Differ Equ* 128:125–167
14. Delfour MC, Zolésio JP (1997) Differential equations for linear shells comparison between intrinsic and classical. In: *Advances in mathematical sciences: CRM's 25 years (Montreal, PQ, 1994)*, Vol. 11 of CRM proceedings and lecture notes, Providence, Rhode Island
15. Delfour MC, Zolésio JP (2011) *Shapes and geometries: metrics, analysis, differential calculus, and optimization*. SIAM, Philadelphia

16. Demlow A (2009) Higher-order finite element methods and pointwise error estimates for elliptic problems on surfaces. *SIAM J Numer Anal* 47:805–827
17. Dziuk G (1988) Finite elements for the beltrami operator on arbitrary surfaces: chapter 6. Springer, Berlin, pp 142–155
18. Dziuk G, Elliott CM (2013) Finite element methods for surface PDEs. *Acta Numer* 22:289–396
19. Elfverson D, Larson MG, Larsson K (2018) A new least squares stabilized Nitsche method for cut isogeometric analysis. ArXiv e-prints [ArXiv:1804.05654](https://arxiv.org/abs/1804.05654)
20. Fries TP (2018) Higher-order surface FEM for incompressible Navier-Stokes flows on manifolds. *Int J Numer Methods Fluids* 88:55–78. <https://doi.org/10.1002/fld.4510>
21. Fries TP, Omerović S, Schöllhammer D, Steidl J (2017) Higher-order meshing of implicit geometries—part I: integration and interpolation in cut elements. *Comput Methods Appl Mech Eng* 313:759–784
22. Fries TP, Schöllhammer D (2017) Higher-order meshing of implicit geometries—part II: approximations on manifolds. *Comput Methods Appl Mech Eng* 326:270–297
23. Grande J, Reusken A (2016) A higher order finite element method for partial differential equations on surfaces. *SIAM* 54:388–414
24. Guo Y, Ruess M, Schillinger D (2017) A parameter-free variational coupling approach for trimmed isogeometric thin shells. *Comput Mech* 59:693–715
25. Gurtin ME, Murdoch IA (1975) A continuum theory of elastic material surfaces. *Arch Ration Mech Anal* 57:291–323
26. Hansbo P, Larson MG (2014) Finite element modeling of a linear membrane shell problem using tangential differential calculus. *Comput Methods Appl Mech Eng* 270:1–14
27. Hansbo P, Larson MG (2017) Continuous/discontinuous finite element modelling of Kirchhoff plate structures in \mathbb{R}^3 using tangential differential calculus. *Comput Mech* 60:693–702
28. Hansbo P, Larson MG, Larsson F (2015) Tangential differential calculus and the finite element modeling of a large deformation elastic membrane problem. *Comput Mech* 56:87–95
29. Hansbo P, Larson MG, Larsson K (2014) Variational formulation of curved beams in global coordinates. *Comput Mech* 53:611–623
30. Hughes TJR, Cottrell JA, Bazilevs Y (2005) Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Comput Methods Appl Mech Eng* 194:4135–4195
31. Jankuhn T, Olshanskii MA, Reusken A (2017) Incompressible fluid problems on embedded surfaces modeling and variational formulations. ArXiv e-prints [arXiv:1702.02989](https://arxiv.org/abs/1702.02989)
32. Kiendl J, Bletzinger K-U, Linhard J, Wüchner R (2009) Isogeometric shell analysis with Kirchhoff–Love elements. *Comput Methods Appl Mech Eng* 198:3902–3914
33. Lebedzik C (2007) Exact boundary controllability of a shallow intrinsic shell model. *J Math Anal Appl* 335:584–614
34. Nguyen VP, Anitescu C, Bordas SPA, Rabczuk T (2015) Isogeometric analysis: an overview and computer implementation aspects. *Math Comput Simul* 117:89–116
35. Nguyen-Thanh N, Valizadeh N, Nguyen MN, Nguyen-Xuan H, Zhuang X, Areias P, Zi G, Bazilevs Y, Lorenzis L, De Rabczuk T (2015) An extended isogeometric thin shell analysis based on Kirchhoff–Love theory. *Comput Methods Appl Mech Eng* 284:265–291
36. Nguyen-Thanh N, Zhou K, Zhuang X, Areias P, Nguyen-Xuan H, Bazilevs Y, Rabczuk T (2017) Isogeometric analysis of large-deformation thin shells using RHT-splines for multiple-patch coupling. *Comput Methods Appl Mech Eng* 316:1157–1178
37. Olshanskii MA, Reusken A (2017) Trace finite element methods for PDEs on surfaces. In: *Lecture notes in computational science and engineering*, vol 121, pp 211–258
38. Olshanskii MA, Xu X (2017) A trace finite element method for PDEs on evolving surfaces. *SIAM* 39:A1301–A1319
39. Osher S, Fedkiw RP (2003) *Level set methods and dynamic implicit surfaces*. Springer, Berlin
40. Piegl L, Tiller W (1997) *The NURBS book* (monographs in visual communication), 2nd edn. Springer, Berlin
41. Radwańska M, Stankiewicz A, Wosatko A, Pamin J (2017) *Plate and shell structures*. Wiley, Chichester
42. Reusken A (2014) Analysis of trace finite element methods for surface partial differential equations. *IMA J Numer Anal* 35:1568–1590
43. Sethian JA (1999) *Level set methods and fast marching methods*, 2nd edn. Cambridge University Press, Cambridge
44. Simo JC, Fox DD (1989) On a stress resultant geometrically exact shell model—part I: formulation and optimal parametrization. *Comput Methods Appl Mech Eng* 72:267–304
45. Simo JC, Fox DD, Rifai MS (1989) On a stress resultant geometrically exact shell model—part II: the linear theory; computational aspects. *Comput Methods Appl Mech Eng* 73:53–92
46. Timoshenko S, Woinowsky-Krieger S (1959) *Theory of plates and shells*, 2nd edn. McGraw-Hill Book Company Inc, New York
47. van Opstal TM, van Brummelen EH, van Zwieten GJ (2015) A finite-element/boundary-element method for three-dimensional, large-displacement fluid structure-interaction. *Comput Methods Appl Mech Eng* 284:637–663
48. Walker SW (2015) *The shapes of things: a practical guide to differential geometry and the shape derivative*. Advances in design and control. SIAM, Philadelphia
49. Wempner G, Talaslidis D (2002) *Mechanics of solids and shells: theories and approximations*. CRC Press, Boca Raton
50. Yao PF (2009) On shallow shell equations. *Discret Contin Dyn Syst Ser S* 2:697–722
51. Zienkiewicz O, Taylor R, Zhu JZ (2013) *The finite element method: its basis and fundamentals*, 7th edn. Elsevier, Oxford

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.