Isogeometric collocation for the Reissner-Mindlin shell problem

Josef Kiendl[a,*], Enzo Marino[b], Laura De Lorenzis[c]

[a]*Department of Marine Technology – Norwegian University of Science and Technology, NO-7491 Trondheim, Norway.*
[b]*Department of Civil and Environmental Engineering – University of Florence, Via di S. Marta 3, 50139 Firenze, Italy.*
[c]*Institute of Applied Mechanics – TU Braunschweig, Bienroder Weg 87, 38106 Braunschweig, Germany.*

Abstract

We present an isogeometric collocation formulation for the Reissner-Mindlin shell problem. After recalling the necessary basics on differential geometry and the shell governing equations, we show that the standard approach of expressing the equilibrium equations in terms of the primal variables is not a suitable way for shells due to the complexity of the underlying equations. We then propose an alternative approach, based on a stepwise formulation, and show its numerical implementation within an isogeometric collocation framework. The formulation is tested successfully on a set of benchmark examples, which comprise important aspects like locking and boundary layers. These test show that locking effects can be conveniently avoided by using high polynomial degrees. An accompanying study on the computational time also confirms that high polynomial degrees are preferable in terms of computational efficiency.

Keywords: Isogeometric, Collocation, Reissner-Mindlin, Shells, NURBS

1. Introduction

The motivation of Isogeometric Analysis (IGA) is to bridge the gap between Computer Aided Design (CAD) and Finite Element Analysis (FEA) by adopting Non-Uniform Rational B-Splines (NURBS), commonly used for geometry representation in CAD, as basis functions for analysis [1]. Moreover, IGA has gained enormous popularity as a numerical analysis method since it exhibits increased accuracy and robustness properties on a per-

*Corresponding author
Email address: josef.kiendl@ntnu.no (Josef Kiendl)

degree-of-freedom basis compared to standard FEA, which is attributed to the higher order and continuity properties inherent in the basis functions [2–5]. A field where IGA had an especially high impact is shell analysis. The smoothness of the basis functions allows for efficient implementations of rotation-free Kirchhoff-Love shell models [6–14] as well as novel approaches like hierarchic shells [15] and rotation-free shear deformable shells [16]. But also for Reissner-Mindlin shells [17–20] and solid-shells [21–25], the continuity properties of the basis functions and the exact geometry description turn out to be very advantageous in comparison to standard FEA. An interesting feature of IGA in the context of shell analysis is also that locking problems can easily be avoided by simply raising the polynomial degree. However, this comes at the price of increased computational cost during assembly, since the typically used quadrature rules, which are optimal for C^0-continuous elements, are suboptimal for elements with smooth basis functions. Different improved quadrature rules have been proposed [26–29], but the development of general and efficient integration rules is still an open problem in IGA.

The lack of efficient integration rules in IGA has recently led to the development of isogeometric collocation (IGA-C) methods [30], where the high continuity of the basis functions is exploited to solve the governing partial differential equations in strong form. In such an approach, no integrals have to be evaluated and only one evaluation point per degree of freedom is needed (which approximately means one point per element), independently of the polynomial degree. This results in a drastically reduced computational cost compared to Galerkin-based IGA, especially for high polynomial degrees. A comprehensive study of the computational costs comparing standard FEA, IGA, and IGA-C can be found in [31], consistency and convergence properties of IGA-C are discussed in [32]. Despite being very new, isogeometric collocation has already been applied successfully to various problems including elastostatics and explicit dynamics [33], structural mechanics of beams [34, 35], spatial rods [36–39], and plates [40, 41], large deformation elasticity [42], contact [42, 43], phase-field modeling [44], and fracture [45]. Clearly, the advantages of IGA-C take the most effect for problems where the total computational cost is governed by the formation of the system matrices, as in the case of explicit dynamics. In fact, an important target of IGA-C is the application to explicit structural dynamics, like crash worthiness simulations. An important

step towards this goal is the development of appropriate structural formulations, in particular for shells. This paper presents the first approach to derive isogeometric collocation formulations for shell analysis. In particular, we present a displacement-based formulation for the Reissner-Mindlin shell problem. We show that the standard approach in IGA-C, where the equilibrium equations are directly written in terms of the primal variables and then collocated, is not a suitable option for shells due to the complexity of the underlying equations. Instead, we propose a stepwise formulation, which drastically decreases the complexity of the equations to be implemented as well as the computational effort. We present the details of the formulation and its implementation, and demonstrate its performance on a series of numerical benchmark examples.

The paper is outlined as follows. In Section 1, we present the basics of differential geometry of surfaces, which are necessary to formulate the shell problem. Section 2 presents the governing equations of the Reissner-Mindlin shell. In Section 3, we present a stepwise formulation of the problem and its implementation in an IGA-C approach. In Section 4, we test the formulation on a set of benchmark examples, and in Section 5, we draw conclusions.

2. Differential geometry of surfaces

In this section, we briefly review the basics of differential geometry, restricting ourselves to what is needed in the following. We use index notation with Greek indices taking on values $\{1, 2\}$ and Latin indices taking on values $\{1, 2, 3\}$, and summation over repeated indices is assumed. Subscript indices indicate covariant quantities, while superscript indices refer to contravariant quantities.

A shell is represented by its midsurface and a thickness h, which we assume constant over the whole shell. A point on the shell midsurface is indicated by $\boldsymbol{r} = \boldsymbol{r}(\theta^1, \theta^2)$ with (θ^1, θ^2) as the natural curvilinear surface coordinates of the midsurface. Furthermore, we have the thickness coordinate (θ^3) in the direction orthogonal to the midsurface. Partial derivatives with respect to these natural coordinates are indicated by comma $(\cdot)_{,i} = \partial(\cdot)/\partial\theta^i$. At each point of the midsurface, a covariant basis is formed by the tangent vectors

$$\boldsymbol{a}_\alpha = \boldsymbol{r}_{,\alpha}, \tag{1}$$

and the unit normal vector

$$\boldsymbol{a}_3 = \frac{\boldsymbol{a}_1 \times \boldsymbol{a}_2}{|\boldsymbol{a}_1 \times \boldsymbol{a}_2|}. \tag{2}$$

Contravariant base vectors are defined by

$$\boldsymbol{a}^i \cdot \boldsymbol{a}_j = \delta^i_j, \tag{3}$$

where δ^i_j is the Kronecker delta. Covariant metric coefficients are obtained by the first fundamental form of surfaces

$$a_{\alpha\beta} = \boldsymbol{a}_\alpha \cdot \boldsymbol{a}_\beta, \tag{4}$$

and contravariant metric coefficients are obtained analogously by

$$a^{\alpha\beta} = \boldsymbol{a}^\alpha \cdot \boldsymbol{a}^\beta, \tag{5}$$

with

$$a_{\alpha\lambda} a^{\lambda\beta} = \delta^\beta_\alpha. \tag{6}$$

Co- and contravariant metric coefficients can be conveniently used to switch between co- and contravariant base vectors

$$\boldsymbol{a}^\alpha = a^{\alpha\beta} \boldsymbol{a}_\beta, \tag{7}$$

$$\boldsymbol{a}_\alpha = a_{\alpha\beta} \boldsymbol{a}^\beta. \tag{8}$$

Furthermore, we note that $\boldsymbol{a}^3 = \boldsymbol{a}_3$. The second fundamental form of surfaces provides the covariant curvature coefficients

$$b_{\alpha\beta} = \boldsymbol{a}_{\alpha,\beta} \cdot \boldsymbol{a}_3. \tag{9}$$

Mixed and contravariant curvature coefficients, which will be also needed, can be obtained via the index raising property of the contravariant metric coefficients

$$b^\alpha_\beta = a^{\alpha\lambda} b_{\lambda\beta}, \tag{10}$$

$$b^{\alpha\beta} = a^{\alpha\lambda} a^{\beta\mu} b_{\lambda\mu}. \tag{11}$$

It should be noted that $b_{\alpha\beta}$ do not provide an objective measure for the curvature but depend on the parametrization. The physical curvatures in directions of $\boldsymbol{a}_1, \boldsymbol{a}_2$ are given by

$$\frac{1}{R_1} = \frac{b_{11}}{a_{11}}, \qquad \frac{1}{R_2} = \frac{b_{22}}{a_{22}}, \tag{12}$$

with R_1, R_2 being the curvature radii. At each point, there exist two directions for which the curvatures have extreme values $1/R_{max}, 1/R_{min}$. The Gaussian curvature is defined as

$$K = \frac{1}{R_{min}} \frac{1}{R_{max}} = \frac{|b_{\alpha\beta}|}{|a_{\alpha\beta}|}, \tag{13}$$

with $|(\cdot)|$ indicating the determinant. The Gaussian curvature is a surface invariant and can be used to classify surfaces pointwise into the categories *elliptic* ($K > 0$), *parabolic* ($K = 0$), and *hyperbolic* ($K < 0$). For many important shell geometries, this condition ($K > 0, K = 0, K < 0$) is constant for the entire surface, and the categories *elliptic*, *parabolic*, and *hyperbolic* are then also used as global attributes. This classification plays an important role in shell analysis since the structural behavior is very different for the different categories, especially if the shell thickness is small [46, 47].

For computing covariant derivatives, we need to introduce the Christoffel symbols, which are defined as

$$\Gamma^{\lambda}_{\alpha\beta} = \boldsymbol{a}_{\alpha,\beta} \cdot \boldsymbol{a}^{\lambda}. \tag{14}$$

Covariant derivatives are indicated by $(\cdot)|_{\alpha}$ and are defined for vectors (i.e., first order tensors) by

$$A_{\alpha}|_{\beta} = A_{\alpha,\beta} - A_{\lambda}\Gamma^{\lambda}_{\alpha\beta}, \tag{15}$$

$$A^{\alpha}|_{\beta} = A^{\alpha}{}_{,\beta} + A^{\lambda}\Gamma^{\alpha}_{\lambda\beta}, \tag{16}$$

for second order tensors by

$$A_{\alpha\beta}|_{\gamma} = A_{\alpha\beta,\gamma} - A_{\lambda\beta}\Gamma^{\lambda}_{\alpha\gamma} - A_{\alpha\lambda}\Gamma^{\lambda}_{\beta\gamma}, \tag{17}$$

$$A^{\alpha}{}_{\beta}|_{\gamma} = A^{\alpha}{}_{\beta,\gamma} + A^{\lambda}{}_{\beta}\Gamma^{\alpha}_{\gamma\lambda} - A^{\alpha}{}_{\lambda}\Gamma^{\lambda}_{\beta\gamma}, \tag{18}$$

$$A^{\alpha\beta}|_{\gamma} = A^{\alpha\beta}{}_{,\gamma} + A^{\lambda\beta}\Gamma^{\alpha}_{\gamma\lambda} + A^{\alpha\lambda}\Gamma^{\beta}_{\gamma\lambda}, \tag{19}$$

while for scalars they are identical to the parametric derivative

$$A|_\alpha = A_{,\alpha}\,. \tag{20}$$

Using the fact that the covariant derivatives of metric coefficients vanish, $a_{\alpha\beta}|_\gamma = a^{\alpha\beta}|_\gamma = 0$, we can compute the covariant derivatives of contravariant or mixed components alternatively as

$$A^\alpha|_\gamma = a^{\alpha\lambda}A_\lambda|_\gamma, \tag{21}$$

$$A^\alpha{}_\beta|_\gamma = a^{\alpha\lambda}A_{\lambda\beta}|_\gamma, \tag{22}$$

$$A^{\alpha\beta}|_\gamma = a^{\alpha\lambda}a^{\beta\mu}A_{\lambda\mu}|_\gamma, \tag{23}$$

which can be very useful if the covariant derivatives of the covariant components $(A_\alpha|_\beta, A_{\alpha\beta}|_\gamma)$ have already been computed. The second covariant derivative of a vector can be computed with the rule for the first covariant derivative of a tensor (17)

$$\begin{aligned} A_\alpha|_{\beta\gamma} &= (A_\alpha|_\beta)|_\gamma \\ &= (A_\alpha|_\beta)_{,\gamma} - A_\lambda|_\beta\Gamma^\lambda_{\alpha\gamma} - A_\alpha|_\lambda\Gamma^\lambda_{\beta\gamma} \\ &= A_{\alpha,\beta\gamma} - A_{\lambda,\gamma}\Gamma^\lambda_{\alpha\beta} - A_\lambda\Gamma^\lambda_{\alpha\beta,\gamma} - A_\lambda|_\beta\Gamma^\lambda_{\alpha\gamma} - A_\alpha|_\lambda\Gamma^\lambda_{\beta\gamma}, \end{aligned} \tag{24}$$

while the second covariant derivative of a scalar can be computed like the first covariant derivative of a vector (15)

$$A|_{\alpha\beta} = (A|_\alpha)|_\beta = (A_{,\alpha})|_\beta = A_{,\alpha\beta} - A_{,\lambda}\Gamma^\lambda_{\alpha\beta}. \tag{25}$$

The parametric derivative of the Christoffel symbol $\Gamma^\lambda_{\alpha\beta,\gamma}$, which appears in (24), can be developed from (14) as

$$\Gamma^\lambda_{\alpha\beta,\gamma} = \boldsymbol{a}_{\alpha,\beta\gamma}\cdot\boldsymbol{a}^\lambda + \boldsymbol{a}_{\alpha,\beta}\cdot\boldsymbol{a}^\lambda_{,\gamma}, \tag{26}$$

with

$$\boldsymbol{a}^\alpha_{,\beta} = -\Gamma^\alpha_{\lambda\beta}\boldsymbol{a}^\lambda + b^\alpha_\beta\boldsymbol{a}_3. \tag{27}$$

Furthermore, we will need covariant derivatives of the curvature coefficients $b_{\alpha\beta}|_\gamma$ and $b^\alpha_\beta|_\gamma$, which can be obtained by applying (17) and (22)

$$b_{\alpha\beta}|_\gamma = b_{\alpha\beta,\gamma} - b_{\lambda\beta}\Gamma^\lambda_{\alpha\gamma} - b_{\alpha\lambda}\Gamma^\lambda_{\beta\gamma} \tag{28}$$

with

$$b_{\alpha\beta,\gamma} = \boldsymbol{a}_{\alpha,\beta\gamma} \cdot \boldsymbol{a}_3 + \boldsymbol{a}_{\alpha,\beta} \cdot \boldsymbol{a}_{3,\gamma}, \tag{29}$$

$$\boldsymbol{a}_{3,\alpha} = -b_\alpha^\lambda \boldsymbol{a}_\lambda, \tag{30}$$

and, finally,

$$b_\beta^\alpha|_\gamma = a^{\alpha\lambda} b_{\lambda\beta}|_\gamma. \tag{31}$$

3. Shell governing equations

For describing the shell equations, we closely follow the work of Başar and Krätzig [48, 49] and adopt the notation therein. The deformation of a shell is described by a displacement vector $\boldsymbol{v}$ of the midsurface and a difference vector $\boldsymbol{w} = \bar{\boldsymbol{a}}_3 - \boldsymbol{a}_3$, where $\bar{\boldsymbol{a}}_3$ refers to the deformed configuration. The displacement vector is represented by two in-plane components and an out-of-plane component, $\boldsymbol{v} = v_\alpha \, \boldsymbol{a}^\alpha + v_3 \, \boldsymbol{a}^3$, while the difference vector is assumed to be tangential ($\boldsymbol{w} \cdot \boldsymbol{a}_3 = 0$) and, thus, is represented by in-plane components only, $\boldsymbol{w} = w_\alpha \, \boldsymbol{a}^\alpha$.

The strain state of the shell is described by the membrane or stretching strain tensor $\alpha_{\alpha\beta}$, the bending strain tensor $\beta_{\alpha\beta}$, and the shear strain vector γ_α, which are defined as

$$\alpha_{\alpha\beta} = \frac{1}{2} \left(v_\alpha|_\beta + v_\beta|_\alpha - 2 b_{\alpha\beta} v_3 \right), \tag{32}$$

$$\beta_{\alpha\beta} = \frac{1}{2} \left(w_\alpha|_\beta + w_\beta|_\alpha - b_\alpha^\lambda v_\lambda|_\beta - b_\beta^\lambda v_\lambda|_\alpha + 2 b_\alpha^\lambda b_{\lambda\beta} v_3 \right), \tag{33}$$

$$\gamma_\alpha = w_\alpha + v_{3,\alpha} + b_\alpha^\lambda v_\lambda. \tag{34}$$

Stresses are represented by the membrane force tensor $n^{\alpha\beta}$, the moment tensor $m^{\alpha\beta}$ and the shear force vector q^α. As described in [48, 49], $n^{\alpha\beta}$ is not symmetric in general, and the symmetric pseudo force tensor $\tilde{n}^{\alpha\beta}$ is introduced

$$\tilde{n}^{\alpha\beta} = n^{\alpha\beta} + b_\lambda^\beta m^{\alpha\lambda}. \tag{35}$$

Assuming linear elastic material, the stress tensors are obtained from the strain tensors as

$$\tilde{n}^{\alpha\beta} = h \, H^{\alpha\beta\lambda\mu} \, \alpha_{\lambda\mu}, \tag{36}$$

$$m^{\alpha\beta} = \frac{h^3}{12} \, H^{\alpha\beta\lambda\mu} \, \beta_{\lambda\mu}, \tag{37}$$

$$q^\alpha = G h \, a^{\alpha\lambda} \gamma_\lambda, \tag{38}$$

where h is the shell thickness, $G = \dfrac{E}{2(1+\nu)}$ is the shear modulus, E, ν are the Young's modulus and Poisson's ratio, respectively, and $H^{\alpha\beta\lambda\mu}$ is the elastic material tensor defined as

$$H^{\alpha\beta\lambda\mu} = \frac{E}{2(1+\nu)}\left(a^{\alpha\lambda}a^{\beta\mu} + a^{\alpha\mu}a^{\beta\lambda} + \frac{2\nu}{1-\nu}a^{\alpha\beta}a^{\lambda\mu}\right). \tag{39}$$

Considering a shell subjected to distributed loads p^α, p^3 and distributed moments c^α, the equilibrium equations read as

$$\tilde{n}^{\alpha\lambda}|_\lambda - b^\alpha_\mu|_\lambda m^{\lambda\mu} - b^\alpha_\mu m^{\lambda\mu}|_\lambda - b^\alpha_\lambda q^\lambda + p^\alpha = 0, \tag{40}$$

$$b_{\lambda\mu}\tilde{n}^{\lambda\mu} - b^\mu_\rho b_{\lambda\mu} m^{\lambda\rho} + q^\lambda|_\lambda + p^3 = 0, \tag{41}$$

$$m^{\alpha\lambda}|_\lambda - q^\alpha + c^\alpha = 0\,. \tag{42}$$

Eq. (40) represents two equations (corresponding to the free index α) for the equilibrium of in-plane forces, (41) represents the equilibrium of transversal forces, while (42) are the two equations (free index α) of rotational equilibrium. So, we have five equilibrium equations for the five unknowns v_1, v_2, v_3, w_1, w_2.

At the boundary, we consider a triad of unit vectors $(\boldsymbol{u}, \boldsymbol{t}, \boldsymbol{a}_3)$, where $\boldsymbol{u}$ is the outward normal vector and $\boldsymbol{t}$ is the tangent vector. The boundary equilibrium equations can then be written as

$$n^{\alpha\beta}u_\alpha u_\beta = (\tilde{n}^{\alpha\beta} - b^\beta_\lambda m^{\alpha\lambda})u_\alpha u_\beta = \bar{n}_u \tag{43}$$

$$n^{\alpha\beta}u_\alpha t_\beta = (\tilde{n}^{\alpha\beta} - b^\beta_\lambda m^{\alpha\lambda})u_\alpha t_\beta = \bar{n}_t \tag{44}$$

$$q^\alpha u_\alpha = \bar{n}_3 \tag{45}$$

$$m^{\alpha\beta}u_\alpha u_\beta = \bar{m}_u \tag{46}$$

$$m^{\alpha\beta}u_\alpha t_\beta = \bar{m}_t \tag{47}$$

with $\bar{n}_u, \bar{n}_t, \bar{n}_3, \bar{m}_u, \bar{m}_t$ as the prescribed boundary values. We note that the boundary equilibrium needs to be satisfied for the real forces $n^{\alpha\beta}$, which, however, can be expressed in terms of the pseudo stresses $\tilde{n}^{\alpha\beta}$ using (35).

4. Displacement-based formulation and isogeometric collocation method

The general approach for a displacement-based collocation method is to express the equilibrium equations in terms of the primal variables, which can be obtained by substituting the kinematic and constitutive equations into the equilibrium equations. However, such an approach is not practical for shells due to the complexity of the underlying equations. For some specific classes of shells, as for example cylindrical shells, the governing equations can be simplified and closed form equations can be found in the literature, but to the best of our knowledge, such closed forms are not available for the general shell problem. In an attempt to derive these equations one finds that the expanded terms become extremely long, complex, and nearly impossible to track manually. We have also used the computer algebra system Mathematica [50] with a specific tensor operation package [51] for deriving these expanded equations symbolically. The resulting equations span several pages and are not suited for being reported here. In Appendix A, we report exemplarily a part of these equations which makes clear that this approach is not a suitable basis for implementation. Besides the difficulty to implement such a formulation, it is computationally very inefficient.

Therefore, we propose an alternative approach, where the kinematic, constitutive, and equilibrium equations are evaluated in sequential order, computing intermediate variables which are then used for the subsequent equations. We refer to this approach as *stepwise formulation* and use it as the basis for the numerical method. This means that after discretizing the primal variables, we compute discretized versions of the intermediate variables and use them to collocate the equilibrium equations, as will be shown in detail in Section 4.2.

4.1. Stepwise formulation of the problem

The stepwise formulation of the problem is basically given by the kinematic, constitutive, and equilibrium equations. However, in order to have the equations in a form which can be implemented, we need to explicitly write all the terms stemming from covariant derivatives. We will not explicitly write all the index summation terms, since these summations can be easily performed in the computer program. In particular, we need to compute first and second covariant derivatives of the primal variables, and first covariant derivatives of the

strain and stress variables.

We start with the first and second derivatives for the displacement variables according to Eqs. (15), (20), (24), (25)

$$v_\alpha|_\beta = v_{\alpha,\beta} - v_\lambda \Gamma^\lambda_{\alpha\beta}, \tag{48}$$

$$v_3|_\beta = v_{3,\beta}, \tag{49}$$

$$v_\alpha|_{\beta\gamma} = v_{\alpha,\beta\gamma} - v_{\lambda,\gamma}\Gamma^\lambda_{\alpha\beta} - v_\lambda \Gamma^\lambda_{\alpha\beta,\gamma} - v_\lambda|_\beta \Gamma^\lambda_{\alpha\gamma} - v_\alpha|_\lambda \Gamma^\lambda_{\beta\gamma}, \tag{50}$$

$$v_3|_{\alpha\beta} = v_{3,\alpha\beta} - v_{3,\lambda}\Gamma^\lambda_{\alpha\beta}. \tag{51}$$

Obviously, the formulas for $v_\alpha|_\beta, v_\alpha|_{\beta\gamma}$ hold equally for $w_\alpha|_\beta, w_\alpha|_{\beta\gamma}$ and are therefore not repeated here. It should also be noted that for v_3 the derivation rules for scalars hold.

With (48)-(51) at hand, we can derive the covariant derivatives of the strain variables (32)-(34)

$$\alpha_{\alpha\beta}|_\gamma = \frac{1}{2}\left(v_\alpha|_{\beta\gamma} + v_\beta|_{\alpha\gamma} - 2b_{\alpha\beta}|_\gamma v_3 - 2b_{\alpha\beta} v_3|_\gamma\right), \tag{52}$$

$$\begin{aligned}\beta_{\alpha\beta}|_\gamma =& \frac{1}{2}\left(w_\alpha|_{\beta\gamma} + w_\beta|_{\alpha\gamma} - b^\lambda_\alpha|_\gamma v_\lambda|_\beta - b^\lambda_\alpha v_\lambda|_{\beta\gamma} - b^\lambda_\beta|_\gamma v_\lambda|_\alpha - b^\lambda_\beta v_\lambda|_{\alpha\gamma}\right.\\ &\left. + 2b^\lambda_\alpha|_\gamma b_{\lambda\beta} v_3 + 2b^\lambda_\alpha b_{\lambda\beta}|_\gamma v_3 + 2b^\lambda_\alpha b_{\lambda\beta} v_3|_\gamma\right),\end{aligned} \tag{53}$$

$$\gamma_\alpha|_\beta = w_\alpha|_\beta + v_3|_{\alpha\beta} + b^\lambda_\alpha|_\beta v_\lambda + b^\lambda_\alpha v_\lambda|_\beta. \tag{54}$$

Note that in (52)-(54), covariant derivatives of the curvature coefficients appear, $b_{\alpha\beta}|_\gamma, b^\alpha_\beta|_\gamma$, which are computed according to (28)-(31). Using the fact that covariant derivatives of the material tensor vanish, $H^{\alpha\beta\gamma\delta}|_\epsilon = 0$, the covariant derivatives of the stress resultants are simply obtained from (36)-(38) as

$$\tilde{n}^{\alpha\beta}|_\gamma = h\, H^{\alpha\beta\lambda\mu}\alpha_{\lambda\mu}|_\gamma, \tag{55}$$

$$m^{\alpha\beta}|_\gamma = \frac{h^3}{12}\, H^{\alpha\beta\lambda\mu}\beta_{\lambda\mu}|_\gamma, \tag{56}$$

$$q^\alpha|_\beta = Gha^{\alpha\lambda}\gamma_\lambda|_\beta. \tag{57}$$

With (55)-(57) and the formulas for the curvatures and their covariant derivatives, we have all terms we need in order to collocate the equilibrium equations (40)-(42). In the same manner, we can compute all terms in order to collocate the boundary equilibrium equations (43)-(47).

4.2. Isogeometric discretization and collocation

In this section we present the isogeometric collocation method following the stepwise formulation presented above, providing also some implementation details. The basics of B-splines, NURBS, and their use in isogeometric methods have been presented in many papers, so we do not repeat them here but refer to [52–54] for an introduction to B-splines and NURBS, and to [1, 55] for the basics of isogeometric analysis. In the following, we briefly review the concept of isogeometric collocation, for more details and a general introduction and overview to the topic, we refer to [30, 56]. The general idea of isogeometric collocation is to discretize the geometry and the solution variables by NURBS (or other isogeometric basis functions) and to collocate the discretized strong form equations at a set of suitable points, such that a square system of equations is obtained. Several sets of collocation points have been proposed so far, as for example the Greville abscissae [30], the Demko abscissae [30], Cauchy-Galerkin points [57] and their estimates, i.e., superconvergent points [57–59], and the search for optimal collocation points is still an active field of research [59]. In this work, we adopt the Greville abscissae as collocation points, which up to date is the standard approach.

In the following, we use capital Latin indices for indices whose range is governed by the discretization, as for example, $I = 1 \dots N_{cp}$, $J = 1 \dots N_{dof}$, with N_{cp} as the number of control points (or shape functions) and N_{dof} as the number of degrees of freedom. The summation convention is also applied to these indices. Assuming a knot vector $\{\xi_1, ..., \xi_{N_{cp}+p+1}\}$, with p being the polynomial degree, the Greville abscissae are defined by

$$\overline{\xi}_I = \frac{\xi_{I+1} + \xi_{I+2} + \ldots + \xi_{I+p}}{p}, \quad \text{for } I = 1, \ldots, N_{cp}. \tag{58}$$

For bivariate B-splines and NURBS, the Greville abscissae are simply obtained by the tensor product of (58) in two parametric directions.

An important advantage of the isogeometric concept especially for shell analysis is that for a wide range of typical shell geometries we obtain an exact geometric description, which means that all the quantities from Section 2 can be computed exactly and in a straightforward way. The NURBS description of the geometry is given by

$$\boldsymbol{r} = N_I(\xi^1, \xi^2)\, \hat{\boldsymbol{x}}_I \tag{59}$$

where $N_I(\xi^1, \xi^2)$ are the bivariate NURBS functions and $\hat{\boldsymbol{x}}_I$ are the control point coordinates. Note that we use single index numbering $I = 1 \dots N_{cp}$, with $N_{cp} = N_{cp1} \times N_{cp2}$ as the total number of control points, and N_{cp1}, N_{cp2} as the numbers of control points in the two parametric directions. The NURBS parametrization naturally provides a curvilinear coordinate system which can be interpreted as the natural shell coordinate system, $(\xi^1, \xi^2) = (\theta^1, \theta^2)$. Accordingly, tangent base vectors are obtained as

$$\boldsymbol{a}_\alpha = N_{I,\alpha}(\xi^1, \xi^2)\, \hat{\boldsymbol{x}}_I \tag{60}$$

where $N_{I,\alpha}(\xi^1, \xi^2) = \partial N_I(\xi^1, \xi^2)/\partial \xi^\alpha$ are simply the natural NURBS derivatives. For the ease of notation, the dependency on (ξ^1, ξ^2) will be skipped in the following. The derivatives of the tangent base vectors are then obtained as

$$\boldsymbol{a}_{\alpha,\beta} = N_{I,\alpha\beta}\, \hat{\boldsymbol{x}}_I \tag{61}$$

$$\boldsymbol{a}_{\alpha,\beta\gamma} = N_{I,\alpha\beta\gamma}\, \hat{\boldsymbol{x}}_I \tag{62}$$

Since third derivatives appear in (62), cubic or higher NURBS need to be used. Having computed (60)-(62), all other geometric quantities can be computed according to equations (2)-(31) in a straightforward manner. It should be noted that the NURBS parametrization does not coincide, in general, with classical parametrizations using, e.g., cylindrical or spherical coordinates. A consequence of this is that, e.g., in a NURBS representation of a cylinder, the metric and curvature coefficients $a_{\alpha\beta}, b_{\alpha\beta}$ are not constant and the Christoffel symbols $\Gamma^\gamma_{\alpha\beta}$ do not vanish, as it would be expected with cylindrical coordinates.

Following the isoparametric concept, we use NURBS also to approximate the unknown variables. For a compact notation, it is useful to consider also generalized displacements $(u_1, u_2, u_3, u_4, u_5) = (v_1, v_2, v_3, w_1, w_2)$. Let us denote by $\hat{u}_J = \hat{u}^I_K$ ($J = 1 \dots N_{dof}$, $I = 1 \dots N_{cp}$, $K = 1 \dots 5$) the Jth global degree of freedom, corresponding to the K-th degree of freedom at control point I. The displacement variables are then approximated as follows (with a slight abuse of notation we use the same symbols for the approximated variables as for the exact ones)

$$v_\alpha = N_I \delta^K_\alpha \hat{u}^I_K = N_I \delta^K_\alpha \hat{u}_J, \tag{63}$$

$$v_3 = N_I \delta_3^K \hat{u}_K^I = N_I \delta_3^K \hat{u}_J, \tag{64}$$

$$w_\alpha = N_I \delta_\alpha^{(K-3)} \hat{u}_K^I = N_I \delta_\alpha^{(K-3)} \hat{u}_J. \tag{65}$$

For the sake of implementation, it is also useful to report Eqs. (63)-(65) in matrix form. Assuming the standard ordering of global degrees of freedom by control points, $\hat{\boldsymbol{u}} = (\hat{u}_1^1\ \hat{u}_2^1\ \hat{u}_3^1\ \hat{u}_4^1\ \hat{u}_5^1\ \hat{u}_1^2\ \hat{u}_2^2\ \hat{u}_3^2\ \hat{u}_4^2\ \hat{u}_5^2\ \ldots)^T$, the matrix form is given by

$$v_\alpha = \begin{pmatrix} N_1 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & 0 & \ldots \\ 0 & N_1 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 & 0 & \ldots \end{pmatrix} \hat{\boldsymbol{u}}, \tag{66}$$

$$v_3 = \begin{pmatrix} 0 & 0 & N_1 & 0 & 0 & 0 & 0 & N_2 & 0 & 0 & \ldots \end{pmatrix} \hat{\boldsymbol{u}}, \tag{67}$$

$$w_\alpha = \begin{pmatrix} 0 & 0 & 0 & N_1 & 0 & 0 & 0 & 0 & N_2 & 0 & \ldots \\ 0 & 0 & 0 & 0 & N_1 & 0 & 0 & 0 & 0 & N_2 & \ldots \end{pmatrix} \hat{\boldsymbol{u}}. \tag{68}$$

An interesting alternative is to order the global degrees of freedom by variable, $\hat{\boldsymbol{u}} = (\hat{u}_1^1\ \hat{u}_1^2\ \ldots\ \hat{u}_2^1\ \hat{u}_2^2\ \ldots\ \hat{u}_3^1\ \hat{u}_3^2\ \ldots\ \hat{u}_4^1\ \hat{u}_4^2\ \ldots\ \hat{u}_5^1\ \hat{u}_5^2\ \ldots)^T$. In this case, the matrices can be constructed conveniently as

$$v_\alpha = \begin{pmatrix} \mathbf{N} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{N} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{pmatrix} \hat{\boldsymbol{u}}, \tag{69}$$

$$v_3 = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{N} & \mathbf{0} & \mathbf{0} \end{pmatrix} \hat{\boldsymbol{u}}, \tag{70}$$

$$w_\alpha = \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{N} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{N} \end{pmatrix} \hat{\boldsymbol{u}}, \tag{71}$$

with the row vectors $\mathbf{N} = (N_1\ N_2\ \ldots)$ and $\mathbf{0} = (0\ 0\ \ldots)$, which both are of length N_{cp}. We note that both (66)-(68) and (69)-(71) are suitable for implementation and both can be obtained from the general form (63)-(65) through the specific relation between the indices I, J, K. For (66)-(68), this relation is given as $J = (I-1)5 + K$, while for (69)-(71), it is $J = (K-1)N_{cp} + I$.

In the following, we will introduce so-called *discretized variables* corresponding to the variables presented in the previous section, which can be directly used for implementation. These *discretized variables* are multidimensional arrays which can be presented in compact form using the indices I, J, K. We begin with the *discretized displacement variables* defined as

$$\mathrm{v}(J,\alpha) = N_I \delta^K_\alpha, \tag{72}$$

$$\mathrm{v3}(J) = N_I \delta^K_3, \tag{73}$$

$$\mathrm{w}(J,\alpha) = N_I \delta^{(K-3)}_\alpha. \tag{74}$$

Clearly, $\mathrm{v}(J,\alpha), \mathrm{v3}(J), \mathrm{w}(J,\alpha)$ simply represent the shape function matrices in (66)-(71). However, the interpretation as *discretized variables* is useful for the stepwise formulation, since it allows us to introduce such discretized versions also for the derivatives and for the strain and stress variables, which can no longer be represented as matrices. The discretized partial derivatives of the displacements are obtained as

$$\mathrm{dv}(J,\alpha,\beta) = N_{I,\beta}\, \delta^K_\alpha, \tag{75}$$

$$\mathrm{dv3}(J,\alpha) = N_{I,\alpha}\, \delta^K_3, \tag{76}$$

$$\mathrm{dw}(J,\alpha,\beta) = N_{I,\beta}\, \delta^{(K-3)}_\alpha. \tag{77}$$

Similarly, we obtain the discretized second partial derivatives as

$$\mathrm{d2v}(J,\alpha,\beta,\gamma) = N_{I,\beta\gamma}\, \delta^K_\alpha, \tag{78}$$

$$\mathrm{d2v3}(J,\alpha,\beta) = N_{I,\alpha\beta}\, \delta^K_3, \tag{79}$$

$$\mathrm{d2w}(J,\alpha,\beta,\gamma) = N_{I,\beta\gamma}\, \delta^{(K-3)}_\alpha. \tag{80}$$

Now we can compute discretized covariant derivatives of the displacements, indicated by a preceding D

$$\mathrm{Dv}(J,\alpha,\beta) = \mathrm{dv}(J,\alpha,\beta) - \mathrm{v}(J,\lambda)\Gamma^\lambda_{\alpha\beta}, \tag{81}$$

$$\mathrm{Dv3}(J,\alpha) = \mathrm{dv3}(J,\alpha), \tag{82}$$

$$\mathrm{Dw}(J,\alpha,\beta) = \mathrm{dw}(J,\alpha,\beta) - \mathrm{w}(J,\lambda)\Gamma^\lambda_{\alpha\beta}, \tag{83}$$

as well as their second covariant derivatives

$$\begin{aligned}\mathrm{D2v}(J,\alpha,\beta,\gamma) &= \mathrm{d2v}(J,\alpha,\beta,\gamma) - \mathrm{dv}(J,\lambda,\gamma)\Gamma^{\lambda}_{\alpha\beta} - \mathrm{v}(J,\lambda)\Gamma^{\lambda}_{\alpha\beta,\gamma} \\ &\quad - \mathrm{Dv}(J,\lambda,\beta)\Gamma^{\lambda}_{\alpha\gamma} - \mathrm{Dv}(J,\alpha,\lambda)\Gamma^{\lambda}_{\beta\gamma}, \qquad (84)\\ \mathrm{D2v3}(J,\alpha,\beta) &= \mathrm{d2v3}(J,\alpha,\beta) - \mathrm{dv3}(J,\lambda)\Gamma^{\lambda}_{\alpha\beta}, \qquad (85)\\ \mathrm{D2w}(J,\alpha,\beta,\gamma) &= \mathrm{d2w}(J,\alpha,\beta,\gamma) - \mathrm{dw}(J,\lambda,\gamma)\Gamma^{\lambda}_{\alpha\beta} - \mathrm{w}(J,\lambda)\Gamma^{\lambda}_{\alpha\beta,\gamma} \\ &\quad - \mathrm{Dw}(J,\lambda,\beta)\Gamma^{\lambda}_{\alpha\gamma} - \mathrm{Dw}(J,\alpha,\lambda)\Gamma^{\lambda}_{\beta\gamma}. \qquad (86)\end{aligned}$$

With the discretized covariant derivatives, we can compute discretized strain variables (32)-(34)

$$\begin{aligned}\mathrm{alpha}(J,\alpha,\beta) &= \frac{1}{2}\left(\mathrm{Dv}(J,\alpha,\beta) + \mathrm{Dv}(J,\beta,\alpha) - 2b_{\alpha\beta}\mathrm{v3}(J)\right), \qquad (87)\\ \mathrm{beta}(J,\alpha,\beta) &= \frac{1}{2}\left(\mathrm{Dw}(J,\alpha,\beta) + \mathrm{Dw}(J,\beta,\alpha)\right. \\ &\quad \left. -b^{\lambda}_{\alpha}\mathrm{Dv}(J,\lambda,\beta) - b^{\lambda}_{\beta}\mathrm{Dv}(J,\lambda,\alpha) + 2b^{\lambda}_{\alpha}b_{\lambda\beta}\mathrm{v3}(J)\right), \qquad (88)\\ \mathrm{gamma}(\mathrm{J},\alpha) &= \mathrm{w}(J,\alpha) + \mathrm{dv3}(J,\alpha) - b^{\lambda}_{\alpha}\mathrm{v}(J,\lambda), \qquad (89)\end{aligned}$$

and their covariant derivatives (52)-(54)

$$\mathrm{Dalpha}(J,\alpha,\beta,\gamma) = \frac{1}{2}\left(\mathrm{D2v}(J,\alpha,\beta,\gamma) + \mathrm{D2v}(J,\beta,\alpha,\gamma) - 2b_{\alpha\beta}|_{\gamma}\mathrm{v3}(J) - 2b_{\alpha\beta}\mathrm{Dv3}(J,\gamma)\right), \qquad (90)$$

$$\begin{aligned}\mathrm{Dbeta}(J,\alpha,\beta,\gamma) &= \frac{1}{2}\left(\mathrm{D2w}(J,\alpha,\beta,\gamma) + \mathrm{D2w}(J,\beta,\alpha,\gamma) - b^{\lambda}_{\alpha}|_{\gamma}\mathrm{Dv}(J,\lambda,\beta)\right. \\ &\quad -b^{\lambda}_{\alpha}\mathrm{D2v}(J,\lambda,\beta,\gamma) - b^{\lambda}_{\beta}|_{\gamma}\mathrm{Dv}(J,\lambda,\alpha) - b^{\lambda}_{\beta}\mathrm{D2v}(J,\lambda,\alpha,\gamma) \\ &\quad \left. +2b^{\lambda}_{\alpha}|_{\gamma}b_{\lambda\beta}\mathrm{v3}(J) + 2b^{\lambda}_{\alpha}b_{\lambda\beta}|_{\gamma}\mathrm{v3}(J) + 2b^{\lambda}_{\alpha}b_{\lambda\beta}\mathrm{Dv3}(J,\gamma)\right), \qquad (91)\\ \mathrm{Dgamma}(J,\alpha,\beta) &= \mathrm{Dw}(J,\alpha,\beta) + \mathrm{D2v3}(J,\alpha,\beta) + b^{\lambda}_{\alpha}|_{\beta}\mathrm{v}(J,\lambda) + b^{\lambda}_{\alpha}\mathrm{Dv}(J,\lambda,\beta). \qquad (92)\end{aligned}$$

Subsequently, we compute discretized stress resultants (36)-(38)

$$\begin{aligned}\mathrm{nt}(J,\alpha,\beta) &= h\, H^{\alpha\beta\lambda\mu}\,\mathrm{alpha}(J,\lambda,\mu), \qquad (93)\\ \mathrm{m}(J,\alpha,\beta) &= \frac{h^3}{12}\, H^{\alpha\beta\lambda\mu}\,\mathrm{beta}(J,\lambda,\mu), \qquad (94)\\ \mathrm{q}(J,\alpha) &= Gha^{\alpha\lambda}\,\mathrm{gamma}(J,\lambda), \qquad (95)\end{aligned}$$

and their covariant derivatives (55)-(57)

$$\mathrm{Dnt}(J,\alpha,\beta,\gamma) = h\, H^{\alpha\beta\lambda\mu}\, \mathrm{Dalpha}(J,\lambda,\mu,\gamma), \tag{96}$$

$$\mathrm{Dm}(J,\alpha,\beta,\gamma) = \frac{h^3}{12}\, H^{\alpha\beta\lambda\mu}\, \mathrm{Dbeta}(J,\lambda,\mu,\gamma), \tag{97}$$

$$\mathrm{Dq}(J,\alpha,\beta) = Gha^{\alpha\lambda}\, \mathrm{Dgamma}(J,\lambda,\beta). \tag{98}$$

Finally, we can collocate the equilibrium equations (40)-(42) as

$$\mathrm{Dnt}(J,\alpha,\lambda,\lambda) - b^{\alpha}_{\mu}|_{\lambda}\mathrm{m}(J,\lambda,\mu) - b^{\alpha}_{\mu}\mathrm{Dm}(J,\lambda,\mu,\lambda) - b^{\alpha}_{\lambda}\mathrm{q}(J,\lambda) = -p^{\alpha}, \tag{99}$$

$$b_{\lambda\mu}\tilde{\mathrm{n}}(J,\lambda,\mu) - b^{\mu}_{\rho} b_{\lambda\mu}\mathrm{m}(J,\lambda,\rho) + \mathrm{Dq}(J,\lambda,\lambda) = -p^{3}, \tag{100}$$

$$\mathrm{Dm}(J,\alpha,\lambda,\lambda) - \mathrm{q}(J,\alpha) = -c^{\alpha}\,. \tag{101}$$

Equations (72)-(101) represent the detailed computer implementation of our approach. Summation over repeated indices can be conveniently done by loops, however, symmetries of several variables may be exploited for the sake of computational efficiency. It should be noted that also the geometric quantities like $\Gamma^{\lambda}_{\alpha\beta}$, $b_{\alpha\beta}$ in these equations represent multidimensional arrays in the implementation. However, we kept them in symbolic notation for better readability of the equations. The left hand sides of Eqs. (99)-(101) represent five rows of the stiffness matrix, collocated at each collocation point, with the free index J corresponding to the columns.

In the same way, we also derive the discretized versions of the boundary equilibrium equations, which are collocated at Neumann boundaries. It should be noted that Eqs. (43)-(47) assume smooth boundaries, while practical problems typically exhibit also sharp corners, where the boundary tangent and normal vectors $\boldsymbol{t}, \boldsymbol{u}$ are not uniquely defined. According to [33], the tangent and normal vectors at these locations are taken as the average of the respective vectors from the two edges meeting at the corner.

5. Numerical tests

In this section we test the proposed formulation on a set of well-known benchmark examples, consisting of the Scordelis-Lo roof from [60], the clamped hemispherical cap and the partly clamped hyperbolic paraboloid from [46, 61], and the cylindrical shell strip from [15].

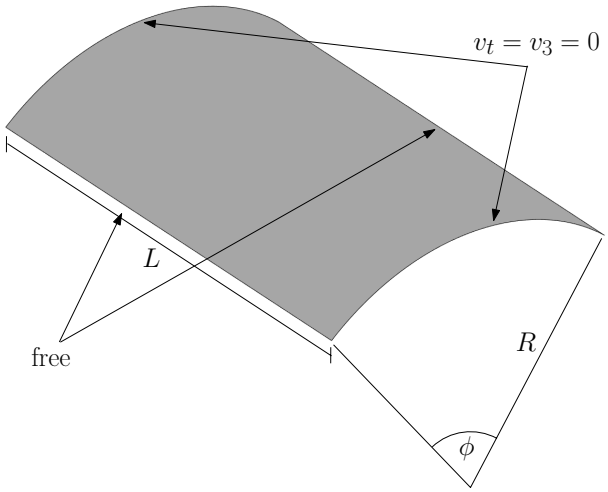


Figure 1: Scordelis-Lo roof. Geometry and boundary conditions.

All examples consist of geometries that can be modeled exactly by NURBS and, furthermore, they cover the three different classes of parabolic, elliptic, and hyperbolic surfaces. Since all examples involve rather thin shells and a standard displacement-based formulation is employed, locking is to be expected. In the following, we perform convergence studies for different polynomial degrees ranging from $p = 3$ to $p = 8$ with maximum inter-element continuity C^{p-1}. Furthermore, two of the examples exhibit boundary layers. As demonstrated in [62], the size of the boundary layers typically scales with $\sqrt{l \cdot h}$, where h is the shell thickness, and l is a characteristic length, typically chosen as the length or radius of a shell. In order to properly resolve the boundary layers, we will use graded meshes, which are more refined on the boundaries, as shown in detail in the respective examples.

5.1. Scordelis-Lo roof

The Scordelis-Lo roof is one of the problems of the so-called *shell obstacle course* [60]. It consists of a cylindrical ($K = 0$) section with radius $R = 25$, opening angle $\phi = 80°$, length $L = 50$, and thickness $h = 0.25$, as depicted in Figure 1. The curved edges are supported by rigid diaphragms, while the straight edges are free, and the shell is subjected to self-weight with $p_z = 90$ per unit area. The material parameters are given by the Young's modulus $E = 4.32 \cdot 10^8$ and Poisson's ratio $\nu = 0.0$. As reference solution, the vertical displacement at the midpoint of the free edges is given as $v_z = -0.3024$ [60]. First, we perform uniform mesh refinement with $[4, 8, 16, 32, 64]^2$ elements, and the results are displayed in

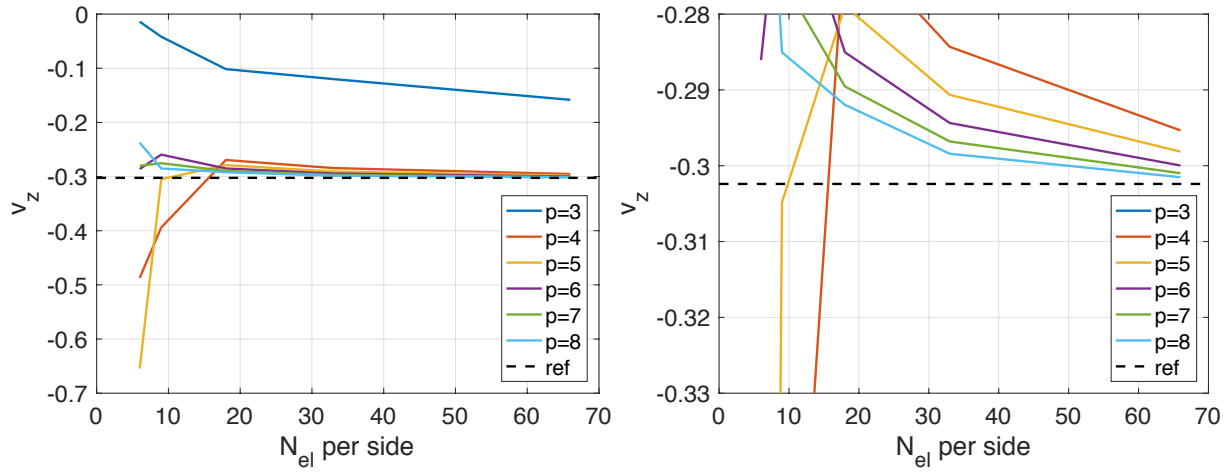


Figure 2: Scordelis-Lo roof. Convergence study with uniform refinement. In the right figure, the $y-$axis has been rescaled for a close-up view.

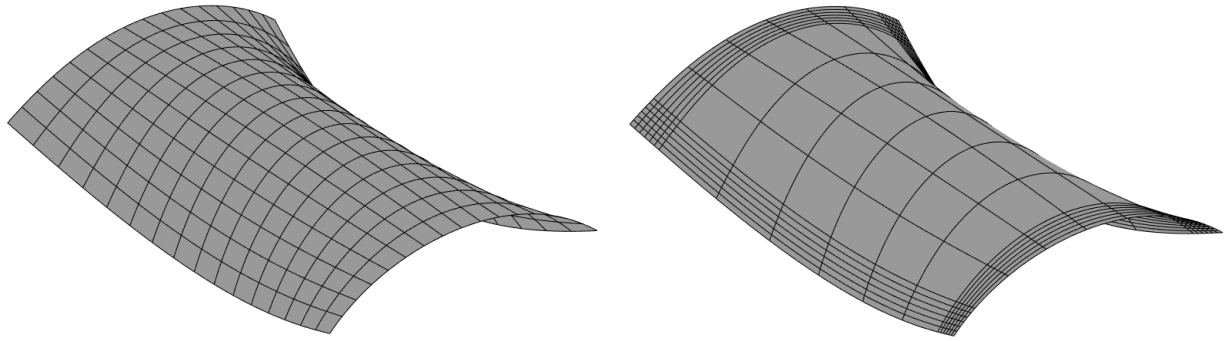


Figure 3: Scordelis-Lo roof. Uniform and boundary refined meshes, displayed on the deformed geometry (deformation scaled for visualization by a factor of 20).

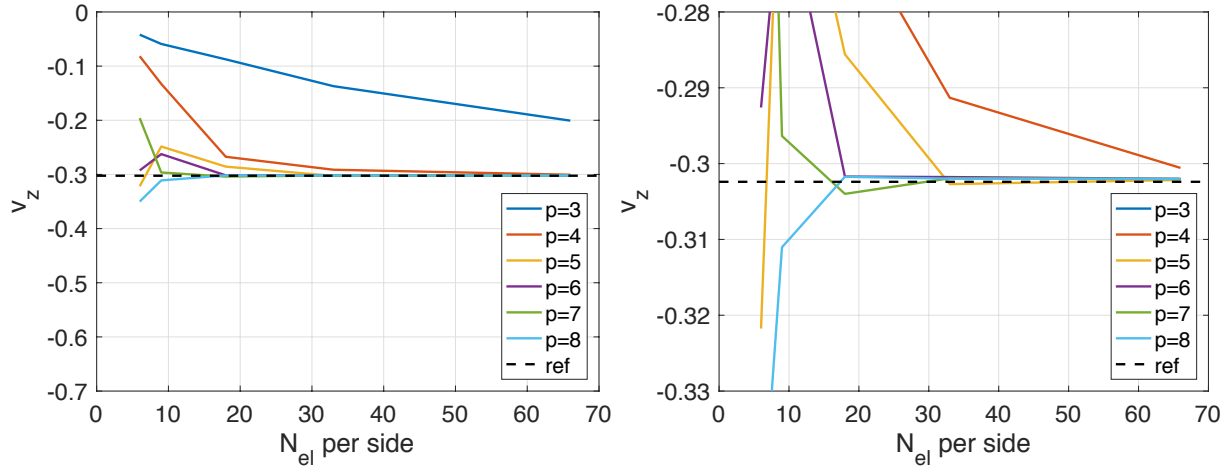
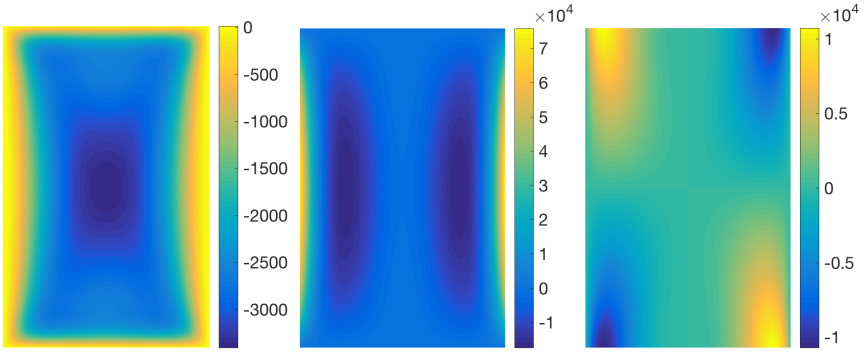
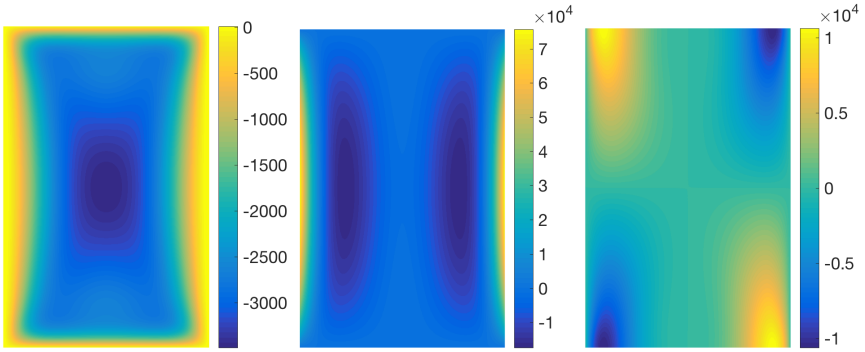


Figure 4: Scordelis-Lo roof. Convergence study with boundary refinement. In the right figure, the $y-$axis has been rescaled for a close-up view.

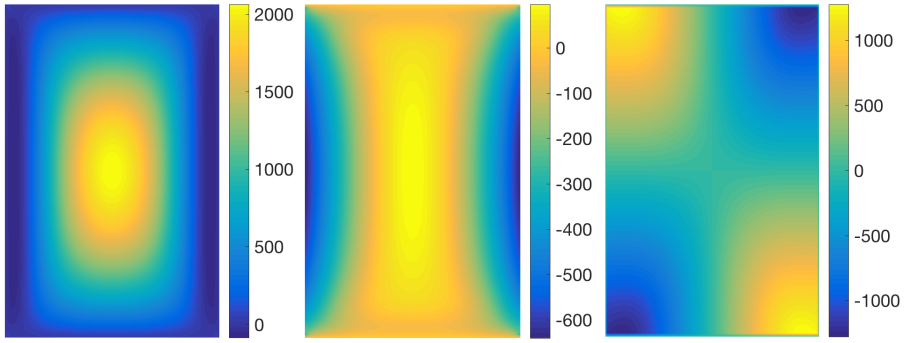


(a) n_{11} (b) n_{22} (c) n_{12}

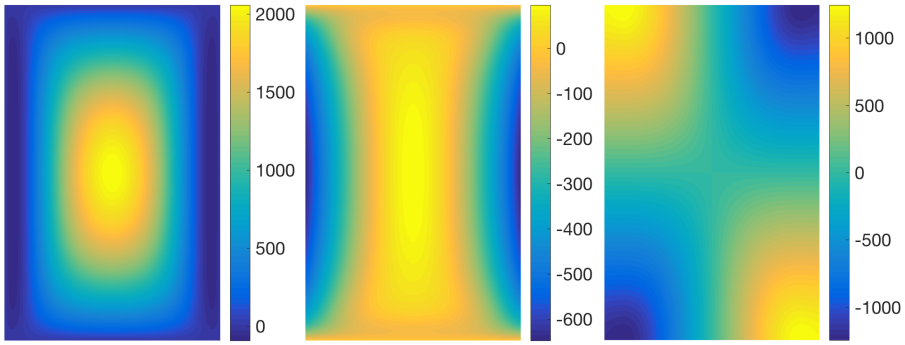


(d) n_{11}^{IGA-KL} (e) n_{22}^{IGA-KL} (f) n_{12}^{IGA-KL}

Figure 5: Scordelis-Lo roof, membrane forces. (a)-(c) show results from the presented collocation approach, (d)-(f) are obtained with isogeometric Kirchhoff-Love shell analysis for comparison.



(a) m_{11} (b) m_{22} (c) m_{12}



(d) m_{11}^{IGA-KL} (e) m_{22}^{IGA-KL} (f) m_{12}^{IGA-KL}

Figure 6: Scordelis-Lo roof, bending and twisting moments. (a)-(c) show results from the presented collocation approach, (d)-(f) are obtained with isogeometric Kirchhoff-Love shell analysis for comparison.

Figure 2. As expected, strong locking can be observed for $p = 3$, with the results being far from the reference solution even for the finest mesh. For $p = 4$ to $p = 8$, the results are significantly better, however, they converge very slowly, even for the highest degrees. In the right plot in Figure 2, the $y-$axis is rescaled such that this effect can clearly be seen. This behavior is due to boundary layers, which are not resolved properly by the uniform mesh refinement. Therefore, we adopt a graded mesh refinement, where we first insert additional knots at $\sqrt{L \cdot h}$ from the boundaries and then perform uniform mesh refinement of the existing elements. In Figure 3, examples for uniform and boundary refined meshes are depicted. Figure 4 shows the convergence curves obtained with boundary refinement. The results of $p = 3$ are still far away from convergence and also $p = 4$ is not fully converged as can be seen in the close-up view in the right figure. However, very good results are obtained for $p \geq 5$, with convergence after few steps for $p \geq 6$. It should be noted that on the x-axis, the total number of elements N_{el} per side is displayed, which means that for the same N_{el} the mesh is much coarser in the interior for the boundary refined cases, as can be seen also in Figure 3. The converged solution is obtained as $v_z = -0.3020$, which is slightly lower than the reference value. Furthermore, we investigate the quality of stress resultants obtained with this method. Since no reference values for stresses or stress resultants are provided in [60], we solve the problem via isogeometric Kirchhoff-Love shell analysis [6] for comparison. In Figure 5, the membrane force components n_{11}, n_{22}, n_{12} are depicted, with the indices 1 and 2 corresponding to the circumferential and longitudinal directions, respectively. Subfigures (a)-(c) show the results obtained with the presented collocation approach on the finest mesh and $p = 8$, while (d)-(f) are the results for comparison, obtained with isogeometric Kirchhoff-Love shell analysis on a 64×64 mesh with $p = 5$. Very good agreement in the results can be observed. Figure 6 depicts the corresponding results for the moments m_{11}, m_{22}, m_{12}, and also here, very good agreement can be observed.

5.2. Clamped hemispherical cap

The second example is taken from [61] and consists of a clamped hemispherical cap ($K > 0$) under a sinusoidal external pressure loading. We model only one quarter of the geometry imposing symmetry conditions on the respective boundaries, as depicted in Figure

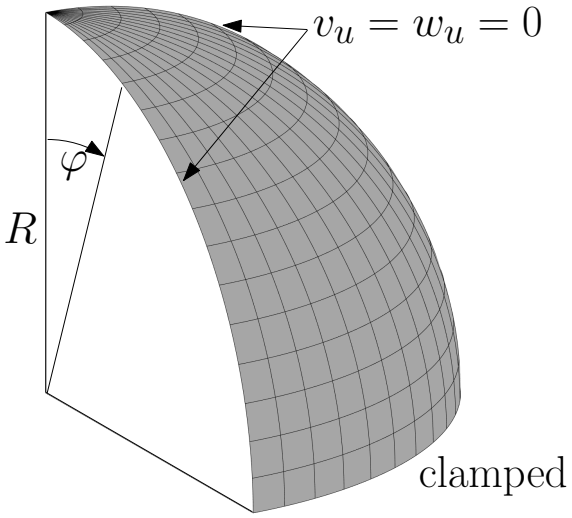


Figure 7: Clamped hemispherical cap. Geometry and boundary conditions. A quarter of the problem with symmetry conditions is modeled.

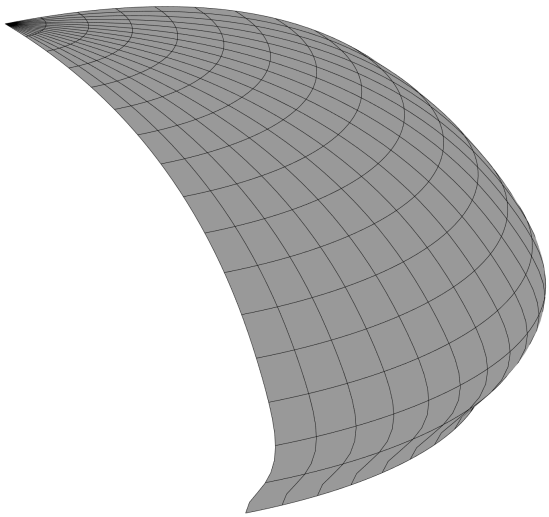


Figure 8: Clamped hemispherical cap. Deformation (scaled for visualization by a factor of $3 \cdot 10^4$).

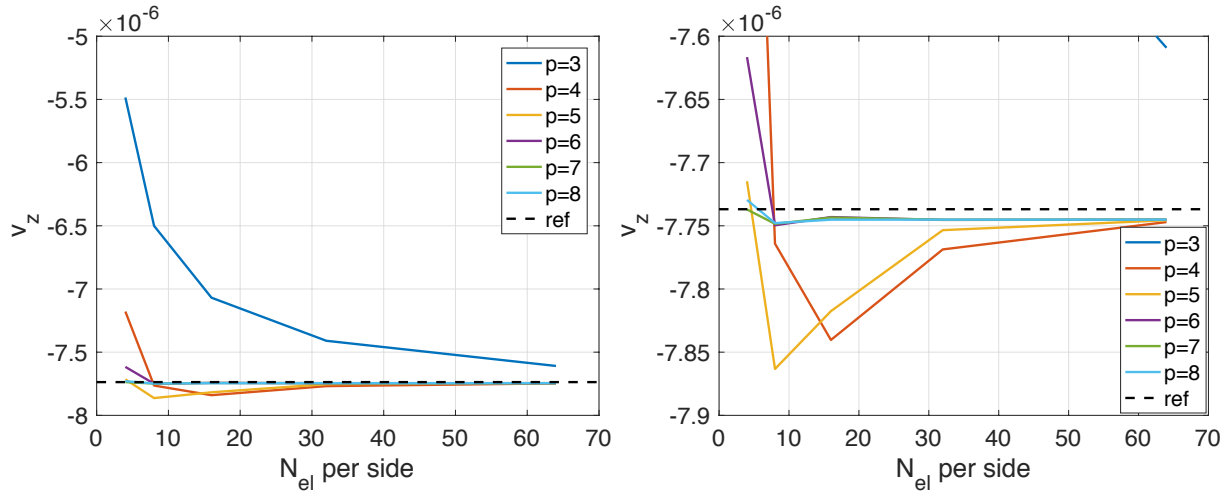


Figure 9: Clamped hemispherical cap. Convergence study. In the right figure, the $y-$axis has been rescaled for a close-up view.

7. The geometric and material properties are given as $R = 1$, $h = 0.01$, $E = 2 \cdot 10^{11}$, and $\nu = 0.3$. The pressure load is given as a function of the polar angle, $p_3(\varphi) = p_0 h \cos(2\varphi)$, with $p_0 = 10^6$. As reference solution, the vertical displacement at the pole is given as $v_z = -7.73688 \cdot 10^{-6}$ [61].

Special consideration has to be given to the imposition of boundary conditions at the pole for avoiding numerical problems due to the geometric singularity. While we have Dirichlet conditions for $v_1 = v_2 = w_1 = w_2 = 0$, the deformation in v_3 is free and the corresponding Neumann condition needs to be imposed. Considering θ^1 as the azimuth direction and θ^2 as the polar direction, we need to impose the Neumann condition $q^2 = 0$ at the pole, which means on all collocation points of this collapsed edge. Due to the singularity, we obtain $a_{11} = b_{11} = 0$ there, while a_{22} and b_{22} are still finite and non-zero. Using that we further have $a_{12} = b_{12} = 0$, the formula for the boundary shear force can be significantly simplified and finally reads as

$$q^2 = \frac{Gh}{a_{22}} \left(w_2 + v_{3,2} + \frac{b_{22}}{a_{22}} v_2 \right), \tag{102}$$

which we collocate on all collocations points coinciding in the pole.

For this example we did not find any influence of boundary layers and the best results were obtained with uniform mesh refinement. Figure 8 depicts the deformed configuration and Figure 9 shows the results of the convergence study. The converged solution is obtained

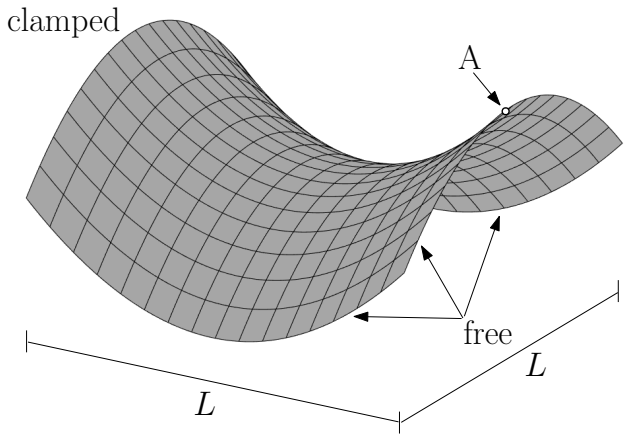


Figure 10: Partly clamped hyperbolic paraboloid (isolines are plotted for visualization but do not represent an analysis mesh).

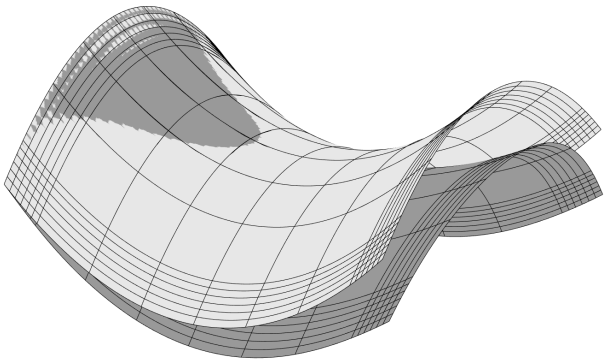


Figure 11: Partly clamped hyperbolic paraboloid. Undeformed (light shading) and deformed (dark shading) configurations with boundary refinement. The deformation has been scaled for visualization by a factor of $1.5 \cdot 10^3$.

as $v_z = -7.74513 \cdot 10^{-6}$, which is slightly higher than the reference value from [61]. Again, we observe strong locking for $p = 3$, which does not reach the converged value within the meshes considered, while all other degrees perform quite well, especially for $p \geq 6$.

5.3. Partly clamped hyperbolic paraboloid

This example is also taken from [61]. It has a hyperbolic geometry ($K < 0$), which is described by $z = x^2 - y^2$, $(x, y) \in [-L/2, L/2]^2$, as depicted in Figure 10. The geometric and material parameters are given as $L = 1$, $h = 0.01$, $E = 2 \cdot 10^{11}$, $\nu = 0.3$. The shell is clamped along the side $x = -L/2$ and subjected to self-weight loading given as $p_z = -8000 \cdot h$. As reference solution, the vertical displacement at point A ($x = L/2$,

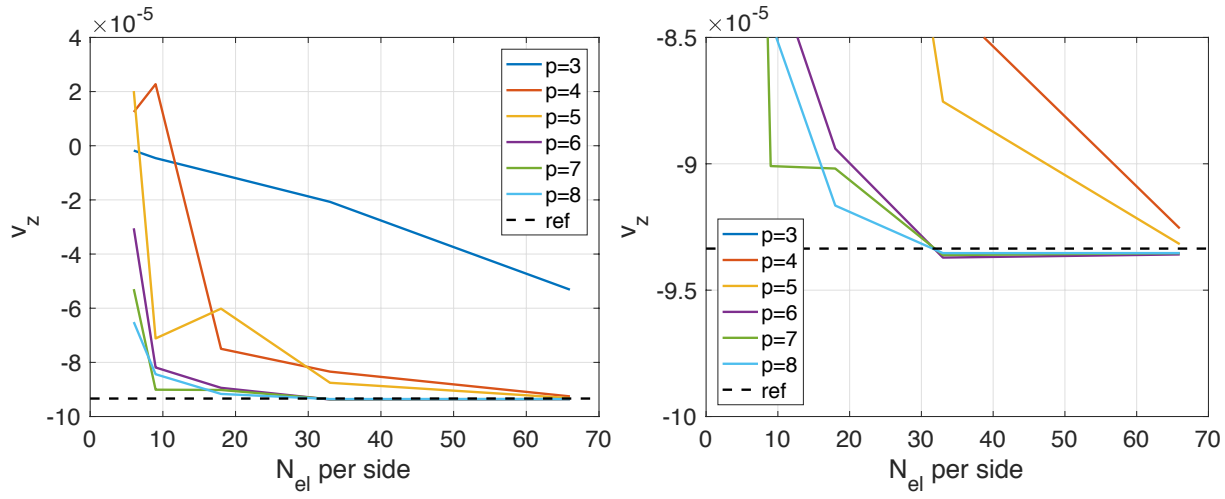


Figure 12: Partly clamped hyperbolic paraboloid. Convergence study. In the right figure, the $y-$axis has been rescaled for a close-up view.

$y = 0$) is given as $v_z = -9.3355 \cdot 10^{-5}$ [61]. This problem exhibits significant boundary layers and we adopt again a graded refinement scheme as described in Section 5.1, with higher refinement in a width of $\sqrt{L \cdot h}$ from the edges. Figure 11 shows a boundary refined mesh on the undeformed and deformed configurations. In Figure 12, the convergence curves are depicted. The converged solution is obtained as $v_z = -9.3533 \cdot 10^{-5}$, which is slightly higher than the reference value. Similar to the previous examples, $p = 3$ performs very badly, but also $p = 4$ and $p = 5$ do not reach convergence within the considered range of meshes. For $p \geq 6$, however, very good convergence can be observed again.

5.4. Cylindrical shell strip

The final example is taken from [15]. It consists of a cylindrical shell strip, which is clamped on one side and subjected to a constant line load in radial direction on the opposite free edge, see Figure 13. The geometric and material parameters are given as $R = 10$, $W = 1$, $E = 10^3$, $\nu = 0$. The applied load is scaled with the shell thickness $F = 0.1 \cdot h^3$ and different thickness values ranging from thick to very thin shells are considered. As reference solution, the radial displacement under the load has been computed in [15] according to Euler-Bernoulli beam theory as approximately $v_r = 0.942$. This example is often used to study membrane locking, but it should be noted that both membrane and shear locking are present. We solve the problem for various thickness values with the slenderness R/h ranging

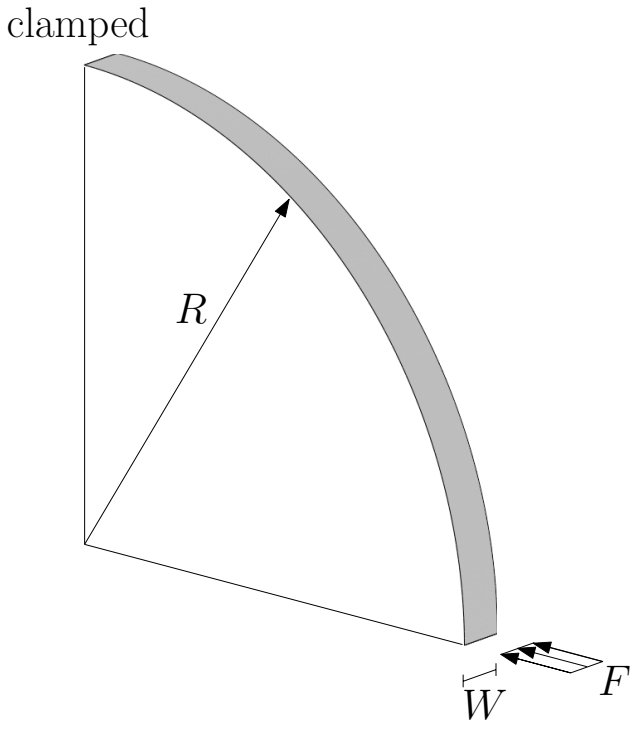


Figure 13: Cylindrical shell strip. Problem setup.

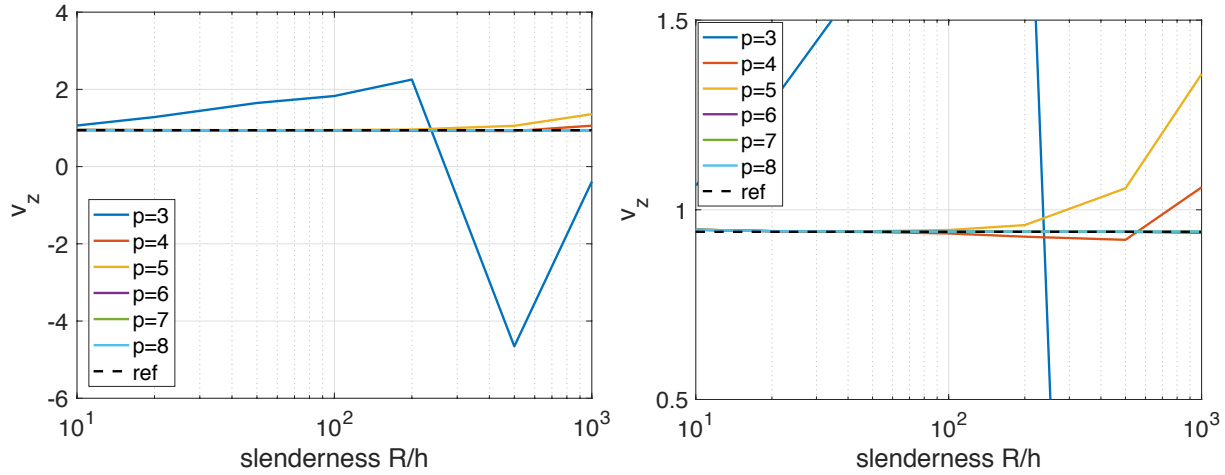


Figure 14: Cylindrical shell strip. Slenderness study. In the right figure, the $y-$axis has been rescaled for a close-up view.

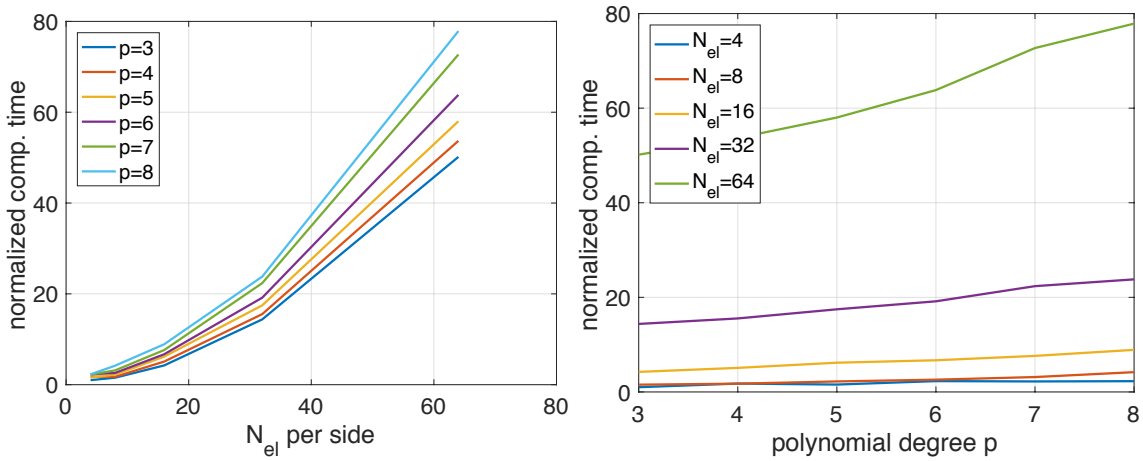


Figure 15: Clamped hemispherical cap. Study on the computational time for assembly, normalized by the result for the coarsest mesh. Left: Results gathered by polynomial degree. Right: Results gathered by mesh size.

from 10 to 1000. The mesh is chosen such that the number of collocation points in the long direction is 30 for all polynomial degrees, i.e., $30 - p$ elements are used. In Figure 14, the convergence curves are depicted. It can be seen that $p = 3$ performs very badly over the whole range of slenderness, $p = 4$ and $p = 5$ are accurate until a slenderness of 100 but deviate from the reference solution for higher values, while the results for $p \geq 6$ are very good even for the very slender cases.

5.5. A short discussion on locking, polynomial degree, and computational efficiency

It is a general feature of IGA-C that primal formulations may suffer from the same locking problems as corresponding Galerkin formulations, which has been observed already in the context of shear-deformable beams, rods, and plates [34, 36–40]. Corresponding locking-free collocation methods can be obtained, e.g., by adopting mixed formulations [34, 36, 38–40], similar to mixed Galerkin methods. In this context it is also worth noting that while the source of locking in IGA-C is the same as in Galerkin methods, namely the unbalance of the discrete approximation spaces, the effects can be different. In contrast to Galerkin methods, the displacements affected by locking in IGA-C do not necessarily tend towards zero in the thin limit, but can also behave rather oscillatory, as can be seen in the results for $p = 3$ in Figure 14. Obviously, the term *locking* is somewhat inappropriate for this behavior from a phenomenological point of view, but it is kept due to the analogy to locking in Galerkin

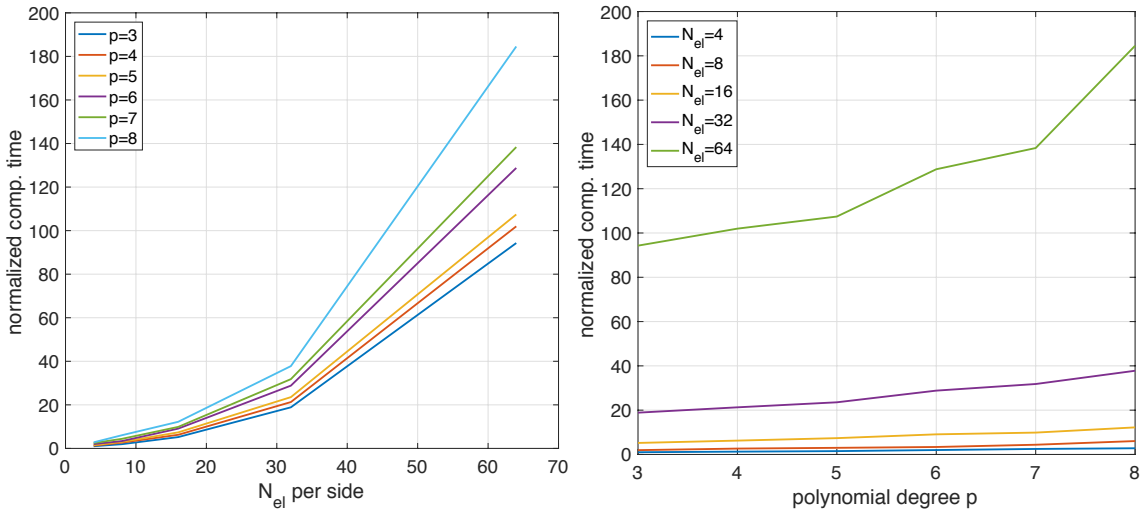


Figure 16: Clamped hemispherical cap. Study on the total computational time, normalized by the result for the coarsest mesh. Left: Results gathered by polynomial degree. Right: Results gathered by mesh size.

methods.

As in Galerkin methods, the effects of locking in IGA-C decrease with increasing polynomial degree. For the presented shell formulation, the numerical tests indicate that polynomial degrees $p > 5$ are necessary to obtain good results without excessively fine meshes. However, the important difference between IGA-C and Galerkin methods is that for the latter, the computational effort for the assembly increases exponentially with p due to numerical quadrature, while in IGA-C only one collocation point per degree of freedom (which is approximately one point per element) is needed, independently of the polynomial degree. For problems affected by locking, like the presented shell formulation, this makes the use of high polynomial degrees a simple and efficient way of avoiding locking.

In the following, we perform a study on the computational time spent on the assembly of the system matrices for different polynomial degrees and meshes. Figure 15 shows exemplarily the results for the hemisphere problem. In the left figure, the curves represent the different polynomial degrees with the number of elements per side on the x-axis, while in the right figure the curves represent the different meshes with the polynomial degree on the x-axis. It can be seen that with mesh refinement, the computational cost increases exponentially, while increasing the polynomial degree has a rather small impact, especially for the coarser meshes. But even for the finest mesh, going from $p = 3$ up to $p = 8$ increases

the computational time by a factor of less than two. Since the total computational cost, including the time for solving the equation system, depends not only on the number of collocation points but also on factors like the bandwidth, which, in turn, depend on p, we also measure the total computational time, comprising both assembly and solving. The results are depicted in Figure 16 and support what has been observed in Figure 15, namely, that the computational cost depends mainly on the mesh size rather than on the polynomial degree. These results confirm that high polynomial degrees are preferable in IGA-C, in particular for the presented case of a primal formulation for Reissner-Mindlin shells.

6. Conclusions

In this paper, we applied the concept of isogeometric collocation to the Reissner-Mindlin shell problem. We started by recalling the necessary background on differential geometry and the governing equations of shear-deformable shells. We showed that the classical approach of expressing the equilibrium equations in terms of primal kinematic variables is not suitable in the case of shells due to the cumbersome form that the expanded equations assume. Therefore, we derived the formulation in a stepwise approach by creating step-by-step discretized objects that can be efficiently used to construct the discretized forms of the governing equations by following closely their expressions in the continuous form. In comparison with the traditional approach, the stepwise formulation offers enormous benefits in terms of both implementation difficulty and computational efficiency. As collocation points, we adopted the standard Greville abscissae. Convergence studies have been performed on different benchmark problems which cover the three different classes of parabolic, elliptic, and hyperbolic shells, and which include important effects like locking and boundary layers. Similar to what has been observed in [40] for Reissner-Mindlin plates, boundary layers significantly affect the convergence behavior if uniform meshes are used, but results can easily be improved by using graded meshes, which are more refined at the boundaries. The results are also strongly affected by locking for lower polynomial degrees, but these effects become insignificant for high polynomial degrees. In our numerical tests, very good results were obtained for $p > 5$. Furthermore, we presented an indicative study on the computational costs, which suggests the use of high polynomial degrees for both accuracy and computational efficiency. In light of

these results we believe that the proposed displacement-based formulation with sufficiently high polynomial degrees presents an efficient and accurate method for a wide range of shell problems. Nevertheless, the development of a locking-free method, e.g. by using a mixed formulation, is of interest as well, and is planned as future research.

Acknowledgements

J. Kiendl was partially supported by the Onsager fellowship program of the Norwegian University of Science and Technology (NTNU). L. De Lorenzis was partially supported by the DFG Priority Program SPP 1748 "Reliable Simulation Techniques in Solid Mechanics". This support is gratefully acknowledged.

Appendix A. Direct displacement-based approach

In this appendix, we present an exemplary part of the equations which are obtained with the *direct approach*, where the equilibrium equations are expressed directly in terms of the primal variables. The equations have been derived with the help of Mathematica by Wolfram [50] together with EinS [51], a specific Mathematica package able to manage indexed objects with the Einstein summation convention. Since the full equations span several pages, we only present here the first component of the first term of Eq. (40), i.e., $\tilde{n}^{1\lambda}|_{\lambda}$.

The fully expanded form reduced to the first and second fundamental forms as well as to the Christoffel symbols and their derivatives is given is given in Eq. (A.1). It should be observed that the selected term involves only the stretching strain tensor $\alpha_{\alpha\beta}$, therefore only three primal kinematic variables appear[1]. Moreover, the kinematic variables appearing in such a large number of terms still need to be discretized. We have coded all the required functions in order to obtain the full system of collocated equations symbolically, but the computational time became too high making this approach completely impractical.

[1]Note that in Eq. (A.1) a slightly different notation for the partial derivatives is used, where $\partial v_{\alpha\beta}$ and $\partial^2 v_{\alpha\beta\gamma}$ correspond to $v_{\alpha,\beta}$ and $v_{\alpha,\beta\gamma}$, respectively. The same applies to v_3, $\Gamma^{\alpha}_{\beta\gamma}$ and $b_{\alpha\beta}$.

$$\tilde{n}^{1\lambda}|_\lambda = \frac{Eh}{-1+\nu^2}\Big(-2\partial^2 v_{111}a^{11^2} + 6\partial v_{11}\Gamma^1_{11}a^{11^2} + 2\partial v_{12}\Gamma^2_{11}a^{11^2} + 4\partial v_{21}\Gamma^2_{11}a^{11^2} - 4\partial^2 v_{112}a^{11}a^{12}$$
$$-2\partial^2 v_{211}a^{11}a^{12} + 4b_{12}\partial v_{31}a^{11}a^{12} + 4\partial v_{12}\Gamma^1_{11}a^{11}a^{12} + 2\partial v_{21}\Gamma^1_{11}a^{11}a^{12} + 12\partial v_{11}\Gamma^1_{12}a^{11}a^{12}$$
$$+6\partial v_{22}\Gamma^2_{11}a^{11}a^{12} + 4\partial v_{12}\Gamma^2_{12}a^{11}a^{12} + 8\partial v_{21}\Gamma^2_{12}a^{11}a^{12} - \partial^2 v_{122}a^{12^2} - \nu\partial^2 v_{122}a^{12^2} - 3\partial^2 v_{212}a^{12^2}$$
$$+\nu\partial^2 v_{212}a^{12^2} + 2b_{22}\partial v_{31}a^{12^2} - 2\nu b_{22}\partial v_{31}a^{12^2} + 2b_{12}\partial v_{32}a^{12^2} + 2\nu b_{12}\partial v_{32}a^{12^2} + 5\partial v_{12}\Gamma^1_{12}a^{12^2}$$
$$+\nu\partial v_{12}\Gamma^1_{12}a^{12^2} + 3\partial v_{21}\Gamma^1_{12}a^{12^2} - \nu\partial v_{21}\Gamma^1_{12}a^{12^2} + 4\partial v_{11}\Gamma^1_{22}a^{12^2} + 8\partial v_{22}\Gamma^2_{12}a^{12^2} + \partial v_{12}\Gamma^2_{22}a^{12^2}$$
$$+\nu\partial v_{12}\Gamma^2_{22}a^{12^2} + 3\partial v_{21}\Gamma^2_{22}a^{12^2} - \nu\partial v_{21}\Gamma^2_{22}a^{12^2} + 2b_{11}a^{11}(\partial v_{31}a^{11} + \partial v_{32}a^{12}) - \partial^2 v_{122}a^{11}a^{22}$$
$$+\nu\partial^2 v_{122}a^{11}a^{22} - \partial^2 v_{212}a^{11}a^{22} - \nu\partial^2 v_{212}a^{11}a^{22} + 2\nu b_{22}\partial v_{31}a^{11}a^{22} + 2b_{12}(1-\nu)\partial v_{32}a^{11}a^{22}$$
$$+3\partial v_{12}\Gamma^1_{12}a^{11}a^{22} - \nu\partial v_{12}\Gamma^1_{12}a^{11}a^{22} + \partial v_{21}\Gamma^1_{12}a^{11}a^{22} + \nu\partial v_{21}\Gamma^1_{12}a^{11}a^{22} + 2\partial v_{11}\Gamma^1_{22}a^{11}a^{22}$$
$$+4\partial v_{22}\Gamma^2_{12}a^{11}a^{22} + \partial v_{12}\Gamma^2_{22}a^{11}a^{22} - \nu\partial v_{12}\Gamma^2_{22}a^{11}a^{22} + \partial v_{21}\Gamma^2_{22}a^{11}a^{22} + \nu\partial v_{21}\Gamma^2_{22}a^{11}a^{22}$$
$$-2\partial^2 v_{222}a^{12}a^{22} + 2b_{22}\partial v_{32}a^{12}a^{22} + 4\partial v_{12}\Gamma^1_{22}a^{12}a^{22} + 2\partial v_{21}\Gamma^1_{22}a^{12}a^{22} + 6\partial v_{22}\Gamma^2_{22}a^{12}a^{22}\Big)$$
$$-\frac{2Eh}{-1+\nu^2}\Big(a^{11^2}\big(-\partial b_{111} + 2b_{12}\Gamma^2_{11}\big) + a^{11}a^{12}\big(-2\partial b_{121} - \partial b_{112} + 2b_{12}\Gamma^1_{11} + 2b_{22}\Gamma^2_{11} + 4b_{12}\Gamma^2_{12}\big)$$
$$+a^{12^2}\big(\nu\partial b_{221} - \partial b_{221} - \partial b_{122} - \nu\partial b_{122} + 3b_{12}\Gamma^1_{12}a^{12^2} - \nu b_{12}\Gamma^1_{12} + 3b_{22}\Gamma^2_{12} - \nu b_{22}\Gamma^2_{12} + b_{12}\Gamma^2_{22} + \nu b_{12}\Gamma^2_{22}\big)$$
$$+a^{11}a^{22}\big(-\nu\partial b_{221} - \partial b_{122} + \nu\partial b_{122} + b_{12}\Gamma^1_{12} + \nu b_{12}\Gamma^1_{12} + b_{22}\Gamma^2_{12} + \nu b_{22}\Gamma^2_{12} + b_{12}\Gamma^2_{22} - \nu b_{12}\Gamma^2_{22}\big)$$
$$+a^{12}a^{22}\big(2b_{12}\Gamma^1_{22} - \partial b_{222} + 2b_{22}\Gamma^2_{22}\big) + b_{11}(2\Gamma^1_{11}a^{11^2} + 4\Gamma^1_{12}a^{11}a^{12} + \Gamma^1_{22}((1+\nu)a^{12^2} + (1-\nu)a^{11}a^{22}))\Big)v_3$$
$$+\frac{2Eh}{-1+\nu^2}\Big(a^{11^2}\big(\partial\Gamma^1_{111} - 2{\Gamma^1_{11}}^2 - 2\Gamma^1_{12}\Gamma^2_{11}\big) + 2\partial\Gamma^1_{121}a^{11}a^{12} + a^{11}a^{12}\big(\partial\Gamma^1_{112} - 2\Gamma^1_{22}\Gamma^2_{11} - 4\Gamma^1_{12}\Gamma^2_{12}\big)$$
$$+a^{12^2}\big(\partial\Gamma^1_{221} - \nu\partial\Gamma^1_{221} + \partial\Gamma^1_{122} + \nu\partial\Gamma^1_{122} - 3{\Gamma^1_{12}}^2 + \nu{\Gamma^1_{12}}^2a^{12^2} - 3\Gamma^1_{22}\Gamma^2_{12} + \nu\Gamma^1_{22}\Gamma^2_{12} - \Gamma^1_{12}\Gamma^2_{22} - \nu\Gamma^1_{12}\Gamma^2_{22}\big)$$
$$a^{11}a^{22}\big(+\nu\partial\Gamma^1_{221} + \partial\Gamma^1_{122} - \nu\partial\Gamma^1_{122} - {\Gamma^1_{12}}^2 - \nu{\Gamma^1_{12}}^2 - \Gamma^1_{22}\Gamma^2_{12} - \nu\Gamma^1_{22}\Gamma^2_{12} - \Gamma^1_{12}\Gamma^2_{22} + \nu\Gamma^1_{12}\Gamma^2_{22}\big)$$
$$+\partial\Gamma^1_{222}a^{12}a^{22} - 2\Gamma^1_{12}\Gamma^1_{22}a^{12}a^{22} - 2\Gamma^1_{22}\Gamma^2_{22}a^{12}a^{22} - \Gamma^1_{11}(6\Gamma^1_{12}a^{11}a^{12} + \Gamma^1_{22}((1+\nu)a^{12^2} + (1-\nu)a^{11}a^{22}))\Big)v_1$$
$$-\frac{2Eh}{-1+\nu^2}\Big(a^{11^2}\big(-\partial\Gamma^2_{111} + 2\Gamma^2_{11}\Gamma^2_{12}\big) + a^{11}a^{12}\big(-2\partial\Gamma^2_{121} - \partial\Gamma^2_{112} + 4\Gamma^1_{12}\Gamma^2_{11} + 4{\Gamma^2_{12}}^2 + 2\Gamma^2_{11}\Gamma^2_{22}\big)$$
$$+a^{12^2}\big(-\partial\Gamma^2_{221} + \nu\partial\Gamma^2_{221} - \partial\Gamma^2_{122} - \nu\partial\Gamma^2_{122} + \Gamma^1_{22}\Gamma^2_{11} + \nu\Gamma^1_{22}\Gamma^2_{11} + 3\Gamma^1_{12}\Gamma^2_{12} - \nu\Gamma^1_{12}\Gamma^2_{12}$$
$$+4\Gamma^2_{12}\Gamma^2_{22}\big) + 2\Gamma^1_{11}a^{11}(\Gamma^2_{11}a^{11} + \Gamma^2_{12}a^{12}) + ((-\nu\partial\Gamma^2_{221} + (-1+\nu)\partial\Gamma^2_{122} + \Gamma^1_{22}\Gamma^2_{11} - \nu\Gamma^1_{22}\Gamma^2_{11}$$
$$+\Gamma^1_{12}\Gamma^2_{12} + \nu\Gamma^1_{12}\Gamma^2_{12} + 2\Gamma^2_{12}\Gamma^2_{22})a^{11} + (-\partial\Gamma^2_{222} + 2(\Gamma^1_{22}\Gamma^2_{12} + {\Gamma^2_{22}}^2))a^{12})a^{22}\Big)v_2\,. \qquad \text{(A.1)}$$

339

References

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

[2] Y. Bazilevs, L. Beirão da Veiga, J.A. Cottrell, T.J.R. Hughes, and G. Sangalli. Isogeometric analysis: approximation, stability and error estimates for h-refined meshes. *Mathematical Models and Methods in Applied Sciences*, 16:1–60, 2006.

[3] L. Beirão da Veiga, A. Buffa, J. Rivas, and G. Sangalli. Some estimates for $h-p-k-$refinement in isogeometric analysis. *Numerische Mathematik*, 118:271–305, 2011.

[4] J.A. Cottrell, A. Reali, Y. Bazilevs, and T.J.R. Hughes. Isogeometric analysis of structural vibrations. *Computer Methods in Applied Mechanics and Engineering*, 195:5257–5296, 2006.

[5] S. Morganti, F. Auricchio, D.J. Benson, F.I. Gambarin, S. Hartmann, T.J.R. Hughes, and A. Reali. Patient-specific isogeometric structural analysis of aortic valve closure. *ICES Report*, 14-10, 2014.

[6] J. Kiendl, K.-U. Bletzinger, J. Linhard, and R. Wüchner. Isogeometric shell analysis with Kirchhoff-Love elements. *Computer Methods in Applied Mechanics and Engineering*, 198:3902–3914, 2009.

[7] J. Kiendl, Y. Bazilevs, M.-C. Hsu, R. Wüchner, and K.-U. Bletzinger. The bending strip method for isogeometric analysis of Kirchhoff-Love shell structures comprised of multiple patches. *Computer Methods in Applied Mechanics and Engineering*, 199:2403–2416, 2010.

[8] N. Nguyen-Thanh, J. Kiendl, H. Nguyen-Xuan, R. Wüchner, K. U. Bletzinger, Y. Bazilevs, and T. Rabczuk. Rotation free isogeometric thin shell analysis using PHT-splines. *Computer Methods in Applied Mechanics and Engineering*, 200(47-48):3410–3424, 2011.

[9] J. Kiendl, M.-C. Hsu, M.C.H. Wu, and A. Reali. Isogeometric Kirchhoff-Love shell formulations for general hyperelastic materials. *Computer Methods in Applied Mechanics and Engineering*, 291:280–303, 2015.

[10] A.B. Tepole, H. Kabaria, K.-U. Bletzinger, and E. Kuhl. Isogeometric Kirchhoff-Love shell formulations for biological materials. *Computer Methods in Applied Mechanics and Engineering*, 293:328–347, 2015.

[11] T.D. Duong, F. Roohbakhshan, and R.A. Sauer. A new rotation-free isogeometric thin shell formulation and a corresponding continuity constraint for patch boundaries. *Computer Methods in Applied Mechanics and Engineering*, 2016.

[12] F. Cirak, M. Ortiz, and P. Schröder. Subdivision surfaces: a new paradigm for thin shell analysis. *International Journal for Numerical Methods in Engineering*, 47:2039–2072, 2000.

[13] F. Cirak and M. Ortiz. Fully C1-conforming subdivision elements for finite deformation thin-shell analysis. *International Journal for Numerical Methods in Engineering*, 51:813–833, 2001.

[14] F. Cirak, M. J. Scott, E. K. Antonsson, M. Ortiz, and P. Schröder. Integrated modeling, finite-element analysis, and engineering design for thin-shell structures using subdivision. *Computer-Aided Design*, 34:137–148, 2002.

[15] R. Echter, B. Oesterle, and M. Bischoff. A hierarchic family of isogeometric shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 254(0):170 – 180, 2013.

[16] B. Oesterle, E. Ramm, and M. Bischoff. A shear deformable, rotation-free isogeometric shell formulation. *Computer Methods in Applied Mechanics and Engineering*, 307:235–255, 2016.

[17] T.-K. Uhm and S.-K. Youn. T-spline finite element method for the analysis of shell structures. *International Journal for Numerical Methods in Engineering*, 80:507–536, 2009.

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

[19] W. Dornisch, S. Klinkel, and B. Simeon. Isogeometric Reissner-Mindlin shell analysis with exactly calculated director vectors. *Computer Methods in Applied Mechanics and Engineering*, 253:491–504, 2013.

[20] W. Dornisch and S. Klinkel. Treatment of Reissner-Mindlin shells with kinks without the need for drilling rotation stabilization in an isogeometric framework. *Computer Methods in Applied Mechanics and Engineering*, 276:35–66, 2014.

[21] S. Hosseini, J.J.C. Remmers, C.V. Verhoosel, and R. de Borst. An isogeometric solid-like shell element for nonlinear analysis. *International Journal for Numerical Methods in Engineering*, 95:238–256, 2013.

[22] S. Hosseini, J.J.C. Remmers, C.V. Verhoosel, and R. de Borst. An isogeometric continuum shell element for non-linear analysis. *Computer Methods in Applied Mechanics and Engineering*, 271:1–22, 2014.

[23] R. Bouclier, T. Elguedj, and A. Combescure. Efficient isogeometric NURBS-based solid-shell elements: Mixed formulation and B-bar-method. *Computer Methods in Applied Mechanics and Engineering*, 267:86–110, December 2013.

[24] J.F. Caseiro, R.A.F. Valente, A. Reali, J. Kiendl, F. Auricchio, and R.J Alves de Sousa. On the Assumed Natural Strain method to alleviate locking in solid-shell NURBS-based finite elements. *Computational Mechanics*, 53:1341–1353, 2014.

[25] J.F. Caseiro, R.A.F. Valente, A. Reali, J. Kiendl, F. Auricchio, and R.J Alves de Sousa. Assumed Natural Strain NURBS-based solid-shell element for the analysis of large deformation elasto-plastic thin-shell structures. *Computer Methods in Applied Mechanics and Engineering*, 284:861–880, 2015.

[26] F. Auricchio, F. Calabrò, T.J.R. Hughes, A. Reali, and G. Sangalli. A simple algorithm

for obtaining nearly optimal quadrature rules for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 249–252:15–27, 2012.

[27] T.J.R. Hughes, A. Reali, and G. Sangalli. Efficient quadrature for NURBS-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 199:301–313, 2010.

[28] D. Schillinger, S.J. Hossain, and T.J.R. Hughes. Reduced Bézier element quadrature rules for quadratic and cubic splines in isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 277:1–45, 2014.

[29] C. Adam, T.J.R. Hughes, S. Bouabdallah, M. Zarroug, and H. Maitournam. Selective and reduced numerical integrations for nurbs-based isogeometric analysis. *Computer Methods in Applied Mechanics and Engineering*, 284:732–761, 2015.

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

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

[32] H. Lin, Q. Hu, and Y. Xiong. Consistency and convergence properties of the isogeometric collocation method. *Computer Methods in Applied Mechanics and Engineering*, 267:471–486, 2013.

[33] F. Auricchio, L. Beirão da Veiga, T.J.R. Hughes, A. Reali, and G. Sangalli. Isogeometric collocation for elastostatics and explicit dynamics. *Computer Methods in Applied Mechanics and Engineering*, 249-252:2–14, 2012.

[34] L. Beirão da Veiga, C. Lovadina, and A. Reali. Avoiding shear locking for the Timoshenko beam problem via isogeometric collocation methods. *Computer Methods in Applied Mechanics and Engineering*, 241-244:38–51, 2012.

[35] J. Kiendl, F. Auricchio, T.J.R. Hughes, and A. Reali. Single-variable formulations and isogeometric discretizations for shear deformable beams. *Computer Methods in Applied Mechanics and Engineering*, 284:988–1004, 2015.

[36] F. Auricchio, L. Beirão da Veiga, J. Kiendl, C. Lovadina, and A. Reali. Locking-free isogeometric collocation methods for spatial Timoshenko rods. *Computer Methods in Applied Mechanics and Engineering*, 263:113–126, 2013.

[37] E. Marino. Isogeometric collocation for three-dimensional geometrically exact shear-deformable beams. *Computer Methods in Applied Mechanics and Engineering*, 307:383–410, 2016.

[38] O. Weeger, S-K. Yeung, and M. L. Dunn. Isogeometric collocation methods for Cosserat rods and rod structures. *Computer Methods in Applied Mechanics and Engineering*, 316:100–122, 2017.

[39] E. Marino. Locking-free isogeometric collocation formulation for three-dimensional geometrically exact shear-deformable beams with arbitrary initial curvature. *Computer Methods in Applied Mechanics and Engineering*, http://dx.doi.org/10.1016/j.cma.2017.06.031, 2017.

[40] J. Kiendl, F. Auricchio, L. Beirão da Veiga, C. Lovadina, and A. Reali. Isogeometric collocation methods for the Reissner-Mindlin plate problem. *Computer Methods in Applied Mechanics and Engineering*, 284:489–507, 2015.

[41] A. Reali and H. Gomez. An isogeometric collocation approach for Bernoulli-Euler beams and Kirchhoff plates. *Computer Methods in Applied Mechanics and Engineering*, 284:623–636, 2015.

[42] R. Kruse, N. Nguyen-Thanh, L. De Lorenzis, and T.J.R. Hughes. Isogeometric collocation for large deformation elasticity and frictional contact problems. *Computer Methods in Applied Mechanics and Engineering*, 296:73–112, 2015.

[43] L. De Lorenzis, J.A. Evans, T.J.R. Hughes, and A. Reali. Isogeometric collocation:

Neumann boundary conditions and contact. *Computer Methods in Applied Mechanics and Engineering*, 284:21–54, 2015.

[44] H. Gomez, A. Reali, and G. Sangalli. Accurate, efficient, and (iso)geometrically flexible collocation methods for phase-field models. *Journal for Computational Physics*, 262:153–171, 2014.

[45] D. Schillinger, M.J. Borden, and H. Stolarski. Isogeometric collocation for phase-field fracture models. *Computer Methods in Applied Mechanics and Engineering*, 284:583–610, 2015.

[46] D. Chapelle and K.-J. Bathe. Fundamental considerations for the finite element analysis of shell structures. *Computers & Structures*, 66(1):19–36, 1998.

[47] M. Bischoff, W.A. Wall, K.-U. Bletzinger, and E. Ramm. Models and finite elements for thin-walled structures. In *Encyclopedia of Computational Mechanics*, volume 2, Solids, Structures and Coupled Problems. Wiley, 2004.

[48] Y. Basar and W. Krätzig. *Mechanik der Flächentragwerke*. Vieweg, Braunschweig, 1985.

[49] Y. Basar and W. Krätzig. *Theory of Shell Structures*. VDI Verlag, 2001.

[50] Wolfram. Mathematica, 2017.

[51] S. A. Klioner. EinS: a Mathematica package for tensorial calculations in astronomical applications of relativistic gravity theories. In M. Francaviglia, editor, *14th international conference on general relativity and gravitation*, page p. A.182, Turin, 1995.

[52] E. Cohen, R.F. Riesenfeld, and G. Elber. *Geometric Modeling with Splines: An Introduction*. A K Peters, Natick, MA, 2001.

[53] L. Piegl and W. Tiller. *The NURBS Book*. Springer-Verlag, New York, 2nd edition, 1997.

[54] D.F. Rogers. *An Introduction to NURBS With Historical Perspective*. Academic Press, San Diego, CA, 2001.

[55] J.A. Cottrell, T.J.R. Hughes, and Y. Bazilevs. *Isogeometric Analysis: Toward Integration of CAD and FEA*. Wiley, 2009.

[56] A. Reali and T.J.R. Hughes. An introduction to isogeometric collocation methods. In G. Beer, editor, *Isogeometric Methods for Numerical Simulation*. Springer, 2015.

[57] H. Gomez and L. De Lorenzis. The variational collocation method. *Computer Methods in Applied Mechanics and Engineering*, 309:152–181, 2016.

[58] C. Anitescu, Y. Jia, J. Zhang, and T. Rabczuk. An isogeometric collocation method using superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 284:1073–1097, 2015.

[59] M. Montardini, G. Sangalli, and L. Tamellini. Optimal-order isogeometric collocation at galerkin superconvergent points. *Computer Methods in Applied Mechanics and Engineering*, 316:741–757, 2017.

[60] T. Belytschko, H. Stolarski, W.K. Liu, N. Carpenter, and J.S.-J. Ong. Stress projection for membrane and shear locking in shell finite elements. *Computer Methods in Applied Mechanics and Engineering*, 51:221–258, 1985.

[61] K.-J. Bathe, A. Iosilevich, and D. Chapelle. An evaluation of the MITC shell elements. *Computers & Structures*, 75:1–30, 2000.

[62] J. Pitkäranta, Y. Leino, O. Ovaskainen, and J. Piila. Shell deformation states and the finite element method: A benchmark study of cylindrical shells. *Computer Methods in Applied Mechanics and Engineering*, 128:81–121, 1995.