VI International Conference on Textile Composites and Inflatable Structures
STRUCTURAL MEMBRANES 2013
K.-U.Bletzinger, B. Kröplin and E. Oñate (Eds)

CODE VERIFICATION EXAMPLES OF A FULLY GEOMETRICAL NONLINEAR MEMBRANE ELEMENT USING THE METHOD OF MANUFACTURED SOLUTIONS

RUPERT FISCH*, JÖRG FRANKE†, ROLAND WÜCHNER* AND KAI-UWE BLETZINGER*

*Chair of Structural Analysis
Technische Universität München (TUM)
Arcisstraße 21, 80333 Munich, Germany
e-mail: rupert.fisch@tum.de, web page: www.st.bgu.tum.de

† Vietnamese-German University (VGU)
Le Lai Street, Hoa Phu Ward, Binh Duong New City, Binh Duong Province, Vietnam
e-mail: joerg.franke@vgu.edu.vn, Web page: www.vgu.edu.vn

Key words: Code Verification, Benchmarks, Method of Manufactured Solutions, MMS, Nonlinear Membrane Element

Abstract. This paper presents an effective method to perform Code Verification of a software which is designed for structural analysis using membranes. The focus lies on initially curved structures with large deformations in steady and unsteady regimes. The material is assumed to be linear elastic isotropic. Code Verification is a part of efforts to guarantee the code's correctness and to obtain finally predictive capability of the code. The Method of Manufactured Solutions turned out to be an effective tool to perform Code Verification, especially for initially curved structures. Here arbitrary invented geometries and analytical solutions are chosen. The computer code must approach this solution asymptotically. The observed error reduction with systematic mesh refinement (i.e. observed order of accuracy) must be in the range of the formal order of accuracy, e.g. derived by a Taylor series expansion. If these two orders match in the asymptotic range, the implemented numerical algorithms are working as intended. The given examples provide a complete hierarchical benchmark suite for the reader to assess other codes, too. In the present case several membrane states were tested successfully and the used code Carat++ assessed to converge - as intended - second order accurately in space and time for all kind of shapes and solutions.

1 INTRODUCTION

In the context of quantitative accuracy assessment of computer codes and their results Verification and Validation activities (V&V) are inevitable. They provide evidences for the

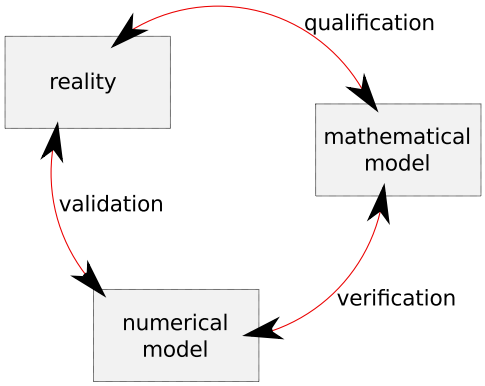

Figure 1: Integration of Verification and Validation in the modeling process

code's and the results' correctness. V&V activities can be separated in three parts: Code Verification, Solution Verification, and Model Validation. Code Verification represents the mathematical procedure to demonstrate that the governing equations, as implemented in the code, are solved consistently [17]. Solution Verification, often seen as part of validation, is the assessment of the correctness and accuracy of obtained solutions for a real (physical) problem if interest and Model Validation finally is the assessment of the model accuracy by comparison with experimental measurements. The relation between these activities and the different model parts are shown in figure 1. More detailed explanations of the topics of V&V and their most common definitions can be found in [10, 13, 15]. Roache gave a much simpler, but also very popular definition of Verification and Valdation in [15]. He states that Verification is the purely mathematical discipline to assess that a code is "solving the equations right" and Validation is the discipline to assess that the code is "solving the right equations".

2 CODE VERIFICATION

Code Verification can be performed in many different ways. As a stair of increasing rigor Oberkampf names the following methods [13]: simple tests, code-to-code comparisons, discretization error quantification, convergence tests, and order-of-accuracy tests. The first two are the weakest methods but they don't need exact solutions of the problem. The latter three methods need exact solutions [13]. This paper focusses on the last method, the order-of-accuracy tests. Here one assesses, if the discretization error reduces with the formal order of the used discretization schemes. For an exact error evaluation of simulations the availability of analytical solutions is a prerequisite for the method. Unfortunately only a limited number of analytical solutions are available. The available test cases are usually much simpler then the application of interest, which reduces their practival value. This is especially the case for initially curved and prestressed, dynamically loaded structures. All other available, highly accurate solutions are insufficient to evaluate the exact discretization error of the simulations. Hence the Method of Manufactured Solutions (MMS) is used.

3 METHOD OF MANUFACTURED SOLUTIONS

Within the order-of-accuracy tests, the MMS provides a framework to generate artificial analytical solutions. It has been used in fluid dynamics [3, 5, 10, 13, 15–17], but also in structural stress computations [2], or recently in monolithic fluid-structure interaction computations [6]. The method can be used as a toolbox to assess implemented discrete procedures for the solution of differential equation(s). A major advantage of the method is the independence of the numerical approach: the method formulates the equations of state in the continuum, therefore it is independent of the discretization method (e.g. Finite Elements, Finite Differences, ...) or the solution procedure (direct solution, fix-point iteration, Newton-Raphson, ...). The main concept of MMS is that an own chosen target solution $\hat{\mathbf{d}}$ represents an analytical (but not necessarily a physically realistic) solution for the primary variables $\mathbf{d}$ of the differential equations in the continuum. After insertion of the target solution into the differential equations a source/force term remains, as the equilibrium no longer holds for the arbitrary solution $\hat{\mathbf{d}}$. The continuous source/force term can be obtained by hand or using symbolic manipulation software like Maple®. If the discrete schemes are correctly implemented one can observe that $\mathbf{d}$ tends towards $\hat{\mathbf{d}}$ for systematically refined meshes. The difference between $\mathbf{d}$ and $\hat{\mathbf{d}}$ represents the exact error of the individual calculation. The development of the error with mesh refinement (mesh refinement factor $r = \frac{h_{coarse}}{h_{fine}}$ with a characteristic element size h) gives the observed order of accuracy $\hat{p}$ in equation 1.

$$\hat{p} = \frac{\log\left(\frac{E_{h_{coarse}}}{E_{h_{fine}}}\right)}{\log\left(r\right)} \tag{1}$$

If the formal order of accuracy p (cf. chapter 6) matches $\hat{p}$ in the asymptotic range of the solution, the following parts of the code have been verified [15]: all coordinate transformations, the order and the programming of the discretization, and the matrix solution procedure. The procedure with all its parts is shown in detail in chapter 7.

If the two orders do not match, there can be many reasons, e.g. programming errors, insufficient grid resolution, singularities, etc. A complete list of reasons is discussed in [14]. MMS is not able to verify individual terms in mixed-order methods. Additionally it is not able to judge the efficiency of a solution process, e.g. the speed of convergence of a nonlinear problem or the iterative convergence rate of steady-state calculations [15]. The prerequisites for the application of the MMS can be found in detail e.g. in [17]. The basic requirement is that the target solution $\hat{\mathbf{d}}$ represents a smooth analytical function with a sufficient number of derivatives. Additionally it should not contain singularities in the function or its derivatives. Furthermore the solution $\hat{\mathbf{d}}$ should be general enough and well balanced to activate all terms of the governing equations. In this paper the given solutions give a stair of complexity or even a benchmark series to assess all containing parts step by step. This stair can be very helpful to localize coding or other mistakes [10]. Therefore each solution should be as simple as possible but as complex as necessary.

4 ERROR DEFINITION

Error in the context of this paper means the error between the exact solution of the continuum formulation and the approximated equation and solution using numerical techniques. The general sources of error in computer simulations are: physical modeling errors, discretization and solution errors, programming errors, computer round-off errors [12]. In this paper, we concentrate on the mathematical exercise and assessment of correctness of the code; therefore the physical modeling errors can be left aside. The computer round-off and iterative error shouldn't affect the procedure. For this purpose the solution tolerance should be near to round-off [4] or the error out of it should be at least 100 times smaller then the discretization error [16]. The remaining errors (originating from discretization and of programming) can be assessed with the MMS. To calculate an error of a complete field of a variable in the domain Ω one can use error norms. The continuous L_2 norm for the variable $\mathbf{d}$ compared to an exact solution $\hat{\mathbf{d}}$ can be seen in equation 2. If we assume a discrete solution (e.g. from a Finite Element solution) and an equidistant domain discretization with N elements or nodes one can reformulate equation 2 in the discrete L_2 norm as seen in equation 3. Besides the L_2 norm, the L_1 norm or the $infinity$ norm are often referenced. Especially in the context of Finite Elements the error is often measured in the energy norm [18, 21].

$$E_2 = \| \mathbf{d} - \hat{\mathbf{d}} \|_2 = \sqrt{\frac{1}{\Omega} \int_\Omega \left(\mathbf{d} - \hat{\mathbf{d}} \right)^2 d\omega} \tag{2}$$

$$E_2 = \| \mathbf{d} - \hat{\mathbf{d}} \|_2 = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(d_n - \hat{d}_n \right)^2} \tag{3}$$

5 THE FULLY GEOMETRICAL NONLINEAR MEMBRANE ELEMENT

This paper concentrates on a fully geometrical nonlinear membrane element for large deformations and small strains with a linear elastic isotropic material behavior and plane stress assumption.

5.1 Equilibrium

The equilibrium equations hold in general for the differential/strong form. It states the conservation of momentum in every point of the continuous structure. It can be shown that the equilibrium w.r.t. the current configuration (equation 4) and the initial configuration (eqn. 5) are equivalent [19]. In equations 4 and 5 $\mathbf{d}$ represents the field of displacements, ρ the density, t the time, $\mathbf{f}$ and $\mathbf{F}$ the forces on the current and the initial configuration, respectively. σ represents the Cauchy and $\mathbf{P}$ the first Piola-Kirchhoff (PK1) stress tensor [19]. It can be shown that the presented equilibrium in the strong form can be transformed into the commonly known weak forms of equilibrium (e.g. with the principle of virtual work in eqn. 6) [9, 19]. This equivalence of the weak and the strong form is the most important requirement for the applicability of the MMS. In equation 6, Ω represents

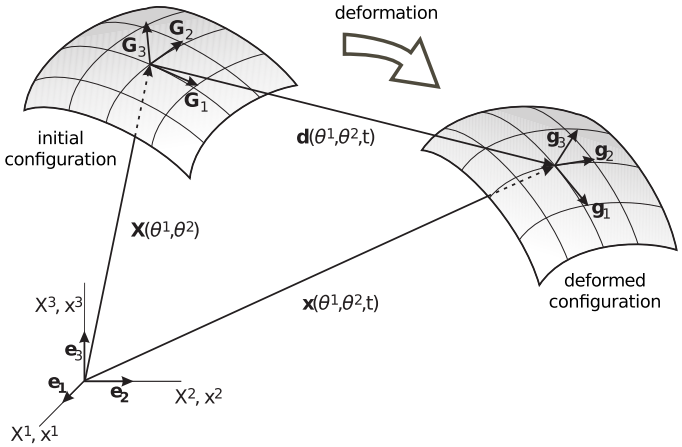

Figure 2: Configurations and deformation process of a body

the initial domain and $\hat{\mathbf{T}}$ the surface forces on the boundary Γ of Ω. $\mathbf{S}$ represents the second Piola-Kirchhoff (PK2) stress and $\mathbf{E}$ the Green-Lagrangian strain tensor. With the goal to perform MMS simulations, all necessary terms of the strong form equilibrium (eqn. 4 resp. 5) have to be determined, completely independent of the implementation of the equilibrium (e.g. eqn. 6).

$$-\rho\frac{\partial^2\mathbf{d}}{\partial t^2} + \nabla\cdot\sigma + \rho\mathbf{f} = 0 \tag{4}$$

$$-\rho\frac{\partial^2\mathbf{d}}{\partial t^2} + \nabla\cdot\mathbf{P} + \rho\mathbf{F} = 0 \tag{5}$$

$$\delta W = -\int_\Omega \rho\frac{\partial^2\mathbf{d}}{\partial t^2}\delta\mathbf{d}d\Omega + \int_\Omega \mathbf{S}:\delta\mathbf{E}d\Omega - \int_\Gamma \hat{\mathbf{T}}\delta\mathbf{d}d\Gamma = 0 \tag{6}$$

5.2 Kinematics

The kinematics of the element (eqn. 7) are shown in figure 2. Capital and lower case letters indicate that quantities belong to the initial (e.g. $\mathbf{X}$) and the current/deformed configuration (e.g. $\mathbf{x}$), respectively. From equation 7 the covariant base vectors in the initial (G_α) and the deformed configuration (g_α) can be derived. θ^α with $\alpha = 1..2$ are the surface parameters along $\mathbf{G}_\alpha$. The base vectors $\mathbf{G}_3$ and $\mathbf{g}_3$ are constructed as normalized cross-product of the first two base vectors. Using the base vectors, the covariant metrics G_{ij} and g_{ij} can be evaluated by $g_{ij} = \mathbf{g}_i \cdot \mathbf{g}_j$ (G_{ij} analogously). The calculation of contravariant base vectors can be performed with the aid of the contravariant metric ($g^{ij} = (g_{ij})^{-1}$) with the rule $\mathbf{g}^i = g^{ij}\mathbf{g}_j$. The deformation gradient tensor $\mathbf{F}$ is calculated in equation 9, where $\otimes$ represents the dyadic product [8,19]. The Green-Lagrangian strain tensor $\mathbf{E}$ is calculated using $\mathbf{F}$ and the unity tensor $\mathbf{I}$ in equation 10.

$$\mathbf{x} = \mathbf{X} + \mathbf{d} = X^i e_i + d^i e_i \tag{7}$$

$$\mathbf{G}_\alpha = \frac{\partial \mathbf{X}}{\partial \theta^\alpha} \qquad \mathbf{g}_\alpha = \frac{\partial \mathbf{x}}{\partial \theta^\alpha} \tag{8}$$

$$\mathbf{F} = \mathbf{g}_i \otimes \mathbf{G}^i \tag{9}$$

$$\mathbf{E} = \frac{1}{2} \cdot \left(\mathbf{F^T F} - \mathbf{I} \right) \tag{10}$$

5.3 Material

The linear elastic isotropic material combined with the plane stress assumptions can be computed with the aid of modified Lamé parameters λ_m and μ_m [11, 20] in equation 11 with 12. Using the material tensor $\mathbf{C}$ and the strain tensor $\mathbf{E}$, the PK2 stress tensor $\mathbf{S}$ can be calculated (equation 13).

$$\mathbf{C} = C^{\alpha\beta\gamma\delta} \mathbf{G}_\alpha \otimes \mathbf{G}_\beta \otimes \mathbf{G}_\gamma \otimes \mathbf{G}_\delta = \lambda_m \cdot G^{\alpha\beta} G^{\gamma\delta} + \mu_m \left(G^{\alpha\gamma} G^{\beta\delta} + G^{\alpha\delta} G^{\beta\gamma} \right) \tag{11}$$

$$\lambda_m = \frac{E \cdot \nu}{(1 - \nu^2)} \qquad \mu_m = \frac{E}{2 \cdot (1 + \nu)} \tag{12}$$

$$\mathbf{S} = \mathbf{C} : \mathbf{E} \tag{13}$$

5.4 Forces

Additionally to the stresses caused by strains, membranes are in general prestressed. The element of interest has its prestress defined in the initial configuration and the prestress tensor $\mathbf{S}_{ps}$ is therefore added to the PK2 stress tensor [11]. As the equilibrum of momentum (eqn. 4 and 5) contains the Cauchy resp. the PK1 stress tensor, they can be evaluated from the present PK2 stress tensor with equations 14 and 15 [8, 19].

$$\sigma = \frac{1}{det(\mathbf{F})} \mathbf{F} \left(\mathbf{S} + \mathbf{S}_{ps} \right) \mathbf{F}^T \tag{14}$$

$$\mathbf{P} = \mathbf{F} \left(\mathbf{S} + \mathbf{S}_{ps} \right) \tag{15}$$

The equilibrium forces required to reach a prescribed deformation $\mathbf{d} = \hat{\mathbf{d}}$ in the context of MMS are shown in equations 16 and 17. For the calculation of the forces it is recommended to use symbolic computation software like Maple®. For ease of understanding the stress tensors are shown as a function of the target displacements $\hat{\mathbf{d}}$ (e.g. $\sigma = \sigma(\hat{\mathbf{d}})$). The general stress tensor components σ^{ij} or P^{ij} can be reduced to the in-plane stresses $n^{\alpha\beta}$ resp. $N^{\alpha\beta}$ in the membrane theory [1]. $\mathbf{n}^\alpha|_\alpha$ represents the covariant derivative of $\mathbf{n}^\alpha$ [1, 19].

$$\hat{\mathbf{f}} = \rho \frac{\partial^2 \hat{\mathbf{d}}}{\partial t^2} - \nabla \cdot \sigma(\hat{\mathbf{d}}) = \rho \frac{\partial^2 \hat{\mathbf{d}}}{\partial t^2} - \mathbf{n}^\alpha|_\alpha = \rho \frac{\partial^2 \hat{\mathbf{d}}}{\partial t^2} - \left(n^{\alpha\beta} \mathbf{g}_\beta \right)|_\alpha \tag{16}$$

$$\hat{\mathbf{F}} = \rho \frac{\partial^2 \hat{\mathbf{d}}}{\partial t^2} - \nabla \cdot \mathbf{P}(\hat{\mathbf{d}}) = \rho \frac{\partial^2 \hat{\mathbf{d}}}{\partial t^2} - \mathbf{N}^\alpha|_\alpha = \rho \frac{\partial^2 \hat{\mathbf{d}}}{\partial t^2} - \left(N^{\alpha\beta} \mathbf{g}_\beta \right)|_\alpha \tag{17}$$

The calculated forces in eqn. 16 or 17 are volume forces. To generate an area force acting on the midplane of the thin membrane they have to be integrated over the thickness b resp. B. It has to be stated again, that the calculated forces ($\hat{\mathbf{f}}$ or $\hat{\mathbf{F}}$) represent the equlibrium forces of the problem for a desired or given displacement field $\hat{\mathbf{d}}$. That means that this force ($\hat{\mathbf{f}}$ or $\hat{\mathbf{F}}$) has to be applied to the code indifferently of the variational method of the implemented equilibrium. All these approaches and their developed equilibria are based on the strong form equilibrium (eqn. 4 resp. 5). Thus, the generality of the method is evident.

5.5 Boundary and Initial Conditions

One can directly determine the Dirichlet and Neumann boundary conditions (BC) for the deformed and the initial configuration with equations 18 resp. 19 [19]. N and n represent the in-plane normal vector, T and t are the traction vectors on the edges of the initial (Γ) and the deformed (γ) configuration,respectively. In steady-state computations the initial conditions (IC) are set to a nonbalanced state (e.g. to zero). In transient problems the IC have to be set to the target value of the variable. At $t = t_0$ the deformed matches the initial configuration (eqn. 20 for both configurations).

$$\mathbf{d}_\gamma = \hat{\mathbf{d}} \qquad \mathbf{t} = \sigma \mathbf{n} \tag{18}$$

$$\mathbf{d}_\Gamma = \hat{\mathbf{d}} \qquad \mathbf{T} = \mathbf{P}\mathbf{N} \tag{19}$$

$$\mathbf{d}(t = t_0) = \hat{\mathbf{d}}(t = t_0) \tag{20}$$

6 FORMAL ORDER OF CONVERGENCE

The discretization method and the chosen form functions for spatial and temporal discretization determine the formal order of convergence [2, 18, 21]. The approximation of the continuum, meaning the geometry, the boundaries, the solution fields and the integrals cause the discretization error in a simulation. One can determine the formal order of convergence p of an approximated expression by comparison with its Taylor series expansion. The first term of the Taylor series which is not approximated is the leading error term within the asymptotic range of the solution. According to [21] the error for an isoparametric polynomial approximation of order j is $\mathcal{O}(h^p) = \mathcal{O}(h^{j+1-m})$. h states the characteristic mesh size and m the magnitude of the mth derivative of the primary variable. For instance, this means for linear form functions ($j = 1$) that the displacement (primary variable, $m = 0$) convergences with a formal order of $p = 2$ while stresses and strains ($m = 1$) converge with a formal order of $p = 1$. The formal order can be reached as long as no other errors (e.g. geometry approximation or integral approximation) with smaller convergence rate occur. A detailed discussion, especially about *Variational Crimes* can be found in [18].

7 PROCEDURE

The procedure applying the MMS is stated in the following. The notation is based on the momentum equilibrium in the initial configuration (cf. eqn. 5, 17, 19 and 20).

1. Invention of a manufactured solution $\hat{\mathbf{d}}$
2. Derivation of the equilibrium forces $\hat{\mathbf{F}}$ using eqn. 17
3. Derivation of the IC (eqn. 19) and BC (eqn. 20)
4. Application of the BC, the IC and the forces $\hat{\mathbf{F}}$ as input of the code to be assessed
5. Performing the simulation with a resulting field $\mathbf{d}$
6. Error evaluation in $\mathbf{d}$ with the aid of $\hat{\mathbf{d}}$ (e.g. eqn. 3)
7. Repetition of steps 4-6 on systematically refined meshes
8. Calculation of the error development with refinement and evaluation of $\hat{p}$ (eqn. 1)
9. Comparison of the formal order of convergence p to $\hat{p}$

There are two different ways to assess the transient functionality of a code: one option is to assess the temporal discretizazion scheme independently of the spatial discretizazion scheme. To do so, one has to isolate the time discretization error from the spatial discretization error. This can be done either with a 0-dimensional problem where spatial discretization doesn't play any role or with a spatial field where the geometry and solution exactely can be represented by the used shape functions. The second option is to assess the temporal and the spatial discretization schemes together. Therefore one has to refine both (time and space) with the same factor $r = r_s = r_t$ in the refinement process. The second procedure is also applicable, if the spatial discretization doesn't match the formal order of the temporal discretization (here, $r_s \neq r_t$) [10]. The error evaluation in step 6 always has to be at the same locations of the mesh and in time (e.g. the midpoints of the coarsest simulation) [10, 12, 13, 15]. This guarantees the comparability of the different resolved solutions.

8 BENCHMARK EXAMPLES

All functions and variables of the following examples are listed in a table, such that the reader is able to construct the force term $\hat{\mathbf{F}}$, the BC and the IC. The choice of the BC type (Dirichlet or Neumann on each edge) is left to the reader. Performing own MMS calculations is therefore possible with the given examples and the procedure of chapter 7. Remember: MMS starts at given equilibrium equations with all its assumptions. It is not necessary that the used parameters are in the range of applicability of the equations. This means that e.g. in the first example the membrane has a thickness B of $0.25m$

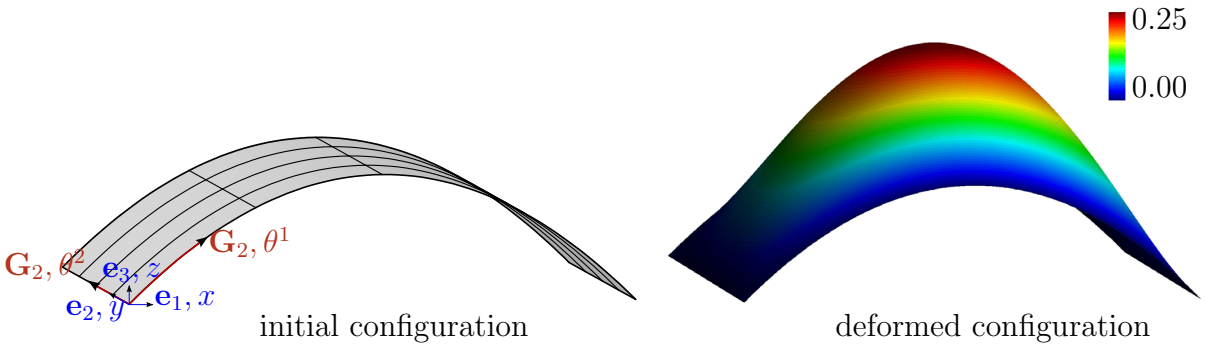

Figure 3: Example 4: Initial (analytically parametrized with parameter lines) and deformed configuration

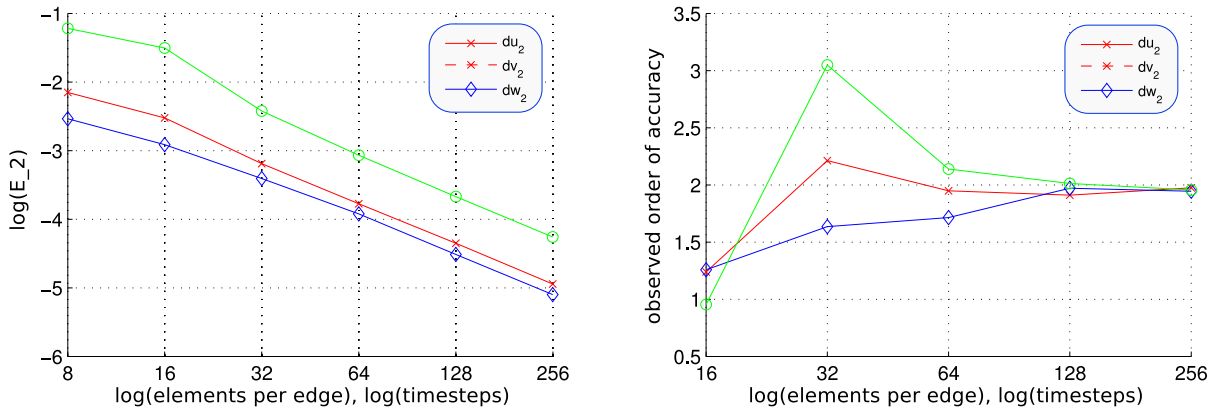

Figure 4: Example 4: Error development and observed order of accuracy $\hat{p}$ over refinement

over an area of only $1m^2$ although this wouldn't make sense in physical applications of the membrane model. For a better understanding of the domain, the parametrization and the geometry are shown for example 4 in figure 3. The logarithmic error and the observed order of accuracy development with refinement for this example are shown in figure 4. The different graphs show the L_2 error norms of the different displacement directions $\mathbf{d} = (du, dv, dw)^T$. The negative inclination - with almost a straight line in the more refined area - of the log-log diagram of figure 4 indicates the error tending against machine accuracy and therefore convergence of the variables. The inclination of the error graph is calculated at each refinement step from a pair of errors (using eqn. 2) and is drawn as a graph in figure 4, right. The calculated inclination states the observed order of accuracy $\hat{p}$. In order to judge an example successful, the observed order of accuracy must reach the formal order of accuracy with refinement.

8.1 Example 1: Plane Structure, In-Plane Deformation steady-state

The simplest example is a plane rectangular membrane under pure tension in a steady-state calculation (cf. table 1). The example assesses the pure normal force action with prestress in the fully geometric nonlinear environment.

Table 1: Overview table of example 1

initial configuration	deformation	material	element properties	domain size
$x = \theta^1$	$d_x = 0.1 \cdot \sin(\theta^1 \pi)$	$E = 70000$	$B = 0.25$	$\theta^1 \in [0..1]$
$y = \theta^2$	$d_y = 0$	$\rho = 0$	$S_{ps_{\theta 1}} = 25000$	$\theta^2 \in [0..1]$
$z = 0$	$d_z = 0$	$\nu = 0$	$S_{ps_{\theta 2}} = 25000$	steady state

8.2 Example 2: Plane Structure, Out-of-Plane Deformation, steady state

Example 2 is a membrane which gets deformed steady-state out-of-plane (cf. table 2). It additionally assesses the geometric transformation through the out-of-plane deformation, shear force action, and the full material law with Poisson's effect.

Table 2: Overview table of example 2

init. conf.	deformation	material	element properties	domain size
$x = \theta^1$	$d_x = 0$	$E = 1000$	$B = 0.001$	$\theta^1 \in [0..1]$
$y = \theta^2$	$d_y = 0$	$\rho = 0$	$S_{ps_{\theta 1}} = 5$	$\theta^2 \in [0..1]$
$z = 0$	$d_z = 0.25 \cdot \sin(\theta^1 \pi) \cdot \sin(\theta^2 \pi)$	$\nu = 0.3$	$S_{ps_{\theta 2}} = 5$	steady state

8.3 Example 3: Plane Structure, Out-of-Plane Deformation, unsteady

Example 3 is a plane membrane which will be deformed out-of-plane in an unsteady calculation (cf. table 3). This example additionally assesses the mass/inertia contribution and the time integration/discretization.

Table 3: Overview table of example 3

init. conf.	deformation	material	element prop.	domain size
$x = \theta^1$	$d_x = 0$	$E = 1000$	$B = 0.001$	$\theta^1 \in [0..1]$
$y = \theta^2$	$d_y = 0$	$\rho = 1000$	$S_{ps_{\theta 1}} = 25$	$\theta^2 \in [0..1]$
$z = 0$	$d_z = 0.25 \sin(\theta^1 \pi) \sin(\theta^2 \pi) \sin(t\pi)$	$\nu = 0.3$	$S_{ps_{\theta 2}} = 25$	$t \in [0..1]$

8.4 Example 4: Curved Structure, Out-of-Plane Deformation, unsteady

The last given example is a plane membrane which will be deformed out-of-plane in an unsteady calculation (cf. table 4). This example additionally assesses the geometric approximation of the initial configuration. The initial and the deformed configuration

of this example are given in figure 3. The error and the observed order of accuracy $\hat{p}$ development are shown in figure 4.

Table 4: Overview table of example 4

init. conf.	deformation	material	element prop.	domain size
$x = \theta^1$	$d_x = 0$	$E = 1000$	$B = 0.001$	$\theta^1 \in [0..1]$
$y = \theta^2$	$d_y = 0$	$\rho = 1000$	$S_{ps_{\theta^1}} = 25$	$\theta^2 \in [0..1]$
$z = \theta^1 - \theta^1\theta^1$	$d_z = \frac{1}{4}\sin(\theta^1\pi)\cos(\theta^2\pi)\sin(\frac{1}{2}\pi t)$	$\nu = 0.3$	$S_{ps_{\theta^2}} = 25$	$t \in [0..2]$

9 CONCLUSIONS

The paper presents an effective method for detailed assessment of all functionalities of a membrane element within steady-state and transient Finite Element codes, containing curved geometries, prestress, and Poisson's effect on curved geometries. Compared to other code verification methods, the MMS represents a very extensive testing method which provides an exact error evaluation. The generality of the MMS method made it very attractive to the authors.The present software code Carat++ could be tested successfully and the formal order of convergence of 2 was confirmed through the performed tests. In addition to the demonstrated application of MMS to structural dynamics with membranes, the methodology can be applied to create benchmarks and to assess other structural dynamics problems, fluid dynamics or even fluid structure interaction software environments, independent of the code structure and the discretization methods.

REFERENCES

[1] Başar, Y. & Krätzig, W. B. *Mechanik der Flaechentragwerke: Theorie, Berechnungsmethoden, Anwendungsbeispiele* Vieweg, 1985

[2] Bathe, K.-J. *Finite element procedures* Prentice-Hall Englewood Cliffs, 1996, 2

[3] Blottner, F. G. *Accurate Navier-Stokes results for the hypersonic flow over a spherical nosetip* Journal of Spacecraft and Rockets, 1990, 27, 113-122

[4] Eça, L. & Hoekstra, M. *Evaluation of numerical error estimation based on grid refinement studies with the method of the manufactured solutions* Computers & Fluids, Elsevier, 2009, 38, 1580-1591

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

[6] Étienne, S. & Garon, A. & Pelletier, D. *Some manufactured solutions for verification of fluid-structure interaction codes* Computers & Structures, Elsevier, 2012, 106-107, 56-67

[7] Fisch, R. & Franke, J. & Wüchner, R. & Bletzinger, K.-U. *Code Verification of OpenFOAM® solvers using the Method of Manufactured Solutions* 7th OpenFOAM Workshop Darmstadt, 2012

[8] Holzapfel, G. A. *Nonlinear solid mechanics: a continuum approach for engineering* John Wiley & Sons Ltd., 2000

[9] Hughes, T. J. *The finite element method: linear static and dynamic finite element analysis* Courier Dover Publications, 2012

[10] Knupp, P. & Salari, K. *Verification of computer codes in computational science and engineering* CRC Press, 2003

[11] Linhard, J. *Numerisch-mechanische Betrachtung des Entwurfsprozesses von Membrantragwerken* Shaker, 2009

[12] Oberkampf, W. & Blottner, F. *Issues in computational fluid dynamics code verification and validation* AIAA journal, 1998, 36, 687-695

[13] Oberkampf, W. & Roy, C. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010

[14] Oberkampf, W. & Trucano, T. *Verification and validation benchmarks* Nuclear Engineering and Design, Elsevier, 2008, 238, 716-743

[15] Roache, P. *Verification and validation in computational science and engineering* Computing in Science Engineering, Hermosa Publishers, 1998, 8-9

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

[17] Salari, K. & Knupp, P. *Code verification by the method of manufactured solutions* Sandia National Labs., Albuquerque, NM (US), 2000

[18] Strang, G. & Fix, G. J. *An analysis of the finite element method* Prentice-Hall Englewood Cliffs, NJ, 1973, 212

[19] Wriggers, P. *Nonlinear finite element methods* Springer, 2008

[20] Wüchner, R. *Mechanik und Numerik der Formfindung und Fluid-Struktur-Interaktion von Membrantragwerken* Shaker, 2007

[21] Zienkiewicz, O. C. & Taylor, R. L. *The finite element method* McGraw-Hill London, 1977, 3