



Code verification examples based on the method of manufactured solutions for Kirchhoff–Love and Reissner–Mindlin shell analysis

M. H. Gfrerer¹ · M. Schanz¹

Received: 2 August 2017 / Accepted: 5 December 2017 / Published online: 16 December 2017
© The Author(s) 2017. This article is an open access publication

Abstract

We propose code verification examples based on the method of manufactured solutions (MMS) for Kirchhoff–Love and Reissner–Mindlin shell analysis. In order to ensure the credibility of numerical simulations reliable software has to be provided. Code verification is the process of ensuring that there are no coding mistakes in the implementation and that the algorithms work properly. The MMS is an elegant method to derive exact solutions to boundary value problems which can be used for rigorous order of accuracy tests within the code verification process. We apply the proposed tests to an in-house developed finite element research code implementing elements for Kirchhoff–Love and Reissner–Mindlin shell analysis.

Keywords Code verification · Method of manufactured solutions · Shell analysis · Kirchhoff–Love · Reissner–Mindlin

1 Introduction

In order to ensure the reliability of complex computational models, verification and validation (V&V) are unavoidable tasks [1, 2]. In the present paper, we follow the concepts on V&V of The American Society of Mechanical Engineers guideline [3]. The process of validation determines how accurate the model represents the real situation. The aim of verification is to show that the simulation results are able to represent accurately the solution of the underlying mathematical model. Thus, one is interested in the numerical error, which is the difference between the simulation results and the exact solution of the mathematical model. The comparison of simulation results with experimental data is only feasible if the results are not significantly influenced by the numerical error. Therefore, the process of verification has to precede the validation process. Verification activities can be split into code verification and calculation verification. Code verification represents the process of demonstrating that the governing equations, as implemented in the code, are solved consistently. Calculation verification is the assessment (estimation) of the numerical error in situations where no exact solution is known [4].

According to [5], the most rigorous tests for code verification are the order-of-accuracy tests. For any discretization method, we expect that the discretization error decreases as the mesh is refined. Within an order-of-accuracy test, the observed rate of decrease in the discretization error is compared with the theoretical rate. In order to evaluate the discretization error exact solutions are needed. These exact solutions can be constructed by the method of manufactured solutions (MMS) [1, 5–8]. Its central idea is to prescribe a solution and to determine an artificial source term which is added to the governing equations, such that the modified equations are fulfilled for the prescribed solution. The MMS can be applied to a wide range of problems. It has been applied to Reynolds-averaged Navier–Stokes solvers [9], nonlinear membrane elements [10], within fluid structure interaction [11, 12], conjugate heat transfer solvers [13], Cahn–Hilliard equation [14], and others.

Within shell analysis, the task is to predict the mechanical response of a thin curved structure. A structure is characterized as thin if one space dimension has a much smaller extension than the other two. Within this geometric setting, a direct solution of the three-dimensional problem is difficult. Therefore, many different shell models exist (see [15] for an overview). In a number of recent papers [16–21] advanced numerical approaches for shell analysis have been developed. In these contributions, order-of-accuracy tests were presented only for special cases. For more general cases, benchmark examples from the shell obstacle course [22] were used. This

✉ M. Schanz
m.schanz@tugraz.at

¹ Institute of Applied Mechanics, Graz University of Technology, Technikerstrasse 4, 8010 Graz, Austria

set of problems consists of the Scordelis-Lo roof problem, the pinched cylinder with a diaphragm, and the hemispherical shell problem. This problem set was introduced in order to compare the performance of different finite elements with respect to locking and accurate representation of rigid body motions. Using this set for code verification one shortcoming is the lack of generality in geometry, since only cylindrical and spherical geometries are taken into consideration. Another drawback is that there are only reference displacement values available, which are obtained by numerical simulation. Therefore, no order-of-accuracy tests are possible.

In the present paper, we investigate code verification of finite element codes for shell analysis based on the MMS. Thus, an exact solution is available and an order of accuracy test can be applied. We propose a series of verification examples with increasing complexity. We apply each test to a research code, which implements high order finite elements for Kirchhoff–Love and Reissner–Mindlin shells.

2 Shell models and finite element method

In this section, the Kirchhoff–Love and the Reissner–Mindlin shell theory is recalled and the finite element formulation is given briefly.

2.1 The 3D shell problem

We start with the statement of the 3D shell problem, which is a 3D problem of linearized elasticity on special domains Ω .

$$\text{Find } \tilde{\mathbf{u}} : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}^3:$$

$$\begin{aligned} \nabla \cdot (\tilde{\boldsymbol{\sigma}}) + \tilde{\mathbf{b}} &= 0 && \text{in } \Omega, \\ \tilde{\boldsymbol{\sigma}} &= \tilde{\mathbb{C}} : \tilde{\boldsymbol{\varepsilon}} && \text{in } \Omega, \\ \tilde{\boldsymbol{\varepsilon}} &= \frac{1}{2}(\text{grad}(\tilde{\mathbf{u}}) + \text{grad}(\tilde{\mathbf{u}})^T) && \text{in } \Omega, \\ \tilde{\mathbf{u}} &= \tilde{\mathbf{u}}_D && \text{on } \Gamma_D, \\ \tilde{\mathbf{t}} &= \tilde{\mathbf{t}}_N && \text{on } \Gamma_N. \end{aligned} \tag{1}$$

We assume that the domain Ω is defined through a parametrization $\bar{\mathbf{g}}$ of a reference surface $\bar{\Omega}$ (see Fig. 1).

$$\begin{aligned} \bar{\mathbf{g}} : \bar{U} \subset \mathbb{R}^2 &\rightarrow \bar{\Omega} \subset \mathbb{R}^3 \\ (\theta^1, \theta^2) &\mapsto \bar{\mathbf{g}}(\theta^1, \theta^2). \end{aligned} \tag{2}$$

In the present paper, we restrict us to the case of rectangular parameter domains $\bar{\Omega}$. Then the parametrization of Ω is given by

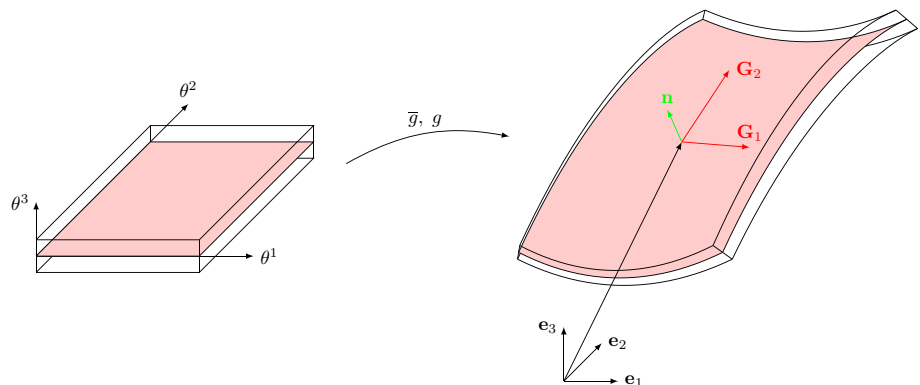
$$\begin{aligned} \mathbf{g} : (\bar{U} \times T) \subset \mathbb{R}^3 &\rightarrow \Omega \subset \mathbb{R}^3 \\ (\theta^1, \theta^2) \times \theta^3 &\mapsto \mathbf{g}(\theta^1, \theta^2, \theta^3) = \bar{\mathbf{g}}(\theta^1, \theta^2) + \theta^3 \mathbf{n}, \end{aligned} \tag{3}$$

with the unit normal vector \mathbf{n} and the thickness interval $T = \left[-\frac{t}{2}, \frac{t}{2}\right]$, where t is the thickness of the shell. In the rest of the paper, Latin indices i, j, \dots take the values 1, 2, 3 whereas Greek indices α, β take the values 1, 2. Furthermore, Einstein summation convention applies. In the following, we need the covariant base vectors $\bar{\mathbf{G}}_\alpha = \frac{\partial \bar{\mathbf{g}}}{\partial \theta^\alpha}$, $\mathbf{G}_\alpha = \frac{\partial \mathbf{g}}{\partial \theta^\alpha}$, the covariant coefficients of the metric $\bar{G}_{\alpha\beta} = \bar{\mathbf{G}}_\alpha \cdot \bar{\mathbf{G}}_\beta$, $G_{ij} = \mathbf{G}_i \cdot \mathbf{G}_j$, the contravariant coefficients of the metric $\bar{G}^{\alpha\beta} = [\bar{G}_{\alpha\beta}]^{-1}$, $G^{\alpha\beta} = [G_{\alpha\beta}]^{-1}$, and the contravariant base vectors $\bar{\mathbf{G}}^\alpha = \bar{G}^{\alpha\beta} \bar{\mathbf{G}}_\beta$, $\mathbf{G}^i = G^{ij} \mathbf{G}_j$. Note that all quantities with a bar are defined on the reference surface and are, therefore, independent of θ^3 . Whereas quantities with no bar are defined on the three-dimensional shell volume. Instead of solving for $\tilde{\mathbf{u}}(\mathbf{x})$, $\mathbf{x} \in \Omega$, we seek the displacement field $\mathbf{u}(\theta^1, \theta^2, \theta^3) = \tilde{\mathbf{u}}(\mathbf{g}(\theta^1, \theta^2, \theta^3))$ defined on the parametric space. We assume that the bodyforce \mathbf{b} is given with respect to a fixed Cartesian frame \mathbf{e}_i . Thus, $\mathbf{b} = b^i \mathbf{e}_i$ holds. Then, the balance of momentum in (1) in parametric coordinates reads

$$\left(\sigma_j^i + \sigma^{kl} \Gamma_{kj}^i + \sigma^{jk} \Gamma_{kl}^i\right) J_l^i + b^i = 0, \tag{4}$$

with the contravariant components of the stress tensor $\boldsymbol{\sigma} = \sigma^{ij} \mathbf{G}_i \otimes \mathbf{G}_j$, the Christoffel symbols of second kind $\Gamma_{ij}^k = \mathbf{G}_{,i}^k \cdot \mathbf{G}_j$, and $J_l^i = \mathbf{G}_l \cdot \mathbf{e}^i$. Here and in the following

Fig. 1 Parametrization of the shell. The parameter space on the left is mapped to the physical space on the right. The reference surface (red) is parametrized by $\bar{\mathbf{g}}$, whereas the shell volume is parametrized by \mathbf{g} .



the notation $(\cdot)_{,j} = \frac{\partial(\cdot)}{\partial\theta^j}$ applies. In the present paper, we employ a linear isotropic material law, where the contravariant components of the elasticity tensor $\mathbb{C} = \mathbb{C}^{ijkl} \mathbf{G}_i \otimes \mathbf{G}_j \otimes \mathbf{G}_k \otimes \mathbf{G}_l$ are given by

$$\begin{aligned} \mathbb{C}^{\alpha\beta\gamma\varphi} &= \lambda \mathbf{G}^{\alpha\beta} \mathbf{G}^{\gamma\varphi} + \mu (\mathbf{G}^{\alpha\gamma} \mathbf{G}^{\beta\varphi} + \mathbf{G}^{\alpha\varphi} \mathbf{G}^{\beta\gamma}), \\ \mathbb{C}^{\alpha\beta 33} &= \mathbb{C}^{33\alpha\beta} = \lambda \mathbf{G}^{\alpha\beta}, \\ \mathbb{C}^{3\alpha 3\beta} &= \mathbb{C}^{\alpha 3\beta 3} = \mathbb{C}^{\alpha 3\beta 3} = \mu \mathbf{G}^{\alpha\beta}, \\ \mathbb{C}^{3\alpha\beta\gamma} &= \mathbb{C}^{\alpha\beta 3\gamma} = \mathbb{C}^{\alpha\beta 3\gamma} = \mathbb{C}^{\alpha\beta 3\gamma} = 0, \\ \mathbb{C}^{333\alpha} &= \mathbb{C}^{33\alpha 3} = \mathbb{C}^{3\alpha 33} = \mathbb{C}^{\alpha 333} = 0, \\ \mathbb{C}^{3333} &= \lambda + 2\mu, \end{aligned} \tag{5}$$

with λ and μ denoting the Lamé constants. For a given displacement field \mathbf{u} the covariant components of the linearized strain $\boldsymbol{\epsilon} = \epsilon_{ij} \mathbf{G}^i \otimes \mathbf{G}^j$ are

$$\begin{aligned} \epsilon_{\alpha\beta} &= \frac{1}{2} \left(\mu_\alpha^\epsilon \bar{\mathbf{G}}_\epsilon \cdot \frac{\partial \mathbf{u}(\theta^i)}{\partial \theta^\beta} + \mu_\beta^\epsilon \bar{\mathbf{G}}_\epsilon \cdot \frac{\partial \mathbf{u}(\theta^i)}{\partial \theta^\alpha} \right), \\ \epsilon_{\alpha 3} &= \frac{1}{2} \left(\mu_\alpha^\epsilon \bar{\mathbf{G}}_\epsilon \cdot \frac{\partial \mathbf{u}(\theta^i)}{\partial \theta^3} + \mathbf{n} \cdot \frac{\partial \mathbf{u}(\theta^i)}{\partial \theta^\alpha} \right), \\ \epsilon_{33} &= \frac{\partial \mathbf{u}(\theta^i)}{\partial \theta^3} \cdot \mathbf{n}, \end{aligned} \tag{6}$$

with the components of the shifter tensor $\mu_\alpha^\beta = (\delta_\alpha^\beta - \theta^3 h_\alpha^\beta)$, where $h_\alpha^\beta = G^{\beta\gamma} \mathbf{G}_{\alpha,\gamma} \cdot \mathbf{n}$.

One common assumption in shell theories is that straight fibers normal to the mid-surface remain straight after deformation. Therefore, we assume a displacement field of the form

$$\bar{\mathbf{u}}(\theta^1, \theta^2, \theta^3) = u^i(\theta^1, \theta^2) \mathbf{e}_i + \theta^3 v^\alpha(\theta^1, \theta^2) \bar{\mathbf{G}}_\alpha. \tag{7}$$

2.2 Kirchhoff–Love model

In the Kirchhoff–Love model $v^\alpha(\theta^1, \theta^2)$ are no independent parameters. The assumption that normals to the undeformed surface remain normal to the deformed surface and remain unstretched leads to

$$v^\alpha = -\bar{G}^{\alpha\gamma} \mathbf{u}_{,\gamma} \cdot \mathbf{n} \tag{8}$$

and

$$\bar{\mathbf{u}}(\theta^1, \theta^2, \theta^3) = u^i(\theta^1, \theta^2) \mathbf{e}_i - \theta^3 \bar{G}^{\alpha\gamma} (\mathbf{u}_{,\gamma} \cdot \mathbf{n}) \bar{\mathbf{G}}_\alpha. \tag{9}$$

Therefore, the components of the consistent strain tensor (6) can be computed

$$\begin{aligned} \epsilon_{11} &= \mu_1^\epsilon u_{,1}^i \bar{J}_{\epsilon i} - \theta^3 \mu_1^\epsilon \left(u_{,\epsilon 1}^i \bar{J}_{3i} - u_{,\epsilon}^i \bar{J}_{\beta i} h_1^\beta - u_{,\gamma}^i \bar{J}_{3i} \bar{\Gamma}_{1\epsilon}^\gamma \right), \\ 2\epsilon_{12} &= \mu_2^\epsilon u_{,1}^i \bar{J}_{\epsilon i} + \mu_1^\epsilon u_{,2}^i \bar{J}_{\epsilon i} - \theta^3 \mu_2^\epsilon \left(u_{,\epsilon 1}^i \bar{J}_{3i} - u_{,\epsilon}^i \bar{J}_{\beta i} h_1^\beta - u_{,\gamma}^i \bar{J}_{3i} \bar{\Gamma}_{1\epsilon}^\gamma \right) \\ &\quad - \theta^3 \mu_1^\epsilon \left(u_{,\epsilon 2}^i \bar{J}_{3i} - u_{,\epsilon}^i \bar{J}_{\beta i} h_2^\beta - u_{,\gamma}^i \bar{J}_{3i} \bar{\Gamma}_{2\epsilon}^\gamma \right), \\ \epsilon_{22} &= \mu_2^\epsilon u_{,2}^i \bar{J}_{\epsilon i} - \theta^3 \mu_2^\epsilon \left(u_{,\epsilon 2}^i \bar{J}_{3i} - u_{,\epsilon}^i \bar{J}_{\beta i} h_2^\beta - u_{,\gamma}^i \bar{J}_{3i} \bar{\Gamma}_{2\epsilon}^\gamma \right), \\ \epsilon_{\alpha 3} &= 0, \\ \epsilon_{33} &= 0. \end{aligned} \tag{10}$$

In order to avoid Poisson thickness locking the zero transverse normal stress assumption ($\sigma_{33} = 0$) has to be included. Enforcing the condition leads to the expression

$$\epsilon_{33} = \frac{\lambda}{\lambda + 2\mu} (G^{11} \epsilon_{11} + 2G^{12} \epsilon_{12} + G^{22} \epsilon_{22}) \tag{11}$$

for the modified transverse normal strain. Within the finite element code this results in a modification in the material law, i.e. in all equations λ has to be replaced with $\frac{2\mu\lambda}{2\mu+\lambda}$.

2.3 Reissner–Mindlin model

In the Reissner–Mindlin model $v^\alpha(\theta^1, \theta^2)$ are two additional independent unknown fields. The components of the strain tensor (6) are now

$$\begin{aligned} \epsilon_{11} &= \mu_1^\epsilon \bar{J}_{\epsilon i} u_{,1}^i + \theta^3 \mu_1^\epsilon \left(v_{,1}^\gamma \bar{G}_{\epsilon\gamma} + v^\gamma \bar{\Gamma}_{\gamma 1\epsilon} \right), \\ 2\epsilon_{12} &= \mu_1^\epsilon \bar{J}_{\epsilon i} u_{,2}^i + \mu_2^\epsilon \bar{J}_{\epsilon i} u_{,1}^i + \theta^3 \mu_1^\epsilon \left(v_{,2}^\gamma \bar{G}_{\epsilon\gamma} + v^\gamma \bar{\Gamma}_{\gamma 2\epsilon} \right) \\ &\quad + \theta^3 \mu_2^\epsilon \left(v_{,1}^\gamma \bar{G}_{\epsilon\gamma} + v^\gamma \bar{\Gamma}_{\gamma 1\epsilon} \right), \\ \epsilon_{22} &= \mu_2^\epsilon \bar{J}_{\epsilon i} u_{,2}^i + \theta^3 \mu_2^\epsilon \left(v_{,2}^\gamma \bar{G}_{\epsilon\gamma} + v^\gamma \bar{\Gamma}_{\gamma 2\epsilon} \right), \\ 2\epsilon_{\alpha 3} &= \mu_\alpha^\epsilon \bar{G}_{\epsilon\gamma} v_{,\gamma}^\alpha + u_{,\alpha}^i J_{3i} + v^\gamma h_{\gamma\alpha}, \\ \epsilon_{33} &= 0. \end{aligned} \tag{12}$$

Again the zero transverse normal stress condition has to be enforced in order to avoid Poisson thickness locking.

2.4 Finite element method

In this section, we describe the used finite element approach briefly. We have the following variational formulation of (1):

Find $\bar{\mathbf{u}} \in V$ such that

$$\int_\Omega \bar{\boldsymbol{\epsilon}}(\bar{\mathbf{v}}) : \tilde{\mathbb{C}} : \bar{\boldsymbol{\epsilon}}(\bar{\mathbf{u}}) \, dx = \int_\Omega \bar{\mathbf{v}} \cdot \tilde{\mathbf{b}} \, dx + \int_{\Gamma_N} \bar{\mathbf{v}} \cdot \tilde{\mathbf{t}} \, ds_x \quad \forall \bar{\mathbf{v}} \in V_0. \tag{13}$$

Here, $V = \{\bar{\mathbf{u}} \in [H^1(\Omega)]^3 \mid \bar{\mathbf{u}} = \bar{\mathbf{u}} \text{ on } \Gamma_D\}$ and $V_0 = \{\bar{\mathbf{u}} \in [H^1(\Omega)]^3 \mid \bar{\mathbf{u}} = 0 \text{ on } \Gamma_D\}$. We consider the change of variables according to $\mathbf{u}(\theta^1, \theta^2, \theta^3) = \bar{\mathbf{u}}(\mathbf{g}(\theta^1, \theta^2, \theta^3))$ for all quantities in (13). Thus, the integrals in (13) are evaluated

on the parametric domain utilizing a quadrature rule. In particular, we use a tensor product quadrature rule composed of one-dimensional Gauss–Legendre quadratures for the volume integrals. Therefore, we have for a generic integrand $\mathcal{A}(\theta^1, \theta^2, \theta^3) = \tilde{\mathcal{A}}(x)$,

$$\int_{\Omega} \tilde{\mathcal{A}}(x) \, dx = \int_{\tilde{U} \times T} \mathcal{A}(\theta^1, \theta^2, \theta^3) (1 - \theta^3 2H + (\theta^3)^2 K) \sqrt{\det \tilde{G}_{\alpha\beta}} \, d\theta^1 \, d\theta^2 \, d\theta^3 \tag{14}$$

$$\approx \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \sum_{k=1}^{n_3} \mathcal{A}(\theta_i^1, \theta_j^2, \theta_k^3) (1 - \theta_k^3 2H + (\theta_k^3)^2 K) \sqrt{\det \tilde{G}_{\alpha\beta}} \, \omega_i \omega_j \omega_k,$$

where $H = \frac{1}{2}(h_a^\alpha)$ is the mean curvature and $K = \det(h_a^\beta)$ is the Gaussian curvature. For $a = 1, 2, 3$, n_a is the number of quadrature points θ_i^a and quadrature weights ω_i .

In order to discretize the sought field \mathbf{u} we apply two steps. First, we resolve the through-the-thickness variation (the dependency on θ^3) of \mathbf{u} either by the Kirchhoff–Love kinematics (9) or the Reissner–Mindlin kinematics (7). The model decision has great impact on the subsequent discretization done in a second step. In the case of the Reissner–Mindlin shell only first order derivatives appear in (12). Therefore, we can use H^1 hierarchical shape functions on quadrilaterals [23] for the discretization of the five components u^i and v^α . Due to the second derivatives in (10) this approach is not feasible for the Kirchhoff–Love model. In order to have a H^2 conforming method we use the Bognor–Fox–Schmidt quadrilateral [24] to discretize the three components u^i . Dirichlet boundary conditions are incorporated in the ansatz space in the case of the Reissner–Mindlin model or enforced by a penalty technique in the case of the Kirchhoff–Love model (cf. [25]).

3 Code verification

In the present paper, we apply code verification based on order-of-accuracy tests and the MMS to a code for solving the shell formulations given in Sect. 2. The necessary prerequisite to apply it to a numerical schema is the knowledge of a formal order of convergence and exact solutions. Thus, an estimate of the type

$$\|u^{\text{exact}} - u^{\text{numerical}}\| \leq C h^q \|u^{\text{exact}}\|, \tag{15}$$

where C is a constant and h is a characteristic element size, has to be known. Then q is called the formal order of convergence with respect to the norm $\|\cdot\|$. For two meshes with characteristic element sizes h_1 and h_2 , the experimental order of convergence (eoc) is defined as

$$\text{eoc} = \frac{\log(e_1) - \log(e_2)}{\log(h_1) - \log(h_2)}, \tag{16}$$

where

$$e_i = \|u^{\text{exact}} - u_{h_i}^{\text{numerical}}\| \tag{17}$$

is the numerical error corresponding to the discretization

h_i . The code is verified, if the eoc matches the formal order of convergence within the asymptotic range. For the finite element method applied in this paper, we expect $q = p + 1$ for the error in the L_2 norm for smooth solutions for the Reissner–Mindlin shell. In the case of the Kirchhoff–Love shell no theoretical error estimates are known to the authors.

In order to evaluate (17), an exact solution has to be available. Within the MMS, such a solution is prescribed. Then, the source term is determined such that the chosen solution fulfills the governing equations. In particular, we perform the following steps:

1. Choose the form of the problem domain, i.e. specify the surface parametrization (2) and the thickness t
2. Choose the form of the manufactured solution \mathbf{u}^M , i.e. specify the independent parameters in (7)
3. Derive the modified governing equations, i.e. compute an artificial source term $\mathbf{b}^M = -\nabla \cdot (\mathbb{C} : \epsilon(\mathbf{u}^M))$ and derive the boundary conditions
4. Solve the discrete form of the modified governing equations on multiple meshes, i.e. solve (13) with $\mathbf{b} = \mathbf{b}^M$
5. Evaluate the numerical error (17) and the eoc (16)
6. Apply the order of convergence test

Additionally to [26], we added the first point. The finite element code needs the components b^i of the source term $\mathbf{b} = b^i \mathbf{e}_i$ with respect to the global Cartesian frame. They are given by

$$b^i = -\left(\sigma_j^i + \sigma^{kl} \Gamma_{kj}^i + \sigma^{jk} \Gamma_{kj}^i\right) J_1^i. \tag{18}$$

As noted in [27], an arbitrary choice of manufactured solution leads easily to a very long computer code for the source term. We have observed this for general complex curved surfaces too. We use the computer algebra system Mathematica™ [28] to derive the source terms in an automated way. Nevertheless, a general verification example assessing all features of the code results in a source code with more than 10^6 characters, where simplifications are prohibitively time-consuming. Therefore, we compute only the derivatives analytically and the algebraic operations are performed

numerically. Instead of exporting a function for b^i , we export functions taking the spatial position as input argument for

- strain tensor ϵ_{ij} ,
- derivatives of the strain tensor $\epsilon_{ij,k}$,
- covariant base vectors \mathbf{G}_i ,
- derivatives of the covariant base vectors $\mathbf{G}_{i,j}$.

We evaluate these quantities at the needed spatial position and compute

$$\begin{aligned}
 G_{ij} &= \mathbf{G}_i \cdot \mathbf{G}_j, \\
 [G^{ij}] &= [G_{ij}]^{-1}, \\
 J_l^i &= \mathbf{G}_l \cdot \mathbf{e}^i, \\
 \Gamma_{kj}^i &= \mathbf{G}_{k,j} \cdot \mathbf{G}_i, \\
 G_{,k}^{ij} &= -\Gamma_{kl}^j G^{li} - \Gamma_{kl}^i G^{lj}, \\
 \sigma^{jl} &= \lambda(G^{ij} G^{kl} \epsilon_{kl} + 2\mu G^{ik} G^{jl} \epsilon_{kl}), \\
 \sigma_j^{il} &= \lambda(G_j^{ij} G^{kl} \epsilon_{kl} + G_j^{ij} G_j^{kl} \epsilon_{kl} + G_j^{ij} G^{kl} \epsilon_{kl,j}) \\
 &\quad + \mu(G_j^{ik} G_j^{jl} \epsilon_{kl} + G^{ik} G_j^{jl} \epsilon_{kl} + G^{ik} G^{jl} \epsilon_{kl,j}).
 \end{aligned}
 \tag{19}$$

Then the source term is obtained with (18).

Remark In the present implementation, we keep the exact surface parametrization during the whole computation. Therefore, we are able to evaluate the error in a continuous norm. In case of an approximation of the surface, one has to use the discrete norm [12]

$$e = \sqrt{\frac{1}{N} \sum_{n=1}^N \left(u^{\text{exact}}(x_n) - u^{\text{numerical}}(x_n) \right)^2}.
 \tag{20}$$

Furthermore, the influence of geometry approximation on the formal order of convergence has to be considered. For a discussion of this ‘variational crime’, we refer to [29]. We remark that in the present approach this is not an issue.

Table 1 Prescribed displacement parameters for the Reissner–Mindlin model

| | Field A | Field B |
|---------|--------------------|---------------------------------------|
| $u^1 =$ | θ^1 | $\sin(\pi\theta^1) \cos(\pi\theta^2)$ |
| $u^2 =$ | θ^2 | $\cos(\pi\theta^1) \sin(\pi\theta^2)$ |
| $u^3 =$ | $\theta^1\theta^2$ | $\sin(\pi\theta^1)\theta^2$ |
| $v^1 =$ | $\theta^1\theta^2$ | $\sin(\pi\theta^1)\theta^2$ |
| $v^2 =$ | $\theta^1\theta^2$ | $\sin(\pi\theta^1)\theta^2$ |

Table 2 Prescribed displacement parameters for the Kirchhoff–Love model, where $f(x, y) = \frac{64}{y^6}(y-x)^3x^3$

| | Field A | Field B | Field C |
|---------|--------------------|--------------------------------------|---------------------------------------|
| $u^1 =$ | θ^1 | $f(\theta^1, 0.56)f(\theta^2, 0.65)$ | $\sin(\pi\theta^1) \cos(\pi\theta^2)$ |
| $u^2 =$ | θ^2 | $f(\theta^1, 0.56)f(\theta^2, 0.65)$ | $\sin(\pi\theta^1) \cos(\pi\theta^2)$ |
| $u^3 =$ | $\theta^1\theta^2$ | $f(\theta^1, 0.56)f(\theta^2, 0.65)$ | $\sin(\pi\theta^1) \cos(\pi\theta^2)$ |

4 Verification examples

In the following, we provide a set of verification examples with increasing complexity. In principle, it would be sufficient to consider only the most general case in order to verify the code, since the special cases are included in the general case [30]. However, in order to have confirmation exercises, we suggest special cases where parts of the code can be tested.

In all examples, we use the parameter space $(\theta^1, \theta^2) \in [0, 0.56] \times [0, 0.65]$ and the thickness $t = 0.07$ m. The material parameters $\lambda = \mu = 4000$ N m⁻² are used, which corresponds to Young’s modulus $E = 1000$ N m⁻² and Poisson’s ratio of $\nu = 0.25$. An additional confirmation exercises is obtained through setting $\nu = 0$.

In the verification examples for the Reissner–Mindlin model, we use the two types of displacement fields given in Table 1. Displacement field A is chosen such that it can be represented exactly with each discretization. Displacement field B is chosen to be general to assess all features of the code. For the Kirchhoff–Love model we present three types of displacement fields, which are given in Table 2. Again field A can be exactly discretized and field C is chosen to be general. The field B is constructed such that it vanishes up to the second derivatives on the boundary. Therefore, one can test simple supported boundary conditions without the need of considering external moments along the boundary. Additional displacement fields for confirmation exercises can be obtained through setting only individual parameters. Different surface parametrizations are given in Table 3. The parametrization $\bar{\mathbf{q}}_1$ is the identity map yielding a plane rectangular mid-surface. With $\bar{\mathbf{q}}_2$ we have a plane mid-surface with curved boundaries. A cylindrical geometry and,

Table 3 Surface parametrizations (The coordinates in physical space (x_1, x_2, x_3) are given as functions of θ^1 and θ^2)

| | $\bar{\mathbf{q}}_1$ | $\bar{\mathbf{q}}_2$ | $\bar{\mathbf{q}}_3$ | $\bar{\mathbf{q}}_4$ |
|---------|----------------------|---------------------------|----------------------|---|
| $x_1 =$ | θ^1 | $\theta^1 + (\theta^2)^2$ | $\cos(\theta^1)$ | $\frac{2\theta^1\theta^2 + 2\theta^1 - \theta^2}{3}$ |
| $x_2 =$ | θ^2 | $\theta^2 + (\theta^1)^2$ | θ^2 | $\frac{-\theta^1\theta^2 + 2\theta^1 + 2\theta^2}{3}$ |
| $x_3 =$ | 0 | 0 | $\sin(\theta^1)$ | $\frac{2\theta^1\theta^2 - \theta^1 + 2\theta^2}{3}$ |

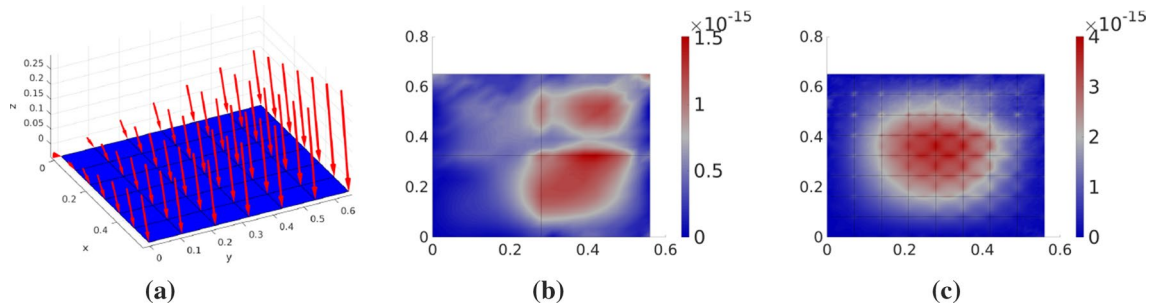


Fig. 2 Example 1: **a** reference surface and body force, **b** numerical error for a coarse mesh (double precision, $p = 2$), **c** numerical error for a fine mesh (double precision, $p = 2$)

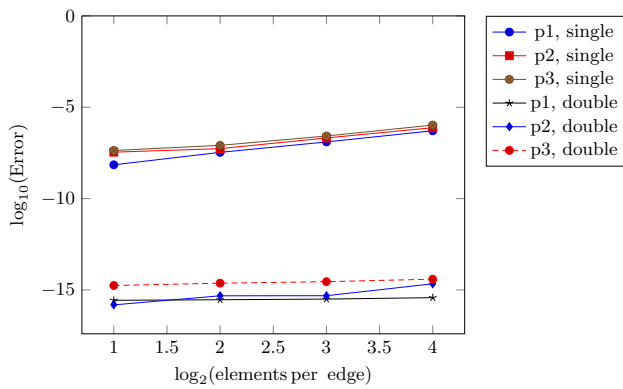


Fig. 3 Numerical error for example 1

therefore, curved (constant curvature) surface is given by \bar{q}_3 . The parametrization \bar{q}_4 defines a general curved surface.

In all examples, we consider all lateral sides as Dirichlet boundary and the top and bottom surfaces as Neumann boundary.

4.1 Results for the Reissner–Mindlin model

Example 1 Plane geometry, exact representation of solution.

The goal of the first example is to determine the influence of round-off errors. To this end, we have chosen the plane geometry given by the parametrization \bar{q}_1 from Table 3 and displacement field A from Table 1. In Fig. 2a the reference surface with the body force is depicted. The displacement field can be exactly represented in the discrete system. Therefore, the remaining error is due to round-off errors. We have executed the code for ansatz orders up to order three for single precision, as well as for double precision. The distribution of the error for a coarse and a fine mesh over the parameter domain is depicted in Fig. 2b, c, respectively. In Fig. 3, the errors for different discretization levels are shown. We observe constant error levels around 10^{-15} for double precision. This agrees with the expected accuracy. The errors for single precision arithmetics are as expected about 10^{-7} . However, using more elements raises the error. The conditioning of the discrete system might be the reason for this.

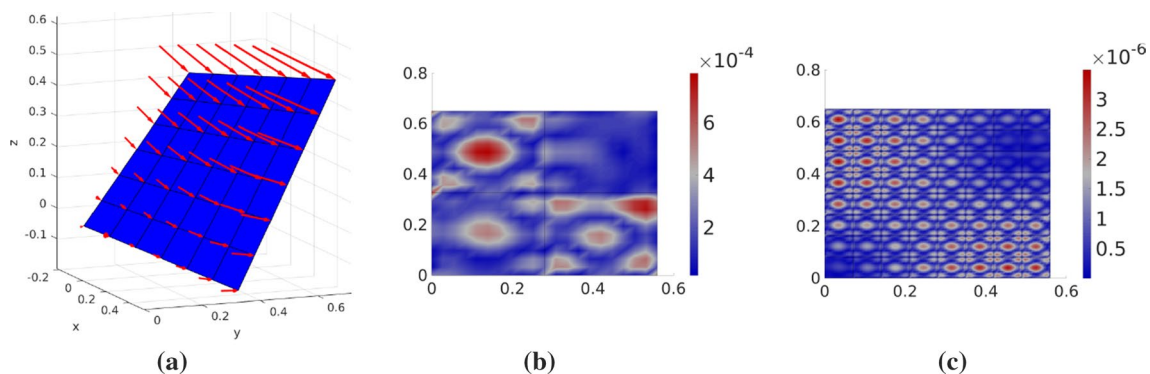


Fig. 4 Example 2: **a** reference surface and body force, **b** numerical error for a coarse mesh ($p = 3$), **c** numerical error for a fine mesh ($p = 3$)

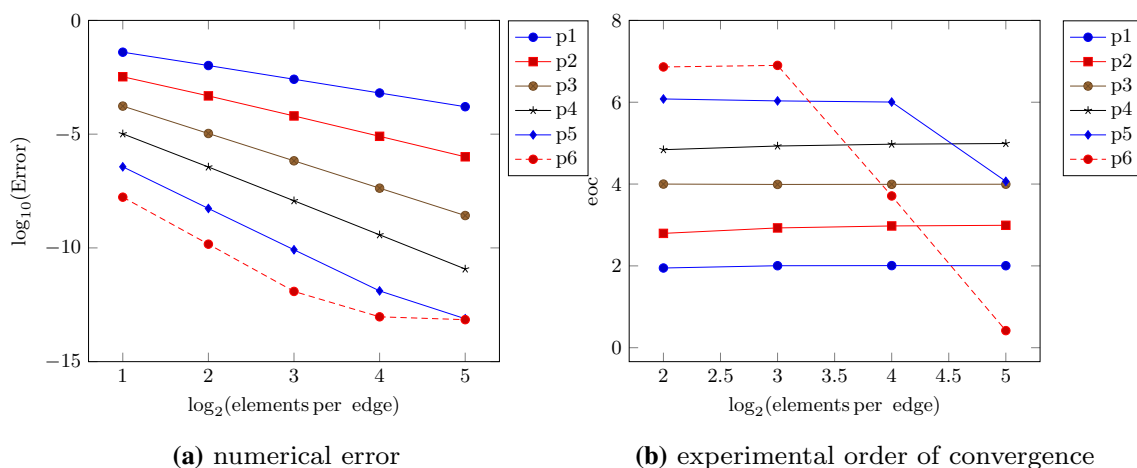


Fig. 5 Development of the numerical error and the eoc for Example 2 **a** numerical error **b** experimental order of convergence

Example 2 *Plane geometry, general solution.* In the second example, the discretization error at a plane geometry is assessed. Therefore, we use the parametrization \bar{g}_1 with displacement field B. The obtained distribution of the body force at the reference surface is shown in Fig. 4a. For ansatz order $p = 3$ the distributions of the errors is displayed in Fig. 4b, c. We observe oscillations in the errors. Such phenomena were also observed in [20] for a benchmark example in the case of elements showing locking. Since in the present element formulation no special method to alleviate locking has been applied, the present oscillations are expected. However, the locking does not deteriorate the convergence of the method. The development of the errors and the eoc for ansatz orders up to order six are shown in Fig. 5. For the ansatz orders one to four the eoc tends to the respective formal order of convergence. However, for ansatz orders five and six, the eoc does not agree with the formal order of convergence. In these cases, the numerical error is dominated by the error introduced due to round-off and not by

the discretization error. This effect is also visible in the subsequent examples. The results verify the tested capabilities of the code.

Example 3 *General geometry, exact representation of solution.* In the third example, a general geometry with displacement field A is considered. For the reference surface parametrization \bar{g}_4 from Table 3 is taken. The necessary body force for equilibrium is depicted in Fig. 6a at the reference surface. The distribution of the numerical error obtained with a coarse and a fine mesh is shown in Fig. 6b, c. Again, we observe strong oscillations in the numerical error in case of the fine mesh. The development of the errors and the eoc for ansatz orders up to order six are shown in Fig. 7. The numerical error present in the example stems from the numerical integration of the integrals in (13) and round-off since displacement field A can be exactly discretized.

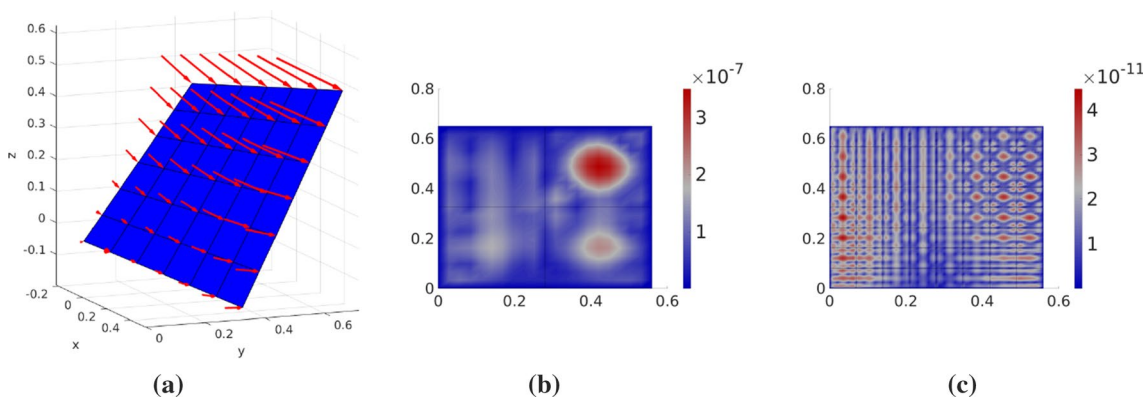


Fig. 6 Example 3: **a** reference surface and body force, **b** numerical error for a coarse mesh ($p = 4$), **c** numerical error for a fine mesh ($p = 4$)

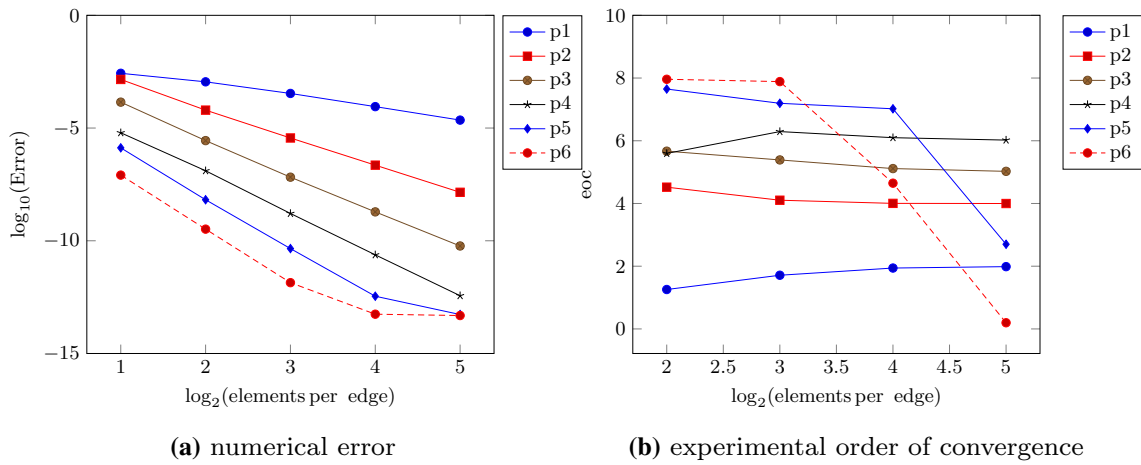


Fig. 7 Development of the numerical error and the eoc for Example 3 **a** numerical error **b** experimental order of convergence

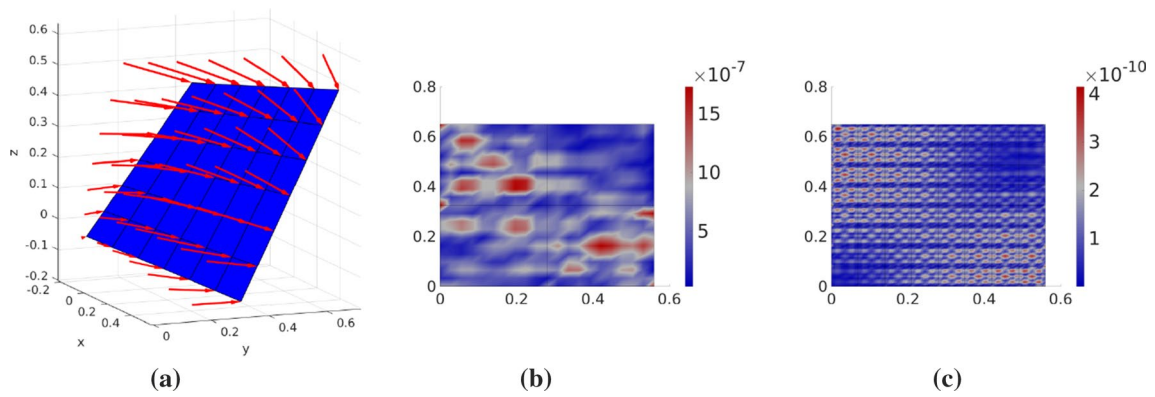


Fig. 8 Example 4: **a** reference surface and body force, **b** numerical error for a coarse mesh ($p = 5$), **c** numerical error for a fine mesh ($p = 5$)

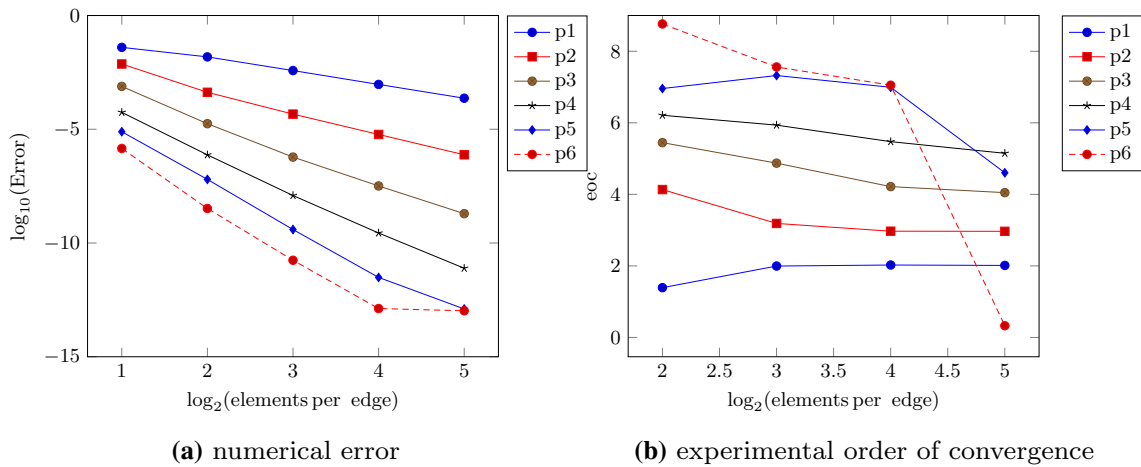


Fig. 9 Development of the numerical error and the eoc for Example 4 **a** numerical error **b** experimental order of convergence

We have set the number of quadrature points in both in-plane directions to the ansatz order. However, as long as the numerical error is dominated by the integration error, the eoc is higher or equal as the formal order of convergence regarding the discretization of the displacement field. Therefore, the integration error will not hamper the overall convergence rate. Note, with more quadrature points the eoc can be increased in this example.

Example 4 *General geometry, general solution.* The fourth example consists of the general surface parametrization \bar{q}_4 and the general displacement field B. Thus, this example assesses all features of the code. The reference surface and the body force are illustrated in Fig. 8a. In Fig. 8b, c, the distribution of the numerical error is depicted for a coarse and a fine mesh. Again, we observe oscillations of the error. However, the eoc is not deteriorated due to this effects. The development of the errors and the eoc for ansatz orders up to order six are shown in Fig. 9. Again, the error corresponding

to ansatz orders five and six is limited by the round-off error. Thus, the eoc drops. For all other ansatz orders the eoc tend to the formal order of convergence. Thus, the code passes this verification example.

4.2 Results for the Kirchhoff–Love model

Example 5 *Plane geometry, exact representation of solution.* The goal of this example is to determine the influence of round-off errors for the implementation of the FEM for the Kirchhoff–Love model. To this end, we have chosen the plane geometry given by the parametrization \bar{q}_1 from Table 3 and displacement field A from Table 2, which can be exactly represented in the discrete system. Due to this setting the body force vanishes for the Kirchhoff–Love model. The distribution of the numerical error for a coarse and a fine mesh is given in Fig. 10. For the coarse mesh we observe that the error is concentrated at the boundary. Due to the use of a penalty method this is reasonable. In contrast to this, for

Fig. 10 Example 5: **a** numerical error for a coarse mesh, **b** numerical error for a fine mesh

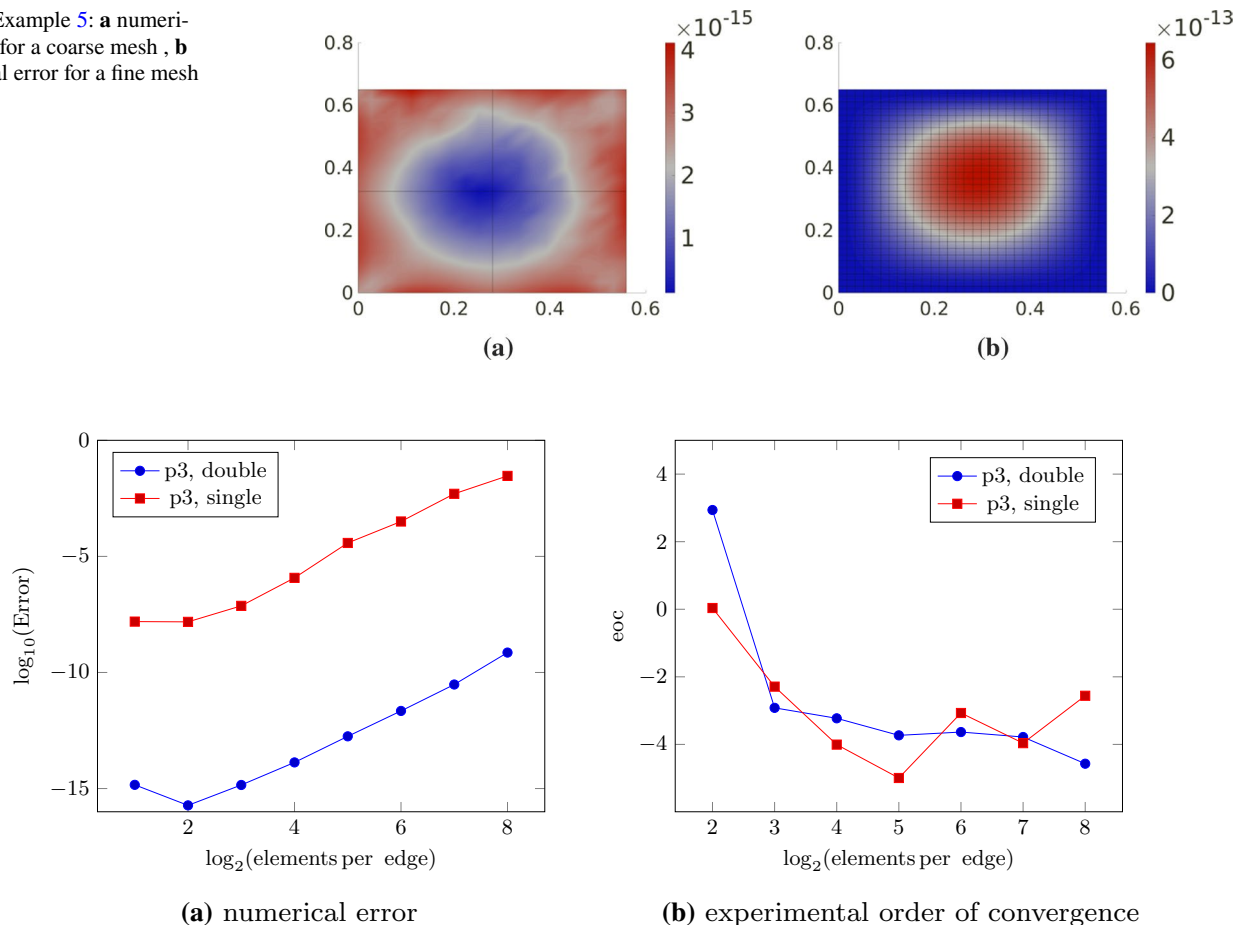


Fig. 11 Development of the numerical error and the eoc for Example 5 **a** numerical error **b** experimental order of convergence

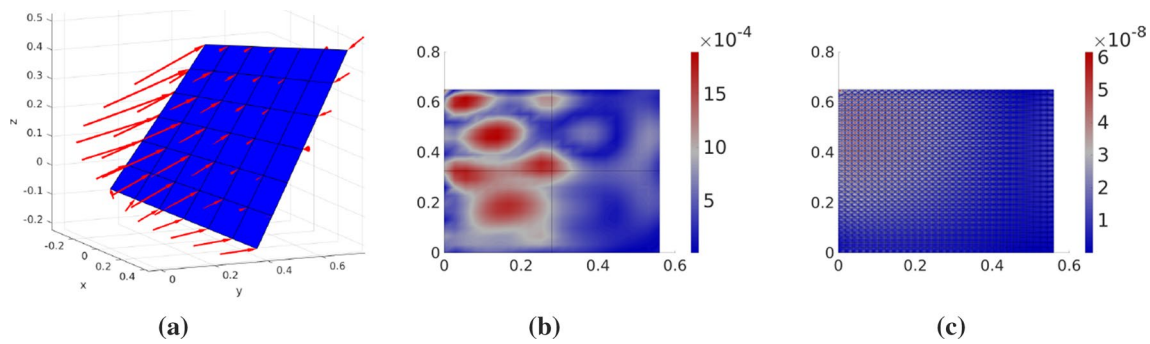


Fig. 12 Example 6: **a** reference surface and body force, **b** numerical error for a coarse mesh, **c** numerical error for a fine mesh

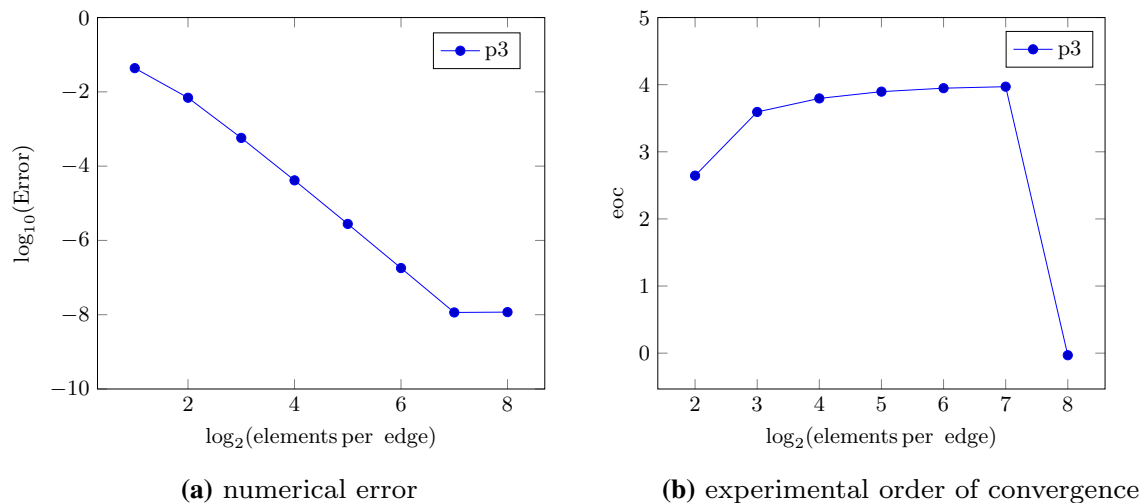


Fig. 13 Development of the numerical error and the eoc for Example 6 **a** numerical error **b** experimental order of convergence

the fine mesh the error is concentrated in the interior. This is in agreement with the results obtained in Example 1. In Fig. 11, the error and eoc for different discretization levels and single as well as double precision arithmetics are shown. From Fig. 11a it can be seen that the error increases with refinement. Furthermore, in Fig. 11b the eoc is at around minus four for single and double precision arithmetics. This is in agreement with the estimated condition number of the stiffness matrix, which raises at the same rate. Thus, we conclude that the numerical error is due to round-off errors and the penalty method used.

Example 6 *General geometry, general solution.* In this last example we use the parametrization $\bar{\mathbf{q}}_4$ and displacement field \mathbf{C} from Table 2. The reference surface and the body force obtained by the MMS is depicted in Fig. 12a. The distribution of the numerical error for a coarse and a fine mesh are illustrated in Fig. 12b, c. For the coarse mesh we observe

oscillations as in the examples for the Reissner–Mindlin model. The results of the performed convergence study are shown in Fig. 13. In Fig. 13a, we observe the convergence of the method up to an error level of about 10^{-8} . This final error level is in accordance with the results from Example 5. The eoc given in Fig. 13b tends to 4 apart for the last considered mesh refinement. Due to the use of third-order ansatz functions this convergence rate is optimal. Based on these results, we conclude that the code passes this general verification example.

5 Conclusion

Code verification of simulation tools is a necessary procedure in order to ensure the reliability. Especially in shell analysis complicated expressions arise which have to be coded. Thus, the error-proneness is high. During the preparation of the present paper, we found that performing order

of accuracy tests is a code verification procedure with high rigor.

In order to derive exact solutions we applied the MMS to the strong form of the 3D elasticity equations in curvilinear coordinates. We have used the computer algebra system Mathematica™ for the derivation of the source term function. In order to avoid long codes for this function we compute only the necessary derivatives analytically.

For code verification in the context of shell analysis we proposed different verification examples. We apply the proposed tests to an in-house developed finite element research code for Kirchhoff–Love and Reissner–Mindlin shell analysis. The examples are designed such that in some examples some parts of the code are not assessed consciously. This can help to identify coding faults. Furthermore, we have seen that not in all cases the optimal order of convergence has to show up. In cases where the discretization error is zero we observe the error due to round-off errors or if existing due to numerical integration.

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.

References

- Oberkampf WL, Roy CJ (2010) Verification and validation in scientific computing. Cambridge University Press, Cambridge
- Oden JT, Belytschko T, Fish J, Hughes T, Johnson C, Keyes D, Laub A, Petzold L, Srolovitz D, Yip S (2006) Revolutionizing engineering science through simulation. National Science Foundation Blue Ribbon Panel Report 65
- ASME V&V 10-2006 (2006) Guide for verification and validation in computational solid mechanics. The American Society of Mechanical Engineers, New York
- Oberkampf WL, Trucano TG, Hirsch C (2004) Verification, validation, and predictive capability in computational engineering and physics. *Appl Mech Rev* 57(5):345–384
- Roy CJ, Nelson C, Smith T, Ober C (2004) Verification of Euler/Navier–Stokes codes using the method of manufactured solutions. *Int J Numer Methods Fluids* 44(6):599–620
- Steinberg S, Roache PJ (1985) Symbolic manipulation and computational fluid dynamics. *J Comput Phys* 57(2):251–284
- Shih T (1985) A procedure to debug computer programs. *Int J Numer Methods Eng* 21(6):1027–1037
- Roache PJ (1998) Verification and validation in computational science and engineering. Hermosa, Albuquerque
- Eça L, Hoekstra M, Hay A, Pelletier D (2007) Verification of RANS solvers with manufactured solutions. *Eng Comput* 23(4):253–270
- Fisch R, Franke J, Wüchner R, Bletzinger KU (2013) Code verification examples of a fully geometrical nonlinear membrane element using the method of manufactured solutions. In: Bletzinger KU, Kröplin B, Oñate E (eds) Proceedings structural membranes, Munich
- Fisch R, Franke J, Wüchner R, Bletzinger KU (2014) Code verification of a partitioned FSI environment for wind engineering applications using the method of manufactured solutions. In: Oñate E, Oliver J, Huerta A (eds) 5th European conference on computational mechanics (ECCM V), Barcelona
- Fisch R (2014) Code verification of partitioned FSI environments for lightweight structures, Schriftenreihe des Lehrstuhls für Statik der Technischen Universität München, vol 23. Lehrstuhl für Statik, Technische Universität München
- Veeraragavan A, Beri J, Gollan R (2016) Use of the method of manufactured solutions for the verification of conjugate heat transfer solvers. *J Comput Phys* 307:308–320
- Kästner M, Metsch P, De Borst R (2016) Isogeometric analysis of the Cahn–Hilliard equation—a convergence study. *J Comput Phys* 305:360–371
- Bischoff M, Bletzinger KU, Wall W, Ramm E (2004) Models and finite elements for thin-walled structures. In: Stein E, de Borst R, Hughes T (eds) Encyclopedia of computational mechanics, vol 2. Wiley, Amsterdam, pp 59–137
- Benson D, Bazilevs Y, Hsu MC, Hughes T (2010) Isogeometric shell analysis: the Reissner–Mindlin shell. *Comput Methods Appl Mech Eng* 199(5):276–289
- Dornisch W, Müller R, Klinkel S (2016) An efficient and robust rotational formulation for isogeometric Reissner–Mindlin shell elements. *Comput Methods Appl Mech Eng* 303:1–34
- Hosseini S, Remmers JJ, Verhoosel CV, De Borst R (2014) An isogeometric continuum shell element for non-linear analysis. *Comput Methods Appl Mech Eng* 271:1–22
- Kiendl J, Bletzinger KU, Linhard J, Wüchner R (2009) Isogeometric shell analysis with Kirchhoff–Love elements. *Comput Methods Appl Mech Eng* 198(49):3902–3914
- Oesterle B, Ramm E, Bischoff M (2016) A shear deformable, rotation-free isogeometric shell formulation. *Comput Methods Appl Mech Eng* 307:235–255
- Guo Y, Ruess M (2015) Nitsches method for a coupling of isogeometric thin shells and blended shell structures. *Comput Methods Appl Mech Eng* 284:881–905
- Belytschko T, Stolarski H, Liu WK, Carpenter N, Ong JS (1985) Stress projection for membrane and shear locking in shell finite elements. *Comput Methods Appl Mech Eng* 51(1):221–258
- Szabo BA, Babuška I (1991) Finite element analysis. Wiley, Amsterdam
- Bogner F, Fox R, Schmit L (1965) The generation of inter-element-compatible stiffness and mass matrices by the use of interpolation formulas. In: Proceedings of the conference on matrix methods in structural mechanics, Wright-Patterson Air Force Base, Ohio
- Barrett JW, Elliott CM (1986) Finite element approximation of the dirichlet problem using the boundary penalty method. *Numer Math* 49(4):343–366
- Roy CJ (2005) Review of code and solution verification procedures for computational simulation. *J Comput Phys* 205(1):131–156
- Malaya N, Estacio-Hiroms KC, Stogner RH, Schulz KW, Bauman PT, Carey GF (2013) Masa: a library for verification using manufactured and analytical solutions. *Eng Comput* 29(4):487–496
- Wolfram Research Inc (2017) Mathematica, Version 11.1. Champaign, Illinois
- Strang G, Fix GJ (1973) An analysis of the finite element method, vol 212. Prentice-Hall, Englewood Cliffs
- Roache PJ (2002) Code verification by the method of manufactured solutions. *J Fluids Eng* 124(1):4–10