ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

CODE VERIFICATION IN SHELL ANALYSIS BY THE METHOD OF MANUFACTURED SOLUTIONS

Michael Gfrerer, Martin Schanz

Graz University of Technology, Institute of Applied Mechanics
Technikerstr. 4, 8010 Graz, Austria
e-mail: m.gfrerer@tugraz.at

Keywords: verification, elasticity, shell analysis, manufactured solutions

Abstract.

The simulation of the mechanical response of shell structures is of great interest in engineering. Nevertheless, creating simulation software is an error prone task, which needs reliable tests to identify code faults. The aim of this paper is to propose a code verification procedure for shell analysis using the method of manufactured solutions.

We apply order of accurecy tests, where the reference solutions are constructed with the method of manufactured solutions. A set of five verification examples is presented. We have chosen the first example such that the numerical error arises only because of round-off errors. The second and the third example consider either a general stress state or a general geometry. The fourth example considers both a general geometry and a general stress state. We show convergence orders for a Matlab code which implements high order finite elements for a shell model based on the Reissner-Mindlin kinematics. Furthermore, the process of manufactured solution generation is discussed.

1 INTRODUCTION

In order to ensure the reliability of a numerical simulation software, verification and validation ($V\&V$) are unavoidable tasks [1, 2]. Figure 1 illustrates the relation between these activities. An observed real world phenomenon can be modeled by mathematical means, leading to a explicit mathematical model. Since the applicability of the Method of Exact Solutions (MES) to mathematical models describing real world problems is limited, numerical solutions are inevitable. The computer implementation of the mathematical model leads to the computational model. Each of these steps introduces errors, which have to be assessed by $V\&V$. The process of validation determines how accurate the mathematical model represents the real (physical) problem. The aim of verification is to show that the numerical method is able to produce a solution which approximates the exact solution of the mathematical model. Thus, one is interested in the numerical error, which is defined as the difference between the numerical solution and the exact solution. The comparison of numerical results with measurement 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 solution verification. Code verification represents the process of demonstrating that the governing equations, as implemented in the code, are solved consistently. Solution 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) [6, 7, 8, 5, 1].

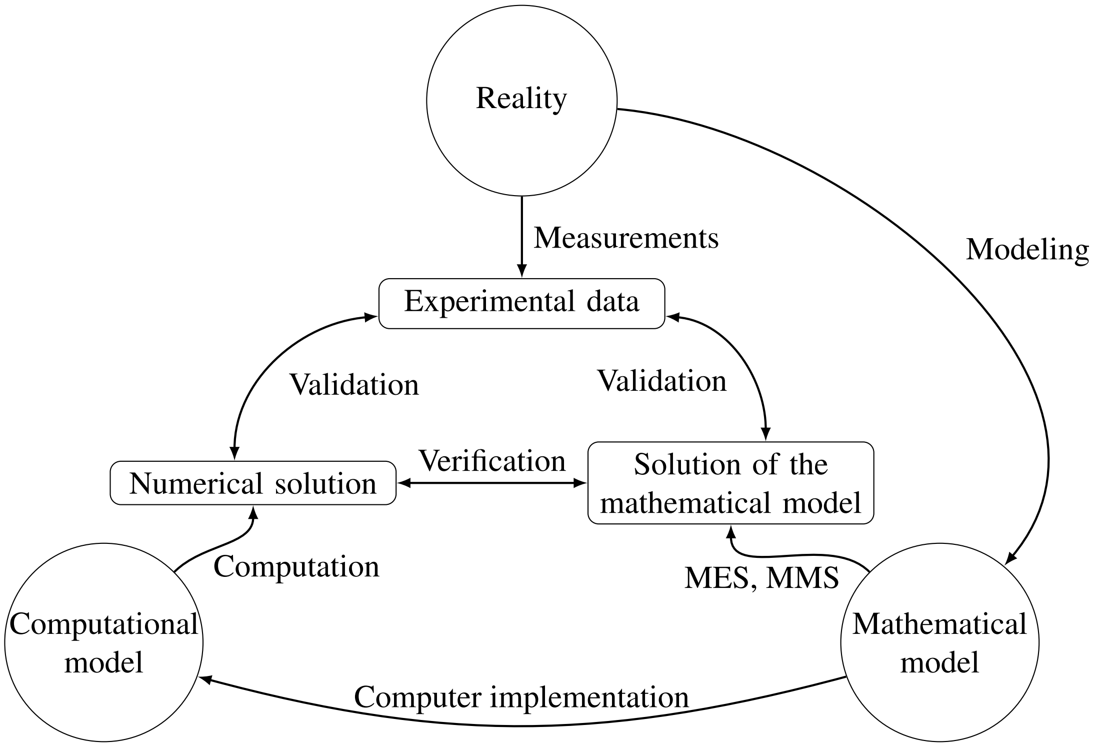

Figure 1: The role of $V\&V$ in numerical simulation, based on [3]

Its central idea is to prescribe a solution and to determine a 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 shell analysis, it is a common practice to verify the code against benchmark solutions. One well known set is the shell obstacle course [16], which consists of the Scordelis-Lo 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. One shortcoming of this test set when used for code verification 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 Matlab [17] research code, which implements high order finite elements for the Reissner-Mindlin theory.

2 SHELL ANALYSIS

The starting point of the mathematical modeling of a shell structure can either be a two dimensional surface (direct approach), or a three dimensional body, both equipped with mechanical properties. The direct approach is based on a deformable Cosserat surface, and has been reviewed in [18]. Its computational aspects have been addressed in a series of papers [19, 20, 21, 22]. Since the strong form of the governing equations is available, the application of the MMS poses no difficulties, other than the necessity of being general.

In case of a three dimensional body which occupies the domain Ω in space, one is interested in solving an elasticity boundary value problem of the form: find the displacement field $\tilde{\mathbf{u}}$ such that

$$\operatorname{div}(\boldsymbol{\sigma}) + \mathbf{b} = 0 \qquad \text{in } \Omega, \tag{1a}$$

$$\boldsymbol{\sigma} = W(\mathbf{E}(\tilde{\mathbf{u}})), \tag{1b}$$

$$\mathbf{u} = \mathbf{u}_D \qquad \text{on } \Gamma_D, \tag{1c}$$

$$\mathbf{t} = \mathbf{t}_N \qquad \text{on } \Gamma_N \tag{1d}$$

are fulfilled. The equilibrium equations are given in (1a), where $\boldsymbol{\sigma}$ stands for the stress tensor and $\mathbf{b}$ for the volume load. In (1b) the material law W assigns a stress tensor to a strain tensor $\mathbf{E}$, which is a function of the displacement field $\tilde{\mathbf{u}}$. The boundary of Ω is denoted by $\Gamma = \Gamma_D \cup \Gamma_N$. On the Dirichlet boundary Γ_D, the displacement vector has a given value $\mathbf{u}_D$, whereas on the Neumann boundary Γ_N, the traction vector $\mathbf{t} = \boldsymbol{\sigma}\mathbf{n}$ has a given value $\mathbf{t}_N$. For a detailed presentation and mathematical foundation of the theory of elasticity, we refer to [23] and [24].

For a given problem, the modeling steps involve the specification of the domain Ω, the respective parts of the boundary Γ, the material law and the strain measure. Finite element solution strategies to (1) can be broadly classified in 3D solid elements, degenerated shell elements, and shell elements based on a shell theory. Within the 3D solid elements approach, the unmodified problem (1) is solved. Thus, the application of the MMS is straightforward. Degenerated shell elements have been introduced in [25] and are based on a continuum discretization of Ω. Further simplifications of the kinematics are introduced in a second step. A shell theory introduces the kinematical assumptions on Ω on the continuous level. Although the degeneration concept and the shell theory approach seem quite different, the formulations are quite similar when based on the same assumptions [26].

In the present paper, we concentrate on linearized elasticity with the following geometric setup. We consider a shell as a three dimensional body which can be described by a two dimensional surface and a thickness parameter. We assume that the surface can be parametrized by a C^2 mapping from a parameter space $\bar{U}$ into the physical space

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

This makes it possible to define two covariant base vectors of the surface

$$\overline{\mathbf{G}}_\alpha = \frac{\partial}{\partial \theta^\alpha} \overline{g} \qquad \alpha = 1, 2, \tag{3}$$

and the normal vector to the surface

$$\mathbf{n} = \frac{\tilde{\mathbf{n}}}{\|\tilde{\mathbf{n}}\|} \quad , \quad \tilde{\mathbf{n}} = \overline{\mathbf{G}}_1 \times \overline{\mathbf{G}}_2. \tag{4}$$

The three dimensional body is parametrized by

$$\begin{aligned} g:\ (\bar{U} \times T) \subset \mathbb{R}^3 &\to \Omega \subset \mathbb{R}^3 \\ (\theta^1, \theta^2) \times \theta^3 &\mapsto g(\theta^1, \theta^2, \theta^3) = \overline{g}(\theta^1, \theta^2) + \theta^3\, \mathbf{n}. \end{aligned} \tag{5}$$

We assume $T = [-{}^t\!/\!_2, {}^t\!/\!_2]$, with constant thickness t. Instead of searching for $\tilde{\mathbf{u}}$ defined on Ω, we refer to $\mathbf{u}$ defined on $\bar{U} \times T$. Then the components of the linearized strain tensor $\boldsymbol{\epsilon} = \epsilon_{ij} \mathbf{G}^i \otimes \mathbf{G}^j$ with respect to curvilinear coordinates are given by

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

The components of the elasticity tensor $\mathbb{C} = \mathbb{C}^{ijkl} \mathbf{G}_i \otimes \mathbf{G}_j \otimes \mathbf{G}_k \otimes \mathbf{G}_l$ for a linear isotropic material behavior are

$$\begin{aligned} \mathbb{C}^{\alpha\beta\gamma\varphi} &= \lambda \mathrm{G}^{\alpha\beta} \mathrm{G}^{\gamma\varphi} + \mu \left(\mathrm{G}^{\alpha\gamma} \mathrm{G}^{\beta\varphi} + \mathrm{G}^{\alpha\varphi} \mathrm{G}^{\beta\gamma} \right), \\ \mathbb{C}^{\alpha\beta 33} &= \mathbb{C}^{33\alpha\beta} = \lambda \mathrm{G}^{\alpha\beta}, \\ \mathbb{C}^{3\alpha 3\beta} &= \mathbb{C}^{3\alpha\beta 3} = \mathbb{C}^{\alpha 33\beta} = \mathbb{C}^{\alpha 3\beta 3} = \mu \mathrm{G}^{\alpha\beta}, \\ \mathbb{C}^{3\alpha\beta\gamma} &= \mathbb{C}^{\alpha 3\beta\gamma} = \mathbb{C}^{\alpha\beta 3\gamma} = \mathbb{C}^{\alpha\beta\gamma 3} = 0, \\ \mathbb{C}^{333\alpha} &= \mathbb{C}^{33\alpha 3} = \mathbb{C}^{3\alpha 33} = \mathbb{C}^{\alpha 333} = 0, \\ \mathbb{C}^{3333} &= \lambda + 2\mu. \end{aligned} \tag{7}$$

In equations (6) and (7), $\mu_\alpha^\xi = \delta_\alpha^\xi - \theta^3 \overline{G}^{\xi\gamma}(\overline{\mathbf{G}}_{\alpha,\gamma} \cdot \mathbf{n})$ are the components of the shifter tensor, $\overline{G}^{\alpha\beta}$ are the components of the inverse of the metric tensor $\overline{G}_{\alpha\beta} = \overline{\mathbf{G}}_\alpha \cdot \overline{\mathbf{G}}_\beta$, and $\mathrm{G}^{\alpha\beta}$ are the components of the inverse of $\mathrm{G}_{\alpha\beta} = \mu_\alpha^\gamma \mu_\beta^\varphi \overline{G}_{\gamma\varphi}$. λ, μ are the Lamé constants. For the sake of brevity, we used the notation $(\cdot)_{,\alpha} = \frac{\partial(\cdot)}{\partial \theta^\alpha}$.

In the present paper, we consider the classical Reissner-Mindlin model, where the thickness expansion for the displacement field reads

$$\mathbf{u}(\theta^1, \theta^2, \theta^3) = u^i(\theta^1, \theta^2)\mathbf{e}_i + \theta^3 \sum_{\alpha=1}^{2} v^\alpha \overline{\mathbf{G}}_\alpha. \tag{8}$$

In (8), u^i are the displacement components of the mid surface and v^α are the rotations. In total, five parameters are used. Furthermore, as the result of the plane stress assumption, λ has to be replaced with $\frac{2\mu\lambda}{2\mu+\lambda}$.

The finite element formulation is based on the virtual work

$$\int_\Omega \hat{\boldsymbol{\epsilon}} : \mathbb{C} : \boldsymbol{\epsilon} \,\mathrm{d}x = \int_\Omega \hat{\mathbf{u}} \cdot \mathbf{b} \,\mathrm{d}x + \int_\Gamma \hat{\mathbf{u}} \cdot \mathbf{t} \,\mathrm{d}s_x, \tag{9}$$

where $\hat{\mathbf{u}}$ is the virtual displacement field and $\hat{\boldsymbol{\epsilon}} = \boldsymbol{\epsilon}(\hat{\mathbf{u}})$. In order to derive finite elements, the parameter space is meshed with quadrilateral elements. The parameters of the shell model are discretized with integrated Legendre polynomials of order p. The discretized virtual work statement (9) results in the linear equation system

$$K\,u = f, \tag{10}$$

where K is the stiffness matrix resulting from the integral on the left side of (9), u the unknown degrees of freedom and f the load vector resulting from the integrals on the right side of (9). All integrals are computed using Gauss-Legendre quadrature for in-plane as well as through the thickness. The system of linear equations (10) is solved with a direct solver.

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 section 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

$$||u^{exact} - u^{numerical}|| \leq C\,h^q\,||u^{exact}||, \tag{11}$$

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

$$eoc = \frac{\log(e_1) - \log(e_2)}{\log(h_1) - \log(h_2)}, \tag{12}$$

where

$$e_i = ||u^{exact} - u^{numerical}_{h_i}|| \tag{13}$$

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.

In order to evaluate (13), an exact solution has to be available. Within the MMS, such a solution is prescribed. 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)
2. Choose the form of the manufactured solution $\mathbf{u}^M$, $i.e$ specify the parameters in (8)
3. Derive the modified governing equations, $i.e.$ compute an artificial source term $\mathbf{b}^M = -\mathrm{div}(\mathbb{C} : \epsilon(\mathbf{u}^M))$
4. Solve the discrete form of the modified governing equations on multiple meshes, $i.e.$ solve (9) with $\mathbf{b} = \mathbf{b}^M$
5. Evaluate the numerical error (13) and the eoc (12)
6. Apply the order of convergence test

Additionally to [27], 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

$$b^i = -\left(\sigma^{jl}_{,j} + \sigma^{kl}\Gamma_{kj}{}^{j} + \sigma^{jk}\Gamma_{kj}{}^{l}\right) J^i_l \tag{14}$$

with the Christoffel symbols of second kind $\Gamma_{kj}{}^{l} = \mathbf{G}_{k,j}\mathbf{G}^l$ and $J^i_l = \mathbf{G}_l \cdot \mathbf{e}^i$. As noted in [28], 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$^{\text{TM}}$ [29] 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 numerize these quantities and compute

$$
\begin{aligned}
G_{ij} &= \mathbf{G}_i \cdot \mathbf{G}_j, \\
[G^{ij}] &= [G_{ij}]^{-1}, \\
J^i_l &= \mathbf{G}_l \cdot \mathbf{e}^i, \\
\Gamma_{kj}{}^{l} &= \mathbf{G}_{k,j} \cdot \mathbf{G}_i G^{il}, \\
G^{ij}_{,k} &= -\Gamma_{kl}{}^{j} G^{li} - \Gamma_{kl}{}^{i} G^{lj}, \\
\sigma^{jl} &= \lambda \left(G^{ij} G^{kl} \epsilon_{kl} + 2\mu G^{ik} G^{jl} \epsilon_{kl} \right), \\
\sigma^{jl}_{,j} &= \lambda \left(G^{ij}_{,j} G^{kl} \epsilon_{kl} + G^{ij} G^{kl}_{,j} \epsilon_{kl} + G^{ij} G^{kl} \epsilon_{kl,j} \right) \\
&\quad + \mu \left(G^{ik}_{,j} G^{jl} \epsilon_{kl} + G^{ik} G^{jl}_{,j} \epsilon_{kl} + G^{ik} G^{jl} \epsilon_{kl,j} \right).
\end{aligned}
\tag{15}
$$

Then the source term is obtained with (14).

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]

$$
e = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(u^{exact}(x_n) - u^{numerical}(x_n) \right)^2}. \tag{16}
$$

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 [30].

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 [31]. However, in order to have confirmation exercises, we suggest special cases where only parts of the code should be exercised.

In all examples, we specify a surface parametrization (2) and the displacement parameters for the shell model (8). In all examples, we use the parameter space $(\theta^1, \theta^2) \in [0, 0.56] \times [0, 0.65]$ and the thickness $t = 0.07$. All quantities are defined in the International System of Units (SI) and their derived expressions. Therefore, we omit units for the input numbers. We use the material parameters $\lambda = \mu = 4000$, which corresponds to Young's modulus $E = 1000$ and Poisson's ratio of $\nu = 0.25$.

In the verification examples, we use two types of displacement fields. 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. The displacement fields for the shell model are given in Table 1.

Table 1: Prescribed displacement parameters for (8)

	displacement field A	displacement field B
$u^1 =$	θ^1	$\sin(\pi\theta^1)\cos(\pi\theta^2)$
$u^2 =$	θ^2	$\cos(\pi\theta^1)\sin(\pi\theta^2)$
$u^3 =$	$\theta^1\theta^2$	$\sin(\pi\theta^1\theta^2)$
$v^1 =$	$\theta^1\theta^2$	$\sin(\pi\theta^1\theta^2)$
$v^2 =$	$\theta^1\theta^2$	$\sin(\pi\theta^1\theta^2)$

4.1 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

$$\begin{aligned} x &= \theta^1, \\ y &= \theta^2, \\ z &= 0, \end{aligned} \tag{17}$$

and displacement field A, which can be exactly represented in the discrete system. 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. In Figure 2, the errors for different discretization levels are shown. The error levels are in accordance with single and double precision.

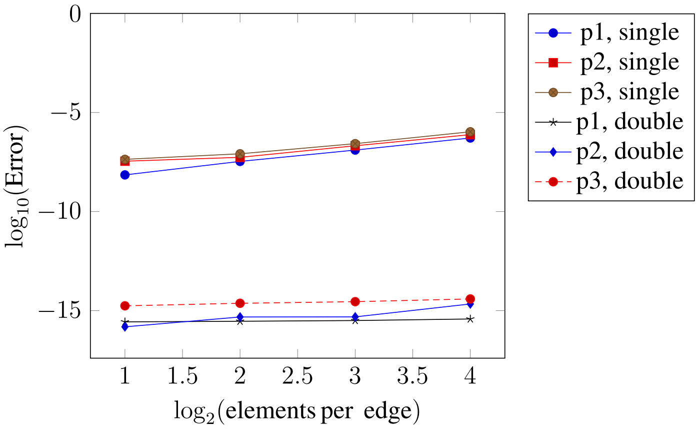

Figure 2: Numerical error for example 1

4.2 Example 2: plane geometry, general solution

In the second example, the discretization error at a plane geometry is assessed. We therefore use (17) with displacement field B. The development of the errors and the *eoc* for ansatz orders up to order six are shown in Figure 3. 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.

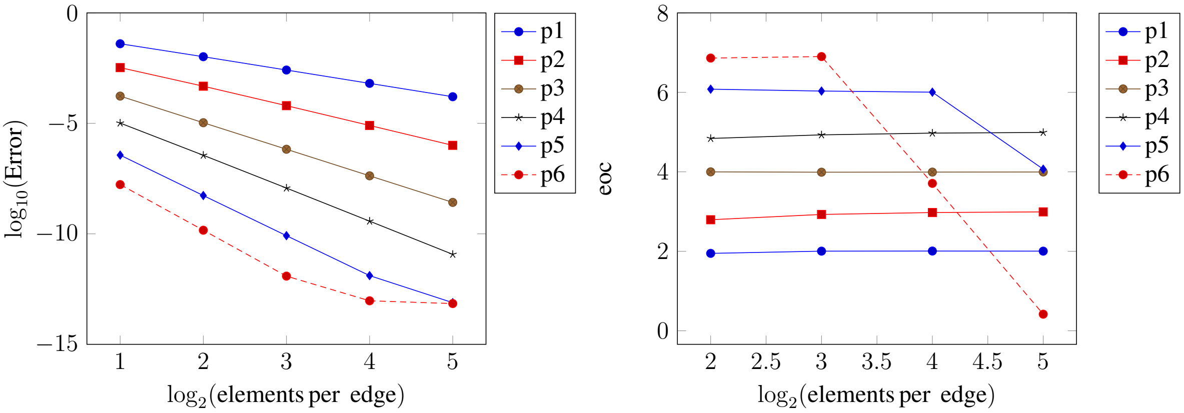

Figure 3: Development of the numerical error and the *eoc* for example 2

4.3 Example 3: general geometry, exact representation of solution

In the third example, a general geometry with displacement field A is considered. The surface parametrization reads

$$\begin{aligned} x &= \theta^1, \\ y &= \theta^2, \\ z &= (\theta^1)^2 - (\theta^2)^2. \end{aligned} \tag{18}$$

The development of the errors and the *eoc* for ansatz orders up to order six are shown in Figure 4. The numerical error present in the example stems from the numerical integration of the integrals in (9) and round-off, since displacement field A can be exactly discretized. 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. We remark that, with more quadrature points the *eoc* can be increased in this example.

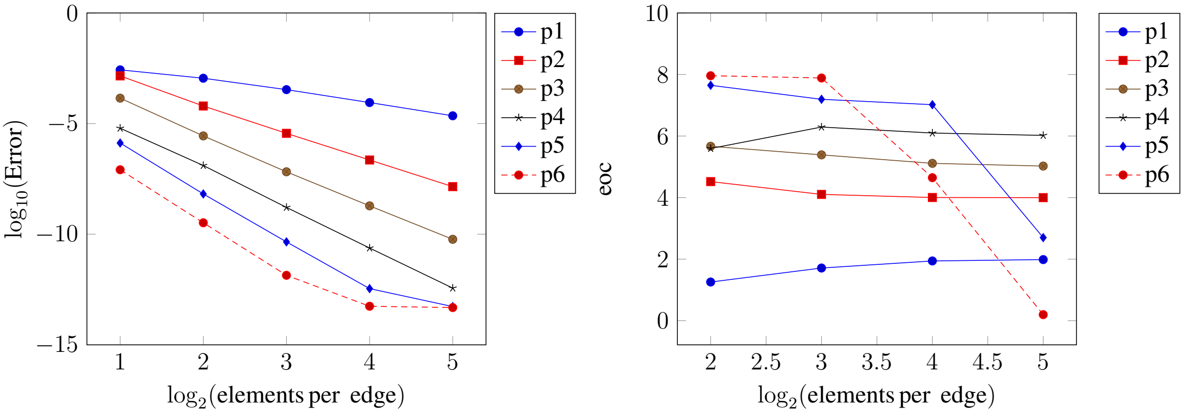

Figure 4: Development of the numerical error and the *eoc* for example 3

4.4 Example 4: general geometry, general solution

The fourth example consists of the general surface parametrization

$$\begin{aligned} x &= \theta^1 + \frac{(\theta^2)^2}{2}, \\ y &= \theta^2 - (\theta^1)^2, \\ z &= (\theta^1)^2 - (\theta^2)^2, \end{aligned} \tag{19}$$

and the general displacement field B. Thus, this example assesses all features of the code. The development of the errors and the eoc for ansatz orders up to order six are shown in Figure 5. 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.

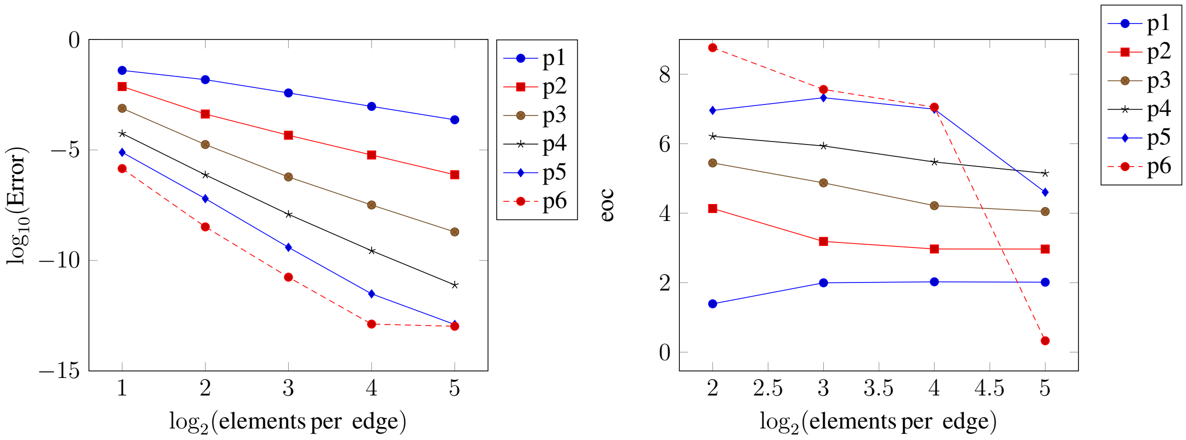

Figure 5: Development of the numerical error and the eoc for example 4

5 CONCLUSIONS

This paper addresses code verification of a finite element code for shell analysis based on the MMS. The MMS provides a powerful procedure to construct exact solutions to the governing equations. Because of the complexity of the resulting source term, we decided to partly numerized its computation. The derivatives are computed analytically and the remaining algebraic operations are performed on numerized quantities. We successfully apply the verification examples to a finite element code for Reissner-Mindlin shells. The approach based on automatic differentiation presented in [28] could further reduce the source term complexity. Nevertheless, the verification of the manufactured solution itself remains an open question.

REFERENCES

[1] W. L. Oberkampf, C. J. Roy, *Verification and validation in scientific computing*. Cambridge University Press, 2010.

[2] J. T. Oden, T. Belytschko, J. Fish, T. Hughes, C. Johnson, D. Keyes, A. Laub, L. Petzold, D. Srolovitz, S. Yip, Revolutionizing engineering science through simulation. *National Science Foundation Blue Ribbon Panel Report*, **65**, 2006.

[3] S. Schlesinger, R. E. Crosbie, R. E. Gagné, G. S. Innis, C. S. Lalwani, J. Loch, R. J. Sylvester, R. D. Wright, N. Kheir, D. Bartos, Terminology for model credibility. *Simulation*, **32**, 103–104, 1979.

[4] W. L. Oberkampf, T. G. Trucano, C. Hirsch, Verification, validation, and predictive capability in computational engineering and physics. *Applied Mechanics Reviews*, **57**, 345–384, 2004.

[5] C. J. Roy, C. Nelson, T. Smith, C. Ober, Verification of Euler/Navier–Stokes codes using the method of manufactured solutions. *International Journal for Numerical Methods in Fluids*, **44**, 599–620, 2004.

[6] S. Steinberg, P. J. Roache, Symbolic manipulation and computational fluid dynamics. *Journal of Computational Physics*, **57**, 251–284, 1985.

[7] T. Shih, A procedure to debug computer programs. *International Journal for Numerical Methods in Engineering*, **21**, 1027–1037, 1985.

[8] P. J. Roache, *Verification and validation in computational science and engineering*. Hermosa, 1998.

[9] L. Eça, M. Hoekstra, A. Hay, D. Pelletier, Verification of RANS solvers with manufactured solutions. *Engineering with computers*, **23**, 253–270, 2007.

[10] R. Fisch, J. Franke, R. Wüchner, K. Bletzinger, Code verification examples of a fully geometrical nonlinear membrane element using the method of manufactured solutions. K.-U.Bletzinger, B. Kröplin, E. Oñate, eds., *Proceedings Structural Membranes 2013*, Munich, Germany, October 9-11, 2013.

[11] R. Fisch, J. Franke, R. Wüchner, K. Bletzinger, Code Verification of a Partitioned FSI Environment for Wind Engineering Applications using the Method of Manufactured Solutions. E. Oñate, J. Oliver, A. Huerta, eds., *5th European Conference on Computational Mechanics (ECCM V)*, Barcelona, Spain, July 20-15, 2014.

[12] R. Fisch, *Code Verification of Partitioned FSI Environments for Lightweight Structures*, Vol. 23 of *Schriftenreihe des Lehrstuhls für Statik der Technischen Universität München*. Lehrstuhl für Statik, Technische Universität München, 2014.

[13] A. Veeraragavan, J. Beri, R. Gollan, Use of the method of manufactured solutions for the verification of conjugate heat transfer solvers. *Journal of Computational Physics*, **307**, 308–320, 2016.

[14] M. Kästner, P. Metsch, R. De Borst, Isogeometric analysis of the Cahn–Hilliard equation–a convergence study. *Journal of Computational Physics*, **305**, 360–371, 2016.

[15] M. Bischoff, K.-U. Bletzinger, W. Wall, E. Ramm, Models and Finite Elements for Thin-Walled Structures. E. Stein, R. de Borst, T. Hughes, eds., *Encyclopedia of computational mechanics*, Vol. 2, 59–137, Wiley Online Library, 2004.

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

[17] MATLAB, *Version 8.5.0.197613 (R2015a)*. The MathWorks Inc., Natick, Massachusetts, 2015.

[18] J. Altenbach, H. Altenbach, V. A. Eremeyev, On generalized Cosserat-type theories of plates and shells: a short review and bibliography. *Archive of Applied Mechanics*, **80**, 73–92, 2010.

[19] J. C. Simo, D. D. Fox, On a stress resultant geometrically exact shell model. Part I: Formulation and optimal parametrization. *Computer Methods in Applied Mechanics and Engineering*, **72**, 267–304, 1989.

[20] J. C. Simo, D. D. Fox, M. Rifai, On a stress resultant geometrically exact shell model. Part II: The linear theory; computational aspects. *Computer Methods in Applied Mechanics and Engineering*, **73**, 53–92, 1989.

[21] J. C. Simo, D. D. Fox, M. S. Rifai, On a stress resultant geometrically exact shell model. Part III: Computational aspects of the nonlinear theory. *Computer Methods in Applied Mechanics and Engineering*, **79**, 21–70, 1990.

[22] J. C. Simo, M. Rifai, D. Fox, On a stress resultant geometrically exact shell model. Part IV: Variable thickness shells with through-the-thickness stretching. *Computer methods in applied mechanics and engineering*, **81**, 91–126, 1990.

[23] J. E. Marsden, T. J. Hughes, *Mathematical foundations of elasticity*. Dover Publications, New York, 1983.

[24] G. A. Holzapfel, *Nonlinear solid mechanics*, Vol. 24. Wiley Chichester, 2000.

[25] S. Ahmad, B. M. Irons, O. Zienkiewicz, Analysis of thick and thin shell structures by curved finite elements. *International Journal for Numerical Methods in Engineering*, **2**, 419–451, 1970.

[26] N. Buechter, E. Ramm, Shell theory versus degenerationa comparison in large rotation finite element analysis. *International Journal for Numerical Methods in Engineering*, **34**, 39–59, 1992.

[27] C. J. Roy, Review of code and solution verification procedures for computational simulation. *Journal of Computational Physics*, **205**, 131–156, 2005.

[28] N. Malaya, K. C. Estacio-Hiroms, R. H. Stogner, K. W. Schulz, P. T. Bauman, G. F. Carey, MASA: a library for verification using manufactured and analytical solutions. *Engineering with Computers*, **29**, 487–496, 2013.

[29] Wolfram Research, Inc., *Mathematica, Version 10.0*. Champaign, Illinios, 2014.

[30] G. Strang, G. J. Fix, *An analysis of the finite element method*, Vol. 212. Prentice-Hall Englewood Cliffs, N.J., 1973.

[31] P. J. Roache, Code verification by the method of manufactured solutions. *Journal of Fluids Engineering*, **124**, 4–10, 2002.