Schriftenreihe des

Lehrstuhls für Statik der
Technischen Universität München

Band 23

Rupert Johannes Fisch

Code Verification of Partitioned FSI Environments for Lightweight Structures

München 2014

Veröffentlicht durch

Kai-Uwe Bletzinger
Lehrstuhl für Statik
Technische Universität München
Arcisstr. 21
80333 München

Telefon: +49(0)89 289 22422
Telefax: +49(0)89 289 22421
E-Mail: kub@tum.de
Internet: www.st.bgu.tum.de

ISBN: 978-3-943683-27-1

TECHNISCHE UNIVERSITÄT MÜNCHEN

Ingenieurfakultät Bau Geo Umwelt

Lehrstuhl für Statik

Code Verification of Partitioned FSI Environments for Lightweight Structures

Rupert Johannes Fisch

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender:

Univ.-Prof. Dr.-Ing. Josef Zimmermann

Prüfer der Dissertation:

1. Univ.-Prof. Dr.-Ing. Kai-Uwe Bletzinger
2. Univ.-Prof. Dr.-Ing. habil. Fabian Duddeck
3. Prof. Dr. Riccardo Rossi, Universitat Politècnica de Catalunya - BarcelonaTech / Spanien

Die Dissertation wurde am 26.06.2014 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 04.11.2014 angenommen.

Abstract

In this thesis, a flexible and modular framework for a stringent Code Verification is developed and applied for partitioned Fluid-Structure Interaction (FSI) environments. The particular application focuses on the simulation of lightweight and elastic membranes in the wind.
In addition to the assessment of the single fields of FSI simulations, the main methodical focus is on the assessment of the Dirichlet-Neumann surface coupling within partitioned FSI analyses using non-matching grids.
This Code Verification assessment focuses on the consistency of the implemented equations and on the convergence of the field variables. In the context of partitioned Fluid-Structure Interaction (FSI), a framework containing analytical benchmark suites with increasing complexity is elaborated which covers all challenging aspects of lightweight surface structures in turbulent flows. Apart from considering the different fields of the structure and the fluid individually, a fully coupled benchmark for FSI is developed.
Whilst the application of the developed framework generates trust in the software, it also reveals deficiencies or even mistakes of the software, thus clearly pointing out the demand for further software development.

The developed framework and the respective benchmark suites are applied to the partitioned FSI environment of the Chair of Structural Analysis of the Technische Universität München. The consistency of this environment and the convergence with at least first order of accuracy in space and time has been confirmed for all parts separately as well as combined. Through this application of the hierarchical benchmark suites the presented framework can be used for further simulations with the long term objective of generating trust in the predictive capability of the numerical wind tunnel environment. Moreover, promising and effective routes of further code development are identified, the treatment of which could significantly improve the quality of the results of further simulations.

It can be concluded that the developed framework is highly qualified for a strict mathematical assessment of engineering software. More importantly, the flexibility and modularity of the developed assessment framework are encouraging for modifications and extensions to other parts of the simulation environment. Consequently, the range of predictive capability can in further research continuously be extended in order to finally obtain reliable results of the numerical wind tunnel.

Zusammenfassung

In der vorliegenden Arbeit wird eine flexible und modulare Umgebung zur stringenten Code Verifikation von Umgebungen zur partitionierten Fluid-Struktur Interaktion (FSI) entwickelt und angewendet. Die spezielle Anwendung konzentriert sich auf die Simulation von leichten und flexiblen Membrantragwerken im Wind.
Das methodische Hauptaugenmerk liegt neben der Bewertung der Einzelfelder von FSI Simulationen auf der Bewertung der Dirichlet-Neumann Oberflächenkopplung mit nicht übereinstimmenden Gittern der beiden Felder innerhalb der partitionierten FSI Analyse.
Die Code-Verifikationsbewertung beschränkt sich hierbei auf die Konsistenz der implementierten Gleichungen sowie die Konvergenz der Feldvariablen. Im Kontext der partitionierten Fluid-Struktur Interaktion (FSI) wird hierzu eine Umgebung entwickelt, die analytische Benchmarks mit ansteigender Komplexität beinhaltet. Diese Benchmarkreihe deckt insbesondere alle speziellen Anforderungen zur Simulation leichter Flächentragwerke in turbulenten Strömungen ab. Hierfür wurde jeweils eine eigene Reihe von Benchmarks für die Struktur und für das Fluid, sowie eine Reihe für die vollständig gekoppelte FSI entwickelt.
Die Anwendung der entwickelten Testumgebung erhöht so die Belastbarkeit der Ergebnisse und führt zu mehr Vertrauen in die bewertete Software. Darüber hinaus kann das vorgestellte Vorgehen Schwächen und Fehler der Software offenlegen und so klare Ansatzpunkte für weitere Entwicklungen aufzeigen.

Die Testumgebung mit ihren Benchmark-Reihen wird hier auf die vorhandene partitionierte FSI Umgebung des Lehrstuhls für Statik der TU München angewandt. Die Konsistenz dieser Berechnungsumgebung und die Konvergenz mit mindestens erster Ordnung in Raum und Zeit werden hierbei sowohl für die einzelnen Bestandteile als auch für die gesamte Berechnungsumgebung bestätigt. Nach der Anwendung der hierarchischen Benchmarkreihen kann die vorgestellte Umgebung für weitere Untersuchungen verwendet werden. Diese Untersuchungen sollen als Fernziel dem Aufbau größeren Vertrauens in die prädiktiven Möglichkeiten des numerischen Windkanals dienen. Ferner werden hierbei vielversprechende und effektive Ansätze zur Weiterentwicklung der Software aufgezeigt, deren Ausarbeitung zu einer signifikanten Verbesserung der Qualität künftiger Simulationsergebnisse beitragen können.

Abschließend bleibt festzuhalten, dass die entwickelte Umgebung für die stringente mathematische Bewertung gekoppelter Software im Ingenieurwesen sehr gut geeignet ist.
Darüber hinaus empfiehlt sich die Modularität und Flexibilität der vorgestellten Bewertungsumgebung für Modifikationen, um die Bewertung auf andere Anwendungen bzw. Simulationsprogramme auszudehnen. Infolgedessen kann durch weitere Forschung der Geltungsbereich der abgesicherten Vorhersagefähigkeit stetig erweitert werden, um letzten Endes verlässliche Ergebnisse des numerischen Windkanals zu erhalten.

Acknowledgments

This dissertation was written from 2009 to 2014 during my time as research scholar and teaching assistant at the Chair of Structural Analysis (Lehrstuhl für Statik) at the Technische Universität München, Munich, Germany.

I would like to thank Prof. Dr.-Ing. Kai-Uwe Bletzinger, who acted as my supervisor and first examiner. His faith in my commitment and capabilities enabled the possibility to persue new scientific ways for the chair. Thanks to fruitful discussions with him, Dr.-Ing. Roland Wüchner, and Dr.-Ing. Jörg Franke, my work could mature to this thesis.

Furthermore, I would like to address my thanks to the members of my examining jury, Univ.-Prof. Dr.-Ing. habil. Fabian Duddeck and Prof. Dr. Riccardo Rossi. Their interest in my work is gratefully appreciated. Also, I want to thank Univ.-Prof. Dr.-Ing. Josef Zimmermann for chairing the jury.

Working at the Chair of Structural Analysis, I always appreciated the atmosphere among the colleagues. I will always warmly recall the close cooperations, long discussions, and leisure activities. Especially I would like to thank Benedikt Philipp, Falko Dieringer, Armin Widhammer, and Stefan Sicklinger for fruitful discussions, their skilled support, and their encouragement.

I am especially grateful towards my family, in particular to my parents Rupert and Anna-Maria, and my wife Katrin. Their enduring support and love provided the basis for my education, my study, and my work.

Munich, November 2014
Rupert Johannes Fisch

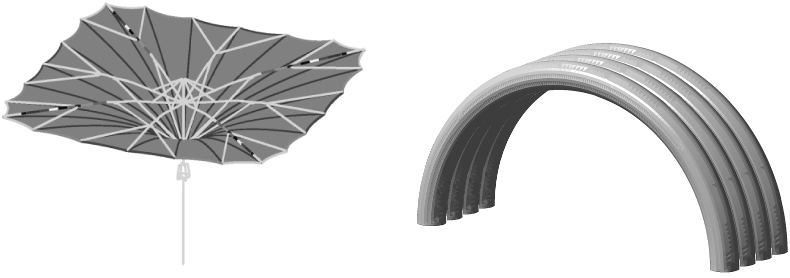
Contents

CHAPTER 1

Introduction

1.1 General

This work addresses the analysis of lightweight elastic structures, sensitive and flexible in the atmospheric wind. The structures in focus for example are elastic membrane tubes or umbrella structures in wind (cf. figure 1.1). The assessment of structures in wind, today and in the past, normally is performed using an experimental wind tunnel analysis beside expert judgment. The limitations of the experimental wind tunnel and the increase in computational power lead engineers more and more towards computer aided approaches in wind engineering. Thus, this results in the development of numerical wind tunnels as complements to the experimental wind tunnels.



(a) A lightweight umbrella structure from a collaboration with SL Rasch, http://www.sl-rasch.de

(b) A prototype of an inflatable four-tubes structure, developed in the ULITES project, http://www.cimne.com/websasp/ulites

Figure 1.1: Examples for lightweight membrane structures sensitive to wind

The design of numerical wind tunnels, encompasses plenty of aspects, such as turbulence, modeling of the atmospheric boundary layer, creating geometric models, solving the problem with all relevant influencing physical factors, et cetera.

Beside these topics, which are complex scientific research areas themselves, the justified skepticism in new approaches has been and still is very high. This can be seen analogously to the early stages of computational structural mechanics using the Finite Element method [61]. The early developments in the Finite Element method in the forties and fifties of the 20th century by Turner, Argyris et al. [3, 116] were also seen very critically and were refused as nonsense from the scientific societies [61]. It took more than 20 years that the Finite Element method found its way into the structural mechanics application in civil engineering (e.g., [56]). One famous breakthrough point of the Finite Element application in civil engineering was the construction of the Olympic stadium 1971 in Munich, Germany. In the eighties, a further development in shell structures et cetera completed the early developments in Finite Elements for civil engineering application [126].
This long term development, from the pioneering publications until the further development and rather the acceptance of Finite Elements as an aiding approach for the design and construction of structures, gives confidence in the promising, but also enduring, developments of numerical wind tunnels [8]. The parallels to Finite Elements rather demonstrate that there is the time, but also the strong requirement for sustainable and high qualitative work to create confidence in software [8, 10, 17, 90, 103, 105].

1.2 Target of this Work

In terms of engineering analysis, the desired simulations deal with Fluid Structure Interaction (FSI) problems. FSI consists of a fluid contribution, a structural contribution, and a coupling part at the common interface between the fluid and the structure. The structural and the fluid part is computed by a Computational Structural Dynamics (CSD) resp. a Computational Fluid Dynamics (CFD) software (cf. figure 1.2).

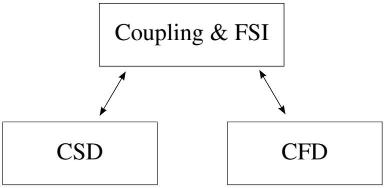


Figure 1.2: Overview of a partitioned FSI environment

This work presents a Code Verification framework for the assessment of FSI environments, to be able to assess the correctness of the implementations of all parts of the environment both individually and jointly. As partitioned FSI environments may consist of different software, possibly with different discretization methods, a special focus is on the topic of non-matching grids at the common interface. Thus it is aimed at increasing the confidence in the predictive capability of the software environment. Furthermore, the presented method gives a basic setup for the reader to adapt or to customize the framework for his particular software environment.

The individual numerical errors arising in the named FSI analyses are identified, shown, and assessed using the Method of Manufactured Solutions (MMS) as a method for Code Verification [73]. Therefore, a stairway of benchmarks with increasing complexity is developed to assess the complete FSI environment altogether, but also all individual parts independently. On the one hand, this stairway of benchmarks compose a framework to evaluate the individual accuracy and to generate trust in the applicability of the environment on the application of interest. On the other hand, the modularity and flexibility of the framework makes adaptations and further developments of Verification tasks possible (cf. sections 3.9, 4.13 and 5.9).

The individual results of the assessment clearly point out the needs and, very important, the most efficient locations of further software development. Furthermore, a successful assessment can be used as a certificate of confidence in the software environment to finally produce predictive results.

1.3 Restrictions for the Application

As it is not possible to completely test and assess all parts of the code and code combinations [73], a target application has to be defined. This definition, or rather restriction, defines the necessary functionality to be tested in a benchmark sequence in the following chapters. The restrictions concerning the FSI functionality due to the available FSI environment are depicted in the following list:

- Partitioned FSI
- Dirichlet-Neumann partitioning using force elimination
- Gauss-Seidel communication

Besides the restrictions due to the environment, further restrictions are made for the assessment framework in the following list:

- Structured grids using quadrilateral and hexahedral volumes
- Incompressible Newtonian fluids using the Navier-Stokes equations
- Membrane structures
- Isotropic St. Venant-Kirchhoff structural material
- Negligence of structural damping

1.4 Requirements

After selecting the target application and carefully defining the restrictions of the physics and the software features, the requirements for the software components can be defined. The precise determination of requirements similarly defines the demand of components to be assessed in the order of accuracy tests in the following chapters. The following sections define the requirements of the three contributing engineering software of the FSI process.

1.4.1 Coupling Interface

As shortly mentioned in section 1.3, a Dirichlet-Neumann decomposition is chosen for the partitioning of the FSI process [117]. This means for the specific case of the chosen environment, that, on the one hand, the displacements (Dirichlet) and, on the other hand, the traction forces (Neumann) at the common interface are used as coupling fields. Therefore, the requirements for the interface is the functionality of surface transfer of geometry and the sampling and mapping of forces and displacements in a conservative respectively consistent way.

1.4.2 Computational Structural Dynamics

The selected target applications and the named restrictions define the Computational Structural Dynamics (CSD) functionality. The requirement therefore defines a prestressed membrane structure, using the linear St. Venant-Kirchhoff material law with the plane-stress assumption. As the structure will behave in an unsteady manner, beside the inner forces out of stresses, inertial/mass forces are required. As already mentioned, structural damping is neglected due to the fact that fluid damping (added mass effect) is assumed to be much larger as structural damping of membranes.

1.4.3 Computational Fluid Dynamics

The selected target applications and the named restrictions define the Computational Fluid Dynamics (CFD) functionality. Dealing with a wind regime, the fluid flow is described by the incompressible Navier-Stokes equations in an unsteady regime. These contain convection, diffusion, inertia effects, and possibly turbulence. Additionally, the fluid boundaries respectively a body in the fluid, must be able to move.

1.5 The Partitioned FSI Environment

The software environment used in this work is a partitioned FSI environment. Partitioned means that the equations of the fluid, the structure, and their coupling are solved in a segregated and staggered way. In the present case, the segregation of the equations is additionally a segregation of software (cf. figure 1.2). The fluid part of the FSI is performed using the CFD software OpenFOAM® (Version 2.1.x from 03/01/2014). It is an open source project software using the Finite Volume discretization. The structural part of the FSI is performed using the CSD software

Carat++ (version from 03/01/2014). Carat++ is an in-house developed code of the Chair of Structural Analysis at the Technische Universität München, using the Finite Element discretization. The FSI environment is set with the software EMPIRE (version from 03/01/2014), and the coupling procedures are provided using the software Emperor (version from 03/01/2014). Both software are provided from http://www.empire-multiphysics.com/. The respective EMPIRE_API is an add-on to each software to guarantee the communication between the single software components in the EMPIRE environment. A software overview of the FSI environment is shown in figure 1.3.

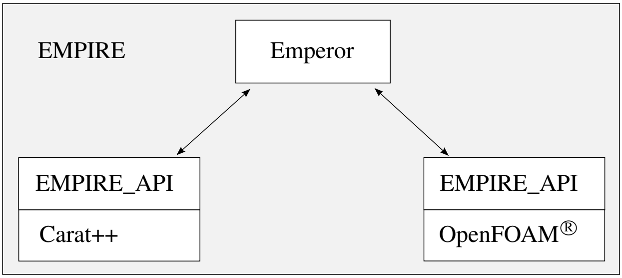


Figure 1.3: Overview of the FSI environment of EMPIRE

1.6 Organization of the Thesis

The introduction in chapter 1 provides a short overview of the topic and the subject focus on Verification of FSI environments for wind engineering application.
Chapter 2 gives an overview of predictive capability of software and the placement of Code Verification tasks. The particular focus lies on the application of the Method of Manufactured Solutions (MMS) as an order of accuracy assessment.
The application of the developed MMS framework on a particular partitioned FSI environment directly provides the arrangement of the subsequent chapters. The structural dynamics are presented in chapter 3, the fluid dynamics are presented in chapter 4, and the coupling and the FSI processes are given in chapter 5.
The mentioned chapters 3-5 have the same setup. The chapter starts with assumptions and restrictions for the intended use of the software, which will be assessed. As it is in general not possible to completely assess a software, the precise definition of the application of interest constitutes a key feature of software assessment. The governing equations in its differential and integral form as well as the chosen discretization and solution method are shown. The formal orders of accuracy are derived from the knowledge of the implemented discretization schemes. Afterwards, the framework of the generalized MMS application is elaborated, and the particular MMS framework is presented. Finally, application examples arranged in an hierarchical stairway of complexity to assess the FSI environment step by step are outlined. These examples are designed and presented in detail in order to serve as benchmarks for the reader. A conclusion of the insights and proposals for potential

adaptations of the MMS framework to encourage the reader to modify and extend the presented framework completes each chapter.

Chapter 6 draws an overall conclusion and an outlook on possible further developments of the framework and the general application of Verification and Validation tasks to generate and to increase the confidence and reliability in the numerical wind tunnels.

CHAPTER **2**

The Role of Code Verification

Code Verification, as a part of Verification and Validation (V&V), contributes to the assessment of predictive capability and reliability of software for modeling and simulation (M&S) [76]. As the terminology in the literature is not unified, all necessary term definitions are given and referenced in the following sections. The work at hand follows the definitions in [73, 76, 79, 91, 100].

2.1 Placement in the Frame of Predictive Capability of Software

This work follows the definition of prediction as the 'use of a [...] model to foretell the state of a physical system under conditions for which the [...] model has not been validated' [76]. This means, the specific simulations of interest are different from the cases that have been validated. Therefore [76] states that 'it is a prediction, not a postdiction'.

2.2 Uncertainty and Error

In the literature the terms uncertainty and error are often intermixed. To evaluate the credibility of software, both have to be differentiated carefully. A good overview is presented in [2, 33, 76, 79, 94, 114]. This work follows the definition of the terms given in [76]:

- Uncertainty: A potential deficiency in any phase of activity of the modeling process that is due to lack of knowledge.
- Error: A recognizable deficiency in any phase or activity of modeling and simulation that is not due to lack of knowledge.

In general, sensitivity analyses (so-called what-if analyses) and uncertainty analyses (using probabilistic methods) are used to evaluate uncertainty and the quality of the physical model (e.g., [21, 51, 73, 95]). Furthermore, it consists of unacknowledged

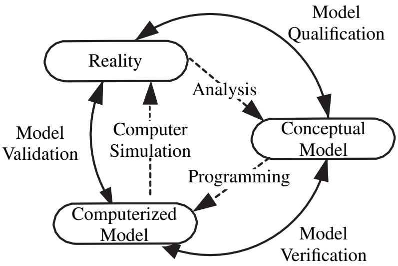


Figure 2.1: Phases of modeling and simulation and the role of V&V [100]

errors like mistakes of the analyst, programming blunders, and compiler errors. There are no straightforward methods for their estimation [78].
The acknowledged error is characterized by knowledge of the divergence from an approach that is considered to be a baseline for accuracy [78]. Acknowledged errors are the finite precision of computers ('round-off'), the transfer of a set of governing equations to an algebraic system of equations ('discretization'), and the incomplete iterative convergence error (IICE) solving the equations iteratively [57, 73, 75].

2.3 Verification & Validation

Oberkampf and Roy present the fundamental elements for the credibility in computational results [73]. Among these, Verification and Validation (V&V) of computational simulations provide evidence for the correctness of the code as well as the results and contain a concept of quantitative accuracy assessment. The necessity of V&V efforts is shown very impressively in Hatton's T-experiments [42], where Hatton assessed software codes, which were thought to work as intended. He analyzed in total 1.7 million lines in Fortran and 1.4 million lines in C [79]. Hatton concluded, 'The T-experiments suggest that the results of scientific calculations carried out by many software packages should be treated with the same measure of disbelief researchers have traditionally attached to the results of unconfirmed physical experiments' [42]. Reasons for the magnitude of errors in the codes are discussed in [79].

Figure 2.1 illustrates the role of V&V in the context of modeling and simulation. Verification means the "process of assessing software correctness and numerical accuracy of the solution to a given mathematical model" [73], or simpler spoken "Verification is about solving the equations right" [91] (cf. figure 2.2). Validation means the "process of assessing the physical accuracy of the mathematical model based on comparison between computational results and experimental data" [73], or simpler spoken "Validation is about solving the right equations" [91] (cf. figure 2.3). A huge effort is taken in the serious Validation of software simulations in comparison with real scale and wind tunnel measurements [1, 66, 73–75, 77–79, 84, 112, 113, 115]. As Validation is not the core of this work, the detailed overview of recent Validation efforts in the work of Blocken is denoted as a reference [8].

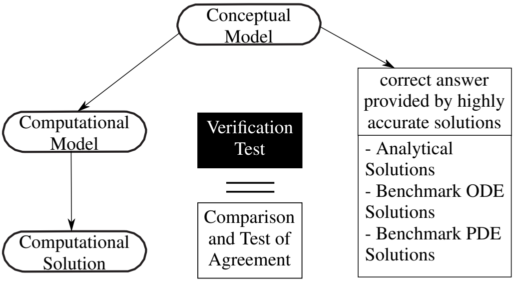


Figure 2.2: Code Verification process [76]

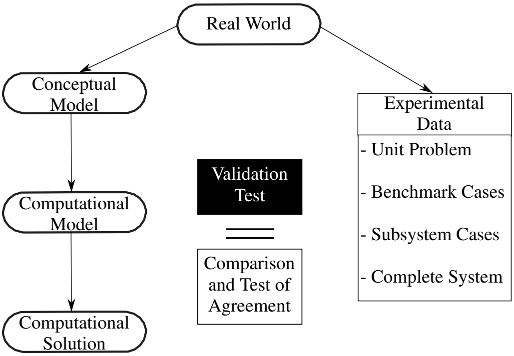


Figure 2.3: Validation process [76]

Verification splits in Code Verification and Solution Verification [73]:

- Code Verification: The process of determining that the numerical algorithms are correctly implemented in the computer code and of identifying errors in the software.
- Solution Verification: The process of determining the correctness of the input data, the numerical accuracy of the solution obtained, and the correctness of the output data for a particular simulation.

2.3.1 Solution Verification

Following [73], Solution Verification addresses the estimation of the magnitude of errors originating from human mistakes in preparation of input data for a simulation, numerical errors due to the execution of the simulation, and of human errors in post-processing of the output data of the simulation. Compared to Code Verification, the simulation in Solution Verification is a physical realistic case. A detailed discussion of Solution Verification and the approaches to estimate the errors can be found in [9, 11–13, 25, 41, 52, 53, 73, 77, 78, 88, 89, 92, 93, 96, 97].

2.3.2 Code Verification

In general, a conservation law defines the (physical) base for engineering modeling and simulation software. The conservation law is formulated in integral, in ordinary differential or in partial differential equations. These equations are called the interior equations. The basic form of the equations constitute the conceptual model of a problem (cf. figure 2.1). Additionally, auxiliary conditions or equations are set. On the one hand, different assumptions are made for the equations, e.g., isotropy of the material, constant density, et cetera. On the other hand, the definition of a solution domain, the boundary conditions, and the initial conditions are set as auxiliary conditions. The interior and the auxiliary equations together form the so-called governing equations. Code Verification assesses whether the governing equations are solved in a consistent way by the software implementation [98].

Numerical schemes are named consistent, if the discretized equations approach the actual governing equations in the limit as the discretization parameters ($\Delta\mathbf{x}, \Delta t$) approach zero [73].
Beside consistency, the convergence needs to be defined. Convergence addresses, whether the exact solution of the discrete equations approach the exact solution of the governing equations in the limit as the discretization parameters ($\Delta\mathbf{x}, \Delta t$) approach zero [73]. Simply spoken, consistency deals with the representation of the equations, convergence deals with representation of the solution.

Discretization is the transfer of the governing equations to an algebraic equation system [75, 78, 98]. Discretization of the governing equations divides the continuous (infinite) domain to a finite number of elements or volumes, where the solution is approximated. The precise solution of the approximated (discretized) governing equations is called the discrete solution. The difference between the discrete and the exact solution, is called the discretization error [98] (cf. figure 2.4). Discretization methods are called consistent, if the error tends to zero as the number of elements respectively volumes increase towards infinity. A more detailed description of the prerequisites of discrete methods, especially the topics of consistency, stability, and convergence can be found in [73].

Code Verification efforts consist, on the one hand, of Software Quality Assurance (SQA) efforts and, on the other hand, of Numerical Algorithm Verification [52, 73, 75, 93]. Numerical Algorithm Verification provides procedures to identify and to quantify the five predominant sources of error in simulations:

- Insufficient spatial convergence (spatial discretization error)
- Insufficient time convergence (time discretization error)
- Incomplete iterative convergence error (IICE)
- Computer round-off error

- Programming mistakes

Assuming a code without programming mistakes, the relationship between numerical error and discretization error is shown in figure 2.4. The goal of Code Verification efforts is "to distinguish errors in mathematical modeling accuracy as clearly as possible from other errors" [78]. In other words, the discretization error is tried to be isolated from the numerical error (cf. figure 2.4).

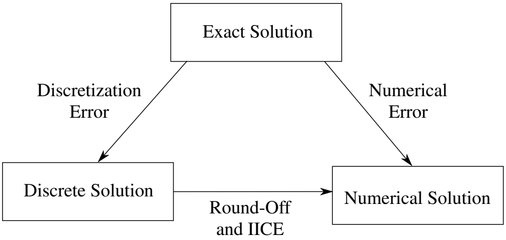


Figure 2.4: Relationship between numerical error and discretization error [57]

Salari and Knupp give an overview of commonly used dynamic testing approaches and their acceptance criteria [98]. Among others, the paper gives a good overview about the applicability and value of Trend Tests, Symmetry Tests, Code-to-Code Comparisons, the Method of Exact Solutions (MES) and the Method of Manufactured Solutions (MMS). A detailed discussion and extensive description of different methods for Code Verification methodologies are given in [73, 91, 98].
Both, the MES and the MMS are based on the concept of dealing with analytical solutions [76]. Both of them are preferred approaches and generate the most confidence in a code's correctness, as they are able to assess the consistency and the order of accuracy of a code; they are therefore called order of accuracy tests. The order of accuracy defines the rate of the solution's improvement during refinement. MES deals with published or self-created (exact) analytical solutions of physical problems (e.g., [108]). The limitation of using the MES is that the code application of interest usually is far more complex than the available solutions of MES.

2.3.3 The Method of Manufactured Solutions

The Method of Manufactured Solutions (MMS) was invented by Steinberg and Roache [104]. The idea of the MMS is similar to the MES. More precisely, the MMS also deals with analytical solutions. However, the MMS is far more flexible and able to assess the equations implemented in the software code in its entirety. The reverse of the medal is that this flexibility in general requires access to the algebraic equations or even the complete code. The MMS has already been used successfully in fluid dynamics, e.g., in [22, 24, 32, 57, 73, 93, 97, 98, 119], but also in structural dynamics [31, 58] or recently in monolithic fluid structure interaction computations [26].
In the work at hand, a framework and benchmarks are developed to assess a partitioned fluid structure interaction environment using the MMS.

2.3.3.1 Idea of the MMS

The MMS can be seen and used as a toolbox to assess implemented numerical schemes and procedures for the solution of conservation equations, e.g., as a partial differential equation (PDE). The principle idea of the MMS is to build own analytical solutions; therefore, they are called manufactured solutions. The manufactured solution in general is not fulfilling the actual PDE. Thus additional source respectively force terms must be added to the algebraic/discretized governing equations. Hence the program is able to compute the manufactured solution asymptotically. The addition of the source respectively force is the reason for the need of accessibility to the algebraic equations in the software as mentioned above. With this concept, all parts of the governing equations can be assessed with a variety of manufactured test cases as benchmarks. Arranging these benchmarks in a hierarchical manner, an obstacle course is built. Successful passing the course means that the mentioned observed order of accuracy matches the formal order of accuracy. In the case of passing, the tested schemes can be used for further and more complex investigations. If benchmarks are not passed, the hierarchical order of the named obstacle course clearly helps to identify inconsistencies or even mistakes in the code [57, 98].

2.3.3.2 Requirements to the Manufactured Solution

As the manufactured solution of the governing equations should rigorously assess the computer code, there are a few requirements to the solution [98]. The major parts are given in the following list. The solution of the governing equations should...

- be composed of smooth and non-trivial functions
- be generally enough and well balanced to exercise every term in the governing equations
- have a sufficient number of non-trivial derivatives
- not contain singularities
- be in a realistic range of the designed application, e.g., no negative density
- be defined on a connected subset of the dimensions in space

It is stated once again, there is no requirement that the manufactured solution is physically a realistic one. Simply spoken, the code is not able to know, whether the problem is physically realistic or non-physical. The code is only solving a set of algebraic equations [98].

2.3.3.3 General Procedure of the MMS

The developed manufactured solution field $\hat{\mathsf{b}}$ represents an analytical solution for the primary variable field b of a PDE in the continuum. The hat above a variable is always used here for indicating a manufactured solution. After insertion of the field solution $\hat{\mathsf{b}}$ into the differential equations, a source respectively force term remains as the equilibrium no longer holds for an arbitrary solution $\hat{\mathbf{b}}$. The source/force term can be obtained by hand or by using symbolic manipulation software, such as Maple® or Mathematica®. If the numerical schemes are implemented consistently, it can be observed that $\mathbf{b}$ tends towards $\hat{\mathbf{b}}$ for systematically refined calculations. During the assessment, it is aimed at determining and observing the development of the discretization error along with systematic grid and time refinement. However, the numerical solution is additionally equipped with round-off errors and IICE (cf. figure 2.4). The round-off error is generally very small compared to the discretization error, if double machine accuracy is used. To keep the IICE small, the solution of the equation systems should be iterated almost until machine accuracy or at least to a certain factor smaller then the discretization error [57, 75]. Considering the named two points, the numerical solution nearly matches the discrete solution (cf. figure 2.4). Therefore the discretization error can be evaluated by comparison of $\mathbf{b}$ and $\hat{\mathbf{b}}$. If all numerical schemes are implemented correctly, it can be observed that the solution $\mathbf{b}$ tends towards the manufactured solution $\hat{\mathbf{b}}$ for systematically refined calculations.

As $\hat{\mathbf{b}}$ constitutes the exact reference solution, the difference between $\mathbf{b}$ and $\hat{\mathbf{b}}$ represents the exact numerical error of the individual calculation. The error of a discrete solution can be determined with formula 2.1:

$$E_h = \| \mathbf{b} - \hat{\mathbf{b}} \| = c \cdot h^p \tag{2.1}$$

In equation 2.1, E_h is the error of a discrete simulation in a chosen error norm (cf. section 2.3.3.7). c is a constant, h is a characteristic element or time step size, and p is the observed order of accuracy. Equation 2.1 constitutes that the discrete solution error reduces with the factor r^p (with $r = \frac{h_{coarse}}{h_{fine}}$) during refinement with a refinement factor r. Additionally, using equation 2.1, the observed order of accuracy between two different refined simulations can be derived with equation 2.2 [57]:

$$p = \frac{\log\left(\frac{E_{h_{coarse}}}{E_{h_{fine}}}\right)}{\log(r)} \tag{2.2}$$

If the formal order of accuracy $\hat{p}$ matches p in the asymptotic range of the solution, the following parts of the code have been verified [91]: all coordinate transformations, the order and the programming of the discretization, and the solution procedure of the algebraic equation system. If the two orders p and $\hat{p}$ do not match, there can be many reasons, e.g., programming mistakes, insufficient grid resolution, singularities, et cetera. An extensive discussion of reasons is given in [73]. The designed MMS procedure with all its parts is shown in detail in the following chapters.

The independence of the numerical approach turned out as a major advantage of the MMS providing analytical benchmark solutions: The governing equations are handled in the continuum; therefore, it is completely independent of the discretization method (e.g., Finite Elements, Finite Volume,...), the solution procedure (direct solution, fix-point iteration, Newton-Raphson,...), or if the equations are linear or non-linear. This point makes it very attractive for the integral assessment of the existing partitioned FSI environment, which consists of different programs using different discretization methods.

The outlined manufactured solutions provide a hierarchical sequence of complexity or even a benchmark series to assess all constituent parts of an FSI process step by step as elaborated in the following chapters. The mentioned hierarchical order can be very helpful to localize coding or other mistakes [57, 98]. Therefore, each solution should be as simple as possible, but as complex as necessary.

In the work at hand, the MMS is applied to assess the Finite Element formulation of a fully geometric nonlinear membrane element in an unsteady regime, fully coupled with the Finite Volume formulation of unsteady incompressible fluid dynamics.

2.3.3.4 Initial and Boundary Conditions

Every solution of a PDE needs to have auxiliary conditions. If the PDE is space-dependent, boundary conditions are necessary. If the PDE is (additionally) time-dependent, initial condition are (additionally) necessary. For the MMS application, the boundary and initial conditions can directly be derived from the solution $\hat{\mathbf{b}}(\mathbf{x}, t)$ [57]. The vector $\mathbf{x}$ denotes the position in space and t denotes the time. Purely space-dependent problems are from now on called steady state problems, and space-time-dependent problems are called unsteady problems.

For the assessment of the steady state solution of a code, the chosen initial conditions as a starting point of the (iterative) solution must be significantly different to $\hat{\mathbf{b}}(\mathbf{x}, t = 0)$. If the chosen initial condition matches the analytical solution, the ability of the code to solve steady state problems will not be approved. Pragmatically, the initial condition at the field is very often set to a zero value field.

For the assessment of the unsteady solution of a code, the initial condition must match the analytical solution at the initial time step ($\mathbf{b}(\mathbf{x}, t = 0) = \hat{\mathbf{b}}(\mathbf{x}, t = 0)$) to guarantee that no artificial error is introduced.

The Dirichlet and Neumann boundary conditions at the boundary γ at the boundary position $\mathbf{x}^\gamma$ with the boundary normal $\mathbf{n}^\gamma$ can directly be calculated respectively derived from the manufactured solution field $\hat{\mathbf{b}}(\mathbf{x}, t)$. The boundary field of a field $\mathbf{b}$ is abbreviated by:

$$\mathbf{b}^\gamma = \mathbf{b}(\mathbf{x} = \mathbf{x}^\gamma, t) \tag{2.3}$$

Using equation 2.3, the Dirichlet and Neumann boundary conditions are set in equations 2.4 and 2.5:

$$\mathbf{b}^\gamma = \hat{\mathbf{b}}^\gamma = \hat{\mathbf{b}}(\mathbf{x} = \mathbf{x}^\gamma, t) \tag{2.4}$$

$$\frac{\partial \mathbf{b}^\gamma}{\partial \mathbf{n}^\gamma}(^*) = \frac{\partial \hat{\mathbf{b}}^\gamma}{\partial \hat{\mathbf{n}}^\gamma}(^*) = \frac{\partial \hat{\mathbf{b}}(\mathbf{x} = \mathbf{x}^\gamma, t)}{\partial \hat{\mathbf{n}}(\mathbf{x} = \mathbf{x}^\gamma, t)}(^*) \tag{2.5}$$

In the context of this work, no other boundary conditions with possibly higher derivatives are used.

2.3.3.5 Assessment of Steady State Simulations

As generally described in section 2.3.3.3, the discretization error of the primary variable $\mathbf{b}(\mathbf{x}, t)$ compared to the manufactured solution $\hat{\mathbf{b}}(\mathbf{x}, t)$ has to be observed.

At steady state simulations, the error of the converged solution is evaluated and observed with mesh refinement to calculate the observed order of accuracy p using equation 2.2. To make the results of the different refined grid simulations comparable, the error at the same sampling points independent of the grid resolution have to be evaluated. As an example, these sampling points could correspond to the available grid points of the coarsest grid used in the study. A detailed explenation and examples can be found in chapters 3 and 4.

2.3.3.6 Assessment of Unsteady Simulations

Similar to the steady state simulations, the error is evaluated always at the same sampling points over time, independent of the time resolution of the particular simulation, to make unsteady simulations comparable. The demand of the sampling points in space correspond to that for the steady state simulations.

There are two different ways to assess an unsteady environment. On the one hand, the time and spatial accuracy are assessed separately. On the other hand, the spatial and time accuracy are assessed at the same time.

To assess the spatial accuracy separately, the manufactured solution of a steady state problem is used. As the time contribution to the PDE vanishes, the spatial terms are isolated.

To assess the time accuracy separately, the spatial contribution should vanish. This is possible in two different ways. The first way is to choose an arbitrary manufactured solution covering all terms of the PDE and to use a superfine grid in all time refinement steps. The assumption herein is that the spatial discretization error of the superfine grid is much smaller than the time discretization error, which should be assessed. However, this assumption is a priori very difficult to estimate and therefore cannot recommended without reservation. The second way is to choose a manufactured solution, which fulfills the solution requirements (cf. section 2.3.3.2) only over time. The solution should therefore be mesh independent. This means, the spatial terms of the PDE herein are, aligned to the spatial discretization, vanishing. For example, if the code uses linear form functions for the spatial discretization, the spatial variation of the manufactured solution should be constant or linear such that the code is able to exactly represent the solution independently of the grid resolution.

Alternatively, the spatial and time accuracy altogether can be assessed jointly. This has the advantage that time-space correlated errors can additionally be assessed

[73]. For this assessment, the manufactured solution is fulfilling the requirements for all terms of the PDE in space and time and the error evaluation is again performed at the same sampling points. If the formal time and the formal spatial orders of accuracy match ($\hat{p} = \hat{p}_t = \hat{p}_s$), the spatial and time refinement factors must match as well ($r = r_t = r_s$). If the formal time and spatial orders of accuracy do not match, the refinement factors must be adapted to really assess both accuracies. This means for example, if a formally second order accurate spatial discretization and a formally first order accurate time discretization are used, the refinement factor over time has to be the square of the spatial refinement factor ($r_t = r_s^2$). As discussed later more in detail, if the grid points in all spatial directions are doubled, the time steps in every refinement step have to be increased by a factor of four [57] as $\hat{p}_t = 1$ and $\hat{p}_s = 2$. If therein an order of accuracy of $p = 2$ is observed, the spatial observed order of accuracy is determined to be $p_s = 2$ and the time observed order of accuracy is determined to be $p_t = 1$ [57].

2.3.3.7 Evaluation of the Global Error

As already mentioned above, the general and omnipresent error sources in computer simulations are physical modeling errors, discretization and incomplete iterative convergence error (IICE), programming mistakes, and computer round-off errors [22, 57, 73, 75]. As the round-off and the IICE are kept very low as described above, the leading error is the discretization error as long as there are no programming mistakes.

Error norms can be used to determine the global error of a field $\mathbf{b}$ in its spatial domain K. An E_2 norm is chosen to have a representative mean field error of the complete domain. The continuous E_2 error norm for the variable $\mathbf{b}$ compared to the exact solution $\hat{\mathbf{b}}$ in the domain K can be seen in equation 2.6. If we assume a discrete solution and an equidistant domain discretization with N elements or nodal results, the discrete E_2 norm of $\mathbf{b}$ compared to the exact solution $\hat{\mathbf{b}}$ can be determined as shown in equation 2.7 [22, 57, 73].

$$E_2 = \parallel \mathbf{b} - \hat{\mathbf{b}} \parallel_2 = \sqrt{\frac{1}{K} \int_K \left(\mathbf{b} - \hat{\mathbf{b}}\right)^2 dK} \tag{2.6}$$

$$E_2 = \parallel \mathbf{b} - \hat{\mathbf{b}} \parallel_2 = \sqrt{\frac{1}{N} \sum_{n=1}^{N} \left(b_n - \hat{b}_n\right)^2} \tag{2.7}$$

Beside the given E_2 norm, a non-normalized E_2 norm is often used, neglecting the division by N of equation 2.7.

The infinity (or inf) norm returns the maximum absolute error over the entire domain (cf. equation 2.8); therefore, it is the most sensitive error measure. It is very proper to detect local discontinuities or singularities.

$$E_i = \max |b_n - \hat{b}_n|, \quad n \in [1, N] \tag{2.8}$$

Besides the given error norms, other error norms are often used in the literature. In the work at hand, we concentrate on the discrete infinity and the E2 norm to report the errors of the individual primary variable field solutions $\mathbf{b}$ compared to $\hat{\mathbf{b}}$.

2.3.3.8 Limitations of the MMS

The MMS is restricted to rigorously assess the accuracy of a code as long as the individual terms have the same formal orders of accuracy. Thus, mixed-order terms cannot be assessed rigorously. An example of CFD can be given: Choosing a first order accurate discretization for the convection part and a second order accurate discretization for the diffusive part, the total accuracy can only be assessed to be (at least) first order. The higher quality of the error reduction of the second order diffusion discretization is overpowered by the lower accurate convection error. Therefore, the ability of the second order discretization cannot be assessed rigorously.

Additionally, coding mistakes which do not affect the order of accuracy of the produced results cannot be verified. A good example is the efficiency of a numerical solution procedure like the Newton-Raphson method for the solution of a nonlinear equation system. The MMS assessment can only verify, whether the correct solution is asymptotically produced by the algorithms at the point of solution convergence. But the method cannot assess, whether the iterative convergence is decreasing with the theoretical rate of the solution procedure. A more detailed discussion about the limitations is given in [91].

Furthermore, there are limitations on the applicability of the MMS. The manufactured solution by definition need to be computed asymptotically by the code. This means that the solution should not contain too short wave-lengths and too high frequencies. For example, if the user can afford a finest time resolution of 1000 steps in a total time of one second, a frequency below one per second of the manufactured solution is recommended to really reach the asymptotic range of the solution.

On the one hand, if a manufactured problem is constructed carelessly regarding the space-time correlations, stability problems are expected, which may make the solution of the problems impossible with the used discretization schemes. On the other hand, the physical restrictions concerning the code have to be considered. This means e.g. that the turbulent kinetic energy of a flow or the density cannot be negative.

2.3.4 Formal Order of Accuracy

The discretization method and the chosen shape functions for spatial and time discretization determine the formal order of accuracy, also called the formal order of convergence. The formal order of convergence $\hat{p}$ of an approximated expression can be determined by comparison with its Taylor series expansion. The first term of the Taylor series expansion, which is not approximated, is called the leading error term. This term represents the main error in the asymptotic range of the solution. In general, the derived formal order can be observed as long as no other errors with smaller convergence rate occur. A detailed discussion of deriving the individual parts is given in the following chapters.

2.3.5 Short Example

For exemplifying the methodology of the order of accuracy test, the approximative calculation of π is investigated. π represents the circumference of a circle with a diameter of 1.

In the following example, the exact solution of $\hat{\pi}$ is approximated on the base of polygon perimeters with an increasing number of edges. The example is adapted from *http://www.logisch-gedacht.de/pi-berechnen/*. As it is commonly known, the perimeter of a circle with radius R is $\hat{U} = 2R\hat{\pi}$.
Assuming the perimeter U and π as unknowns, the solution of $\hat{U}$ has to be approximated to determine π. Starting with (the coarse discretization of) a hexagon to approximate the circle, the perimeter can be calculated as $U_{hexa} = 6R$ and gives an approximation of $\pi_{hexa} = \frac{U_{hexa}}{2R} = 3.0$. Next, a refinement with a factor of $r = 2$ is performed. Therefore, the circle is approximated with a dodecagon, which is a 12-sided figure. The circle and its approximations with a hexagon and a dodecagon are exemplary shown in figure 2.5.

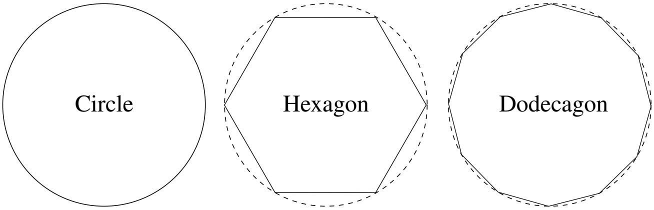


Figure 2.5: A circle and its approximation with a hexagon and a dodecagon

The length of an edge can be calculated using equation 2.9:

$$s_{dodeca} = \sqrt{\left(\frac{R}{2}\right)^2 + \left(R - \sqrt{R^2 - \frac{R^2}{4}}\right)^2} \tag{2.9}$$

Using the calculation of the perimeter ($U_{dodeca} = 12 \cdot s_{dodeca}$), the approximation of the circle gives $\pi_{dodeca} = 3.106$. For further refinement steps with a constant refinement factor of $r = 2$, the edge length of the next refined polygon s_{n+1} can be determined by leveraging the result of the previous coarser polygon s_n as described by equation 2.10:

$$s_{n+1} = \sqrt{\frac{s_n^2}{4} + \left(R - \sqrt{R^2 - \frac{s_n^2}{4}}\right)^2} \tag{2.10}$$

$s_{n=1}$ corresponds to the edge length of the dodecagon. Assuming $R = 1$ for simplicity, equation 2.10 transfers to 2.11:

$$s_{n+1} = \sqrt{\frac{s_n^2}{4} + \left(1 - \sqrt{1 - \frac{s_n^2}{4}}\right)^2} \tag{2.11}$$

The polygon with the edge length s_{n+1} has $6 \cdot 2^{n+1}$ edges. π can therefore be approximated by a polygon with equation 2.12

$$\pi_{n+1} = \frac{1}{2} \cdot 6 \cdot 2^{n+1} \cdot \sqrt{\frac{s_n^2}{4} + \left(1 - \sqrt{1 - \frac{s_n^2}{4}}\right)^2} \tag{2.12}$$

The individual error of the approximations is directly calculated by the difference of the calculated result π_n and the target solution $\hat{\pi}$. The results and the errors for a few refinement steps with a refinement factor of $r = 2$ are given in table 2.1, using 19 digits, calculated with Microsoft Excel®.

Table 2.1: Short example: Approximation of π using polygons

number of edges	approximation of $\hat{\pi}$: π_n	error size $E_n = \hat{\pi} - \pi_n$
6	3.0	0.141592653589793000
12	3.105828541230250000	0.035764112359544200
24	3.132628613281240000	0.008964040308555350
48	3.139350203046870000	0.002242450542926380
96	3.141031950890510000	0.000560702699283322
192	3.141452472285460000	0.000140181304330689
384	3.141557607911860000	0.000035045677935219
768	3.141583892148320000	0.000008761441474547
1536	3.141590463228050000	0.000002190361742649
3072	3.141592105999270000	0.000000547590521371
6144	3.141592516692160000	0.000000136897635450

Using the produced results of the refinement study in table 2.1, the observed order of accuracy can be obtained applying equation 2.2 as shown in equation 2.13 [57]:

$$p_{n+1} = \frac{\log\left(\frac{E_n}{E_{n+1}}\right)}{\log(r)} \tag{2.13}$$

The pairwise results of the observed order of accuracy p are depicted in table 2.2.

Table 2.2: Short example: Observer order of accuraccy p

number of edges coarse grid	fine grid	refinement factor r	observed order of accuracy p
6	12	2	1.98516187
12	24	2	1.99629159
24	48	2	1.99907297
48	96	2	1.99976825
96	192	2	1.99994206
192	384	2	1.99998552
384	768	2	1.99999638
768	1536	2	1.99999909
1536	3072	2	1.99999977
3072	6144	2	1.99999995

Beside the table representation of the error and the observed order of accuracy in tables 2.1 and 2.2, a graphical representation of the results is shown in figure 2.6 that is further explained in the following paragraphs. The error E_n is plotted in a log-log diagram in figure 2.6 in the left panel. The negative slope of the log-log diagram of figure 2.6 indicates that the error is decreasing during refinement and tends towards machine accuracy. Therefore it indicates the convergence of the computation of π towards $\hat{\pi}$.

The slope of the error graph is calculated at each refinement step from a pair of errors using equation 2.13. The pairwise calculated error slope is drawn as a graph in figure 2.6 in the right panel. As seen in equation 2.13, the calculated slope represents exactly the observed order of accuracy p. In order to judge a simulation successfully, the observed order of accuracy must reach the formal order of accuracy with refinement. Using the results in table 2.2 and figure 2.6, the observed order of accuracy for systematic refined approximations reaches $p = 2$ asymptotically. The example used polygons for the approximation of the circle. The polygons are described as a combination of linear functions. As the formal order of accuracy for the arc length calculation of an arbitrary curve using linear functions is $\hat{p} = 2.0$, the calculation process therefore is positively assessed respectively verified.

Finally, it has to be pointed out that this example does not fulfill all requirements of the MMS as the target solution $\hat{\pi}$ actually is an infinite long number. Nevertheless, the example clearly demonstrates the general procedure and the purpose of order of accuracy tests like the Method of Manufactured Solutions.

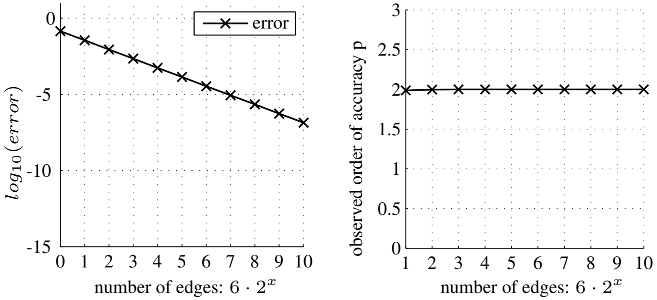


Figure 2.6: Error plot of the approximation of π, $\log - \log$ plot (left) and the observed order of accuracy p with refinement (right)

2.4 Assessment of a Linear Error Estimator

During the iterative solution of an equation system, it is very helpful to know, when the solution is converged. This is equivalent to the fact that the IICE converges to a very small number (cf. section 2.3.3.3). For the abandoning of an iterative solution, in general a convergence criterion is set. As the real iterative error is not known a priori, error estimators are used for the determination of the IICE. The linear error estimator with its reasonable computational effort that has been proposed, among others, in [30], is a popular tool. The estimator is originally developed for the error estimation of linear equation systems, but is also commonly used for nonlinear equation systems.

Subsequently, the calculation procedure of the error estimator is briefly described and then illustrated by an application.

2.4.1 Linear Error Estimator

The difference between two consecutive solution fields ϕ at the iterative steps n and $n+1$ is given in equation 2.14.

$$\boldsymbol{\delta}^n = \boldsymbol{\phi}^{n+1} - \boldsymbol{\phi}^n \tag{2.14}$$

λ_1 defines the largest eigenvalue, also called the spectral radius of the iteration matrix. It can be estimated by equation 2.15.

$$\lambda_1 \approx \frac{\parallel \boldsymbol{\phi}^n \parallel}{\parallel \boldsymbol{\phi}^{n-1} \parallel} \tag{2.15}$$

As error norm, the discrete E_2 norm (cf. equation 2.7) is chosen. The error ϵ in every point respectively the total error e can therefore be estimated using equation

2.16 respectively equation 2.17 :

$$\boldsymbol{\epsilon}_1^n \approx \frac{\boldsymbol{\delta^n}}{\lambda_1 - 1} \tag{2.16}$$

$$e = \parallel \boldsymbol{\epsilon}_1^n \parallel \approx \frac{\parallel \boldsymbol{\delta^n} \parallel}{\lambda_1 - 1} \tag{2.17}$$

2.4.2 Application

The application of the error estimator is exemplarily shown for the steady state CFD benchmark with a grid of 128 by 128 cells, given in section 4.12.10.2. The IICE error e is, on the one hand, estimated using equation 2.17. On the other hand, the IICE error is determined as the difference of the finally converged solution and the current solution in step n. During the application, the complete numerical error NE, containing the discretization error as well, is known by the analytical MMS solution. These three errors are illustrated over the number of iterations n in the figures 2.7 and 2.8 for the velocity $\mathbf{u}$ respectively the pressure P. In the given figures 2.7 and 2.8, it is obvious that the error estimation for the nonlinear equation system solved by an iterative scheme is almost matching the error estimated by the fully converged solution.
Furthermore, the amount of discretization error in the example is much larger than the IICE error. This information could be used for the setup of more efficient computations. Combining the informations of discretization error estimations with the IICE estimation, many iterations can be saved without loosing accuracy of the simulation results. It can be additionally seen that the machine accuracy does not limit the simulation as the errors do not reach a horizontal line.

2.5 Setup of Iterative Solver Tolerances

As presented above, many error sources enter the simulation procedure. Beside the discretization error, the numerical solution contains IICE and round-off errors (cf. figure 2.4). The commonly known statement of the literature (e.g. in [73]) that double machine accuracy brings the computer round-off far beyond the other contributing errors, coincides with the results of simulations shown in the following chapters.

IICE error is always induced as soon as the software solves an equation in an iterative process. On the one hand, there are nonlinear or coupled equations which cannot be solved directly. On the other hand, the iterative solution of a linear equation system induces IICE, which is especially relevant for the solution of large equation systems. Simply spoken, this topic concerns almost all engineering software, including the presented partitioned FSI environment and also FSI simulations in general.

The presented MMS assessment framework always requires a minimization of the IICE error in order to isolate the discretization error from the numerical error (cf. figure 2.4). For the software application on "real" problems, the setup of the numerical effort should be balanced to reach the most accurate result for the individual computational effort. This means precisely that the iterative convergence criterion can be set to the magnitude of the discretization error without loosing accuracy of the solution and saving computational costs compared to a fully converged iterative solution. An example of this setup is given above. In figures 2.7 and 2.8, the IICE and the total numerical error NE can be observed. In this case, the pressure and the velocity solutions iterated equal to $1 \cdot 10^{-10}$. The numerical error curves (dotted lines) indicate that the discretization error dominates the total error from an iterative convergence of $\approx 1 \cdot 10^{-5}$ on. From this stage onwards the effort to converge the iterative solution to the level of $1 \cdot 10^{-10}$ has not noticeably improved the solution and can, therefore, be regarded as waste of computational effort.

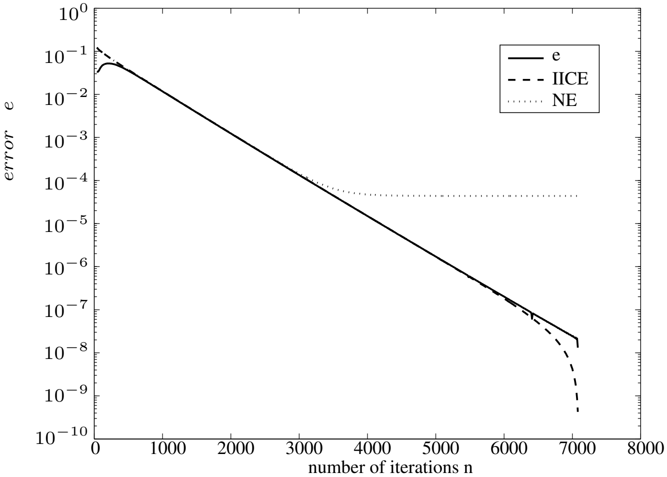


Figure 2.7: Error development over the nonlinear iteration of the velocity solution $\mathbf{u}$

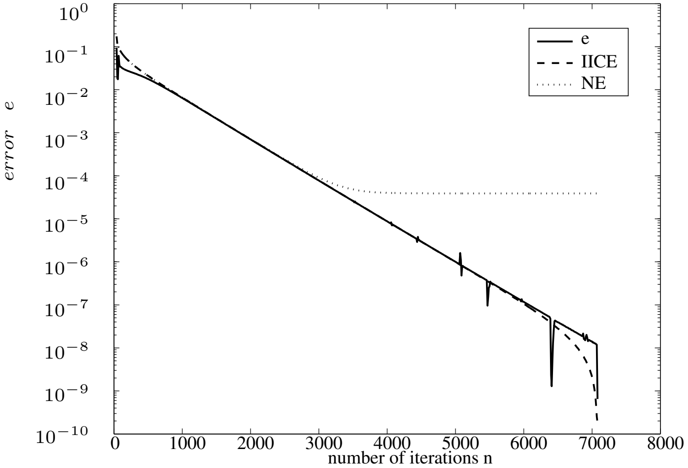


Figure 2.8: Error development over the nonlinear iteration of the pressure solution P

CHAPTER **3**

Structural Dynamics

Structural Dynamics is concerned with the behavior of structures subjected to static and dynamic loads. In the work at hand, a special focus is on the modeling of membrane structures with large deformations.
The chapter aims at providing an overview of the requirements, the assumptions, the balance equations, and the implementations of the structural dynamics with the named geometrical nonlinear membrane element in the CSD software Carat++. The notation of the equations use the Einstein summation convention. A detailed description can be found, e.g., in [6, 38, 122].

For the application of the MMS in the context of structural dynamics, a framework for the assessment is presented. Furthermore, the application of the MMS, the development of a hierarchical benchmark series, and the assessment results are elaborated. Using these results and the derived formal order of accuracy, the CSD code can be assessed for the intended application.

The derivation of the balance equations of the membrane is kept compact. A more detailed derivation containing discussions of different possibilities to model membrane structures using Finite Elements can be found in [19, 43, 68].

3.1 General Assumptions

As defined in section 1.3, the structural part of the FSI analysis consists of flexible, elastic membrane structures containing prestress. Prestress is a typical matter of membranes as these structures in general are pre-tensioned or prestressed to generate an initial stiffness. For the target application of the membrane structures, large deformations with small strains are expected. Therefore, a linear isotropic material with the plane stress assumption and a fully nonlinear strain measure is used. As the structure will behave in an unsteady manner, inertial/mass forces are necessary beside the inner forces out of stresses, while structural damping is neglected.

For the spatial description, the Total Lagrangian approach is chosen. The discretization of the structural analysis part is performed using the Finite Element method in space and a Newmark scheme over time. An overview of the assessed features of the software Carat++ is given in figure 3.1.

CSM - Carat++
Finite Element Method
Total Lagrangian Formulation
Unsteady state using Newmark time discretization rule
Isotropic St. Venant-Kirchhoff material
Nonlinear strains
Large deflections
Membrane structures
Plane stress assumption
Pre-stress

Figure 3.1: Overview of the CSD software to assess: Carat++

3.2 Equilibrium

In structural dynamics, the conservation of momentum in every point of a continuous structure defines the interior equilibrium equation in the so-called strong form. It can be shown that the equilibrium with respect to the current/deformed configuration (equation 3.1) and the initial configuration of a structure (equation 3.2) are equivalent [123].

Equations 3.1 and 3.2 describe the conservation of momentum in the Lagrangian description derived from Cauchy's first equation of motion [6, 45]. Herein, ρ_s represents the density, which is assumed to be constant, and t the time. $\mathbf{d}_c = \mathbf{d}(\mathbf{x}, t)$ is the field of displacements with respect to the current configuration $\mathbf{x}$, and $\mathbf{d} = \mathbf{d}(\mathbf{X}, t)$ with respect to the initial configuration $\mathbf{X}$ [45]. Analogously, the fields of displacements $\mathbf{q}$ and $\mathbf{Q}$ are the volume forces with respect to the current respectively the initial configuration. $\boldsymbol{\sigma}$ represents the Cauchy and $\mathbf{P}$ the first Piola-Kirchhoff (PK1) stress tensor [123]. It can be shown that the presented equilibria in the strong form can be transformed into the commonly known weak forms of equilibrium using, e.g., the principle of virtual work in equation 3.3 [46, 71, 123]. In equation 3.3, Ω represents the initial domain and $\mathbf{T}$ the traction forces on the surface A of Ω. $\mathbf{S}$ represents the second Piola-Kirchhoff (PK2) stress and $\mathbf{E}$ the Green-Lagrangian strain tensor. The named equivalence of the weak and the strong form compose a key feature for the applicability of the MMS. With the goal to perform the MMS simulations, all necessary terms of the strong form equilibrium (equation 3.1 respectively 3.2) have to be determined, completely independent of the implementation of the equilibrium (e.g., equation 3.3).

$$-\rho_s \frac{d^2\mathbf{d}_c}{dt^2} + \nabla \cdot \boldsymbol{\sigma} + \rho_s \mathbf{q} = 0 \tag{3.1}$$

$$-\rho_s \frac{\partial^2\mathbf{d}}{\partial t^2} + \nabla \cdot \mathbf{P} + \rho_s \mathbf{Q} = 0 \tag{3.2}$$

$$\delta W = -\int_\Omega \rho_s \frac{\partial^2\mathbf{d}}{\partial t^2} \delta\mathbf{d} d\Omega + \int_\Omega \mathbf{S} : \delta\mathbf{E} d\Omega - \int_A \mathbf{T}\delta\mathbf{d} dA + \int_\Omega \rho_s \mathbf{Q}\delta\mathbf{d} d\Omega = 0 \tag{3.3}$$

3.3 Kinematics

The kinematics of the membrane are shown in figure 3.2. Capital and lower case letters generally indicate, whether quantities belong to the initial (e.g., $\mathbf{X}$) or the current/deformed configuration (e.g., $\mathbf{x}$) respectively. Using equations 3.4 and 3.5, the covariant base vectors in the initial ($\mathbf{G}_\alpha$) and the deformed configuration ($\mathbf{g}_\alpha$) can be derived. θ^α are the surface parameters along the base vector $\mathbf{G}_\alpha$. Greek index letters always count from 1 to 2, and Latin index letters always count from 1 to 3 at the Einstein summation convention (cf. [38, 122]). As membranes are very thin, their kinematics can be completely described using the two named in-plane base vectors. The third base vectors $\mathbf{G}_3$ respectively $\mathbf{g}_3$ are constructed as normalized cross-product of the first two base vectors to be orthogonal on the tangent plane. 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 the contravariant base vectors can be performed with the aid of the contravariant metric ($g^{ij} = (g_{ij})^{-1}$) using $\mathbf{g}^i = g^{ij}\mathbf{g}_j$.

The deformation gradient $\mathbf{F}$ is calculated in equation 3.6, where $\otimes$ represents the dyadic product [45, 123]. The Green-Lagrangian strain tensor $\mathbf{E}$ is calculated using $\mathbf{F}$ and the unity tensor $\mathbf{I}$ in equation 3.7.

$$\mathbf{x} = \mathbf{X} + \mathbf{d} = X^i e_i + d^i e_i \tag{3.4}$$

$$\mathbf{G}_\alpha = \frac{\partial \mathbf{X}}{\partial \theta^\alpha} \qquad \mathbf{g}_\alpha = \frac{\partial \mathbf{x}}{\partial \theta^\alpha} \tag{3.5}$$

$$\mathbf{F} = \mathbf{g}_\alpha \otimes \mathbf{G}^\alpha \tag{3.6}$$

$$\mathbf{E} = \frac{1}{2} \cdot \left(\mathbf{F}^\mathbf{T}\mathbf{F} - \mathbf{I}\right) \tag{3.7}$$

3.4 Constitutive Equations

The isotropic St. Venant-Kirchhoff material combined with the plane stress assumptions can be calculated with the aid of modified Lamé parameters λ_m and μ_m [68, 124] in equation 3.8 with 3.9. Using the material tensor $\mathbf{C}$ and the strain tensor $\mathbf{E}$, the PK2 stress tensor $\mathbf{S}$ can be calculated (equation 3.10). E represents the Young's

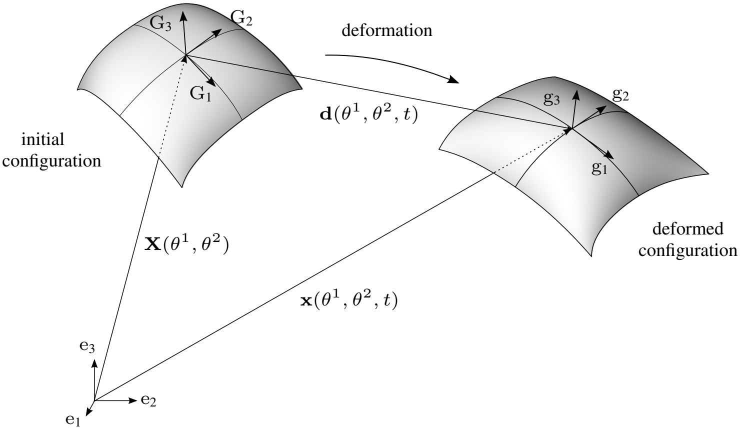


Figure 3.2: Different configurations and the deformation process of a surface body

modulus, and ν the Poisson's ratio of the material.

$$\begin{aligned}\mathbf{C} &= C^{\alpha\beta\gamma\delta}\mathbf{G}_\alpha \otimes \mathbf{G}_\beta \otimes \mathbf{G}_\gamma \otimes \mathbf{G}_\delta \\ &= \lambda_m \cdot G^{\alpha\beta}G^{\gamma\delta} + \mu_m \left(G^{\alpha\gamma}G^{\beta\delta} + G^{\alpha\delta}G^{\beta\gamma}\right) \qquad (3.8)\end{aligned}$$

$$\lambda_m = \frac{E \cdot \nu}{(1-\nu^2)} \qquad \mu_m = \frac{E}{2 \cdot (1+\nu)} \qquad (3.9)$$

$$\mathbf{S} = \mathbf{C} : \mathbf{E} \qquad (3.10)$$

3.5 Stress

Additionally to the stresses caused by strains, membranes in general are prestressed. The prestress of the present element is set and defined with respect to the initial configuration. The prestress tensor $\mathbf{S}_{ps}$ is therefore added to the PK2 stress tensor [68]. As the equilibrium of momentum (equations 3.1 and 3.2) contains the Cauchy stress tensor $\boldsymbol{\sigma}$ respectively the PK1 stress tensor $\mathbf{P}$, they can be evaluated from the present PK2 stress tensor with equations 3.11 and 3.12 [45, 123].

$$\boldsymbol{\sigma} = \frac{1}{\det(\mathbf{F})}\mathbf{F}\left(\mathbf{S} + \mathbf{S}_{ps}\right)\mathbf{F}^T \qquad (3.11)$$

$$\mathbf{P} = \mathbf{F}\left(\mathbf{S} + \mathbf{S}_{ps}\right) \qquad (3.12)$$

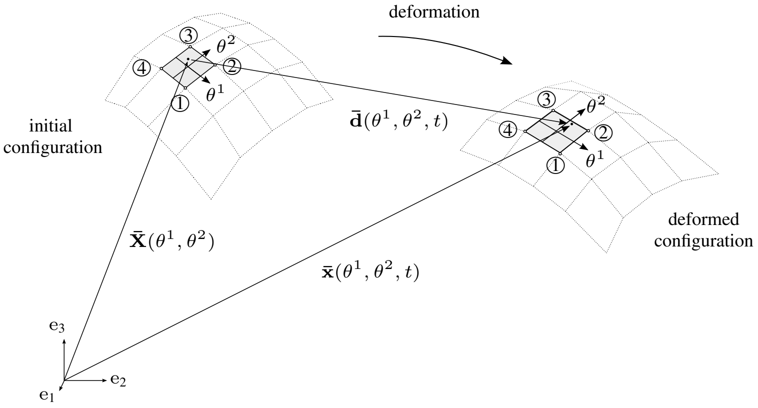


Figure 3.3: Discretized configurations of the deformation process of a surface body

3.6 Discretization

3.6.1 Spatial Discretization

The chosen spatial discretization method is known as the Finite Element method, e.g. in [5, 46, 106, 123, 127]. Herein, the domain is approximated by a set of non-overlapping elements constructed by nodes. Figure 3.3 gives an example of a discretized surface using 16 four-node elements. Following the concept of isoparametric finite elements, the field information, such as the geometry $\mathbf{X}$ respectively $\mathbf{x}$ or the displacement field $\mathbf{d}$ are approximated by the fields $\bar{\mathbf{X}}$, $\bar{\mathbf{x}}$, and $\bar{\mathbf{d}}$ (compare figure 3.2 and 3.3). This approximation is described using a finite number of nodal information at a node i (e.g., $\tilde{\mathbf{d}}_i$) combined with a set of shape functions N_i of the solution. The displacement field between the discrete nodal information is approximated by a chosen shape function in the directions of θ^1 and θ^2. Therefore, the deformation $\mathbf{d}$ of a discrete element is a combination of the deformation of its nodes:

$$\mathbf{d}(\theta^1,\theta^2,t) \approx \bar{\mathbf{d}}(\theta^1,\theta^2,t) = \sum_{i=1}^{n_{nodes}} N_i(\theta^1,\theta^2)\tilde{\mathbf{d}}_i(t) \tag{3.13}$$

The mentioned isoparametric elements use the same shape functions for the description of the geometry as for the solution field. Therefore, the geometry in the initial and the deformed configuration respectively is described by:

$$\mathbf{X}(\theta^1, \theta^2) \approx \bar{\mathbf{X}}(\theta^1, \theta^2) = \sum_{i=1}^{n_{nodes}} N_i(\theta^1, \theta^2)\tilde{\mathbf{X}}_i \tag{3.14}$$

$$\mathbf{x}(\theta^1, \theta^2, t) \approx \bar{\mathbf{x}}(\theta^1, \theta^2, t) = \sum_{i=1}^{n_{nodes}} N_i(\theta^1, \theta^2)\tilde{\mathbf{x}}_i(t) \tag{3.15}$$

In the context of this work, quadrilateral elements with bilinear shape functions are used. This means, the number of nodes $n_{nodes} = 4$ and the shape functions $N_i(\theta^1, \theta^2)$ are formulated as shown in equations 3.16-3.19. For the sake of simplicity, the discrete nodal information of the geometry and displacements are noted without the tilde.

$$N_1(\theta^1, \theta^2) = \frac{1}{4}\left(1 + \theta^1\right)\left(1 + \theta^2\right) \tag{3.16}$$

$$N_2(\theta^1, \theta^2) = \frac{1}{4}\left(1 - \theta^1\right)\left(1 + \theta^2\right) \tag{3.17}$$

$$N_3(\theta^1, \theta^2) = \frac{1}{4}\left(1 - \theta^1\right)\left(1 - \theta^2\right) \tag{3.18}$$

$$N_4(\theta^1, \theta^2) = \frac{1}{4}\left(1 + \theta^1\right)\left(1 - \theta^2\right) \tag{3.19}$$

$$\text{with} \quad \theta^1 \in [-1; 1] \quad \text{and} \quad \theta^2 \in [-1; 1]$$

3.6.2 Time Discretization

For the discretization of the time derivative respectively the time integration, an explicit Newmark scheme is used and will be presented [72]. The velocity $\mathbf{v}$ is equal to the first time derivative of the displacement field $\mathbf{d}$ (cf. equation 3.20). The acceleration $\mathbf{a}$ is equal to the second time derivative of the displacement field $\mathbf{d}$ (cf. equation 3.21).

$$\mathbf{v} = \frac{\partial \mathbf{d}}{\partial t} = \dot{\mathbf{d}} \tag{3.20}$$

$$\mathbf{a} = \frac{\partial^2 \mathbf{d}}{\partial t^2} = \ddot{\mathbf{d}} \tag{3.21}$$

The continuous time period $[t_0, T]$ is split into discrete time increments m with a size of Δt. With the assumption of a constant Δt, the current time t_m therefore is described with equation 3.22:

$$t_m = t_0 + \Delta t \cdot m \tag{3.22}$$

The Newmark time integration is able to describe and approximate the displacements, the velocity and the acceleration of the new time step t_{m+1} completely in known terms of the last time step m and the primary variable of the balance equations. In the present case, the primary variable is the displacement field $\mathbf{d}$. Therefore, the velocity and the acceleration at time step $m+1$ are described in equations 3.23 and 3.24 [19, 60]:

$$\dot{\mathbf{d}}_{m+1} = \frac{\gamma}{\beta\Delta t}[\mathbf{d}_{m+1} - \mathbf{d}_m] - \frac{\gamma-\beta}{\beta}\dot{\mathbf{d}}_m - \frac{\gamma-2\beta}{2\beta}\Delta t\ddot{\mathbf{d}}_m \tag{3.23}$$

$$\ddot{\mathbf{d}}_{m+1} = \frac{1}{\beta\Delta t^2}[\mathbf{d}_{m+1} - \mathbf{d}_m] - \frac{1}{\beta\Delta t}\dot{\mathbf{d}}_m - \frac{1-2\beta}{2\beta}\ddot{\mathbf{d}}_m \tag{3.24}$$

An intermediate time step approximation, where the displacements, the velocities, and the accelerations are approximated using a linear interpolation between the two timesteps m and $m+1$ as introduced in [16] is given in equations 3.25-3.27:

$$\mathbf{d}_{m+1-\alpha_f} = (1-\alpha_f)\mathbf{d}_{m+1} + \alpha_f\mathbf{d}_m \tag{3.25}$$

$$\dot{\mathbf{d}}_{m+1-\alpha_f} = (1-\alpha_f)\dot{\mathbf{d}}_{m+1} + \alpha_f\dot{\mathbf{d}}_m \tag{3.26}$$

$$\ddot{\mathbf{d}}_{m+1-\alpha_m} = (1-\alpha_m)\ddot{\mathbf{d}}_{m+1} + \alpha_m\ddot{\mathbf{d}}_m \tag{3.27}$$

The choice of the parameters α_f, α_m, β, and γ regulate the stability and the order of accuracy of the discretization method. A detailed discussion of the method and its accurate choice of the parameters is presented in [36, 60].

3.7 Formal Order of Accuracy

The discretization method, the chosen shape functions for spatial and time discretization, and the numerical integration determine the formal order of convergence [5, 106, 127].

The formal order of accuracy $\hat{p}$ of an approximated expression can be determined 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. 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 can be found in [106].

3.7.1 Taylor Series Expansion of d

The basis of an assessment of the formal order of convergence $\hat{p}$ of a variable field $\mathbf{d}$ is the comparison of its Taylor series expansion with its approximation. The Taylor series of $\mathbf{d}$ is given over time and space in equations 3.28 and 3.29. The spatial Taylor series expansion is simplified here to one dimension. The superscript m indicates the initial timestep in equation 3.28. The subscript p indicates the initial location in equation 3.29.

$$\begin{aligned}\mathbf{d}(t^m+\Delta t) =&\mathbf{d}(t^{m+1}) = \mathbf{d}^m + \Delta t\left(\frac{\partial \mathbf{d}}{\partial t}\right)^m + \frac{\Delta t^2}{2}\left(\frac{\partial^2 \mathbf{d}}{\partial t^2}\right)^m + \\ &\frac{\Delta t^3}{3!}\left(\frac{\partial^3 \mathbf{d}}{\partial t^3}\right)^m + ... + \frac{\Delta t^n}{n!}\left(\frac{\partial^n \mathbf{d}}{\partial t^n}\right)^m \end{aligned} \tag{3.28}$$

$$\begin{aligned}\mathbf{d}(\theta) =&\mathbf{d}_p + (\theta-\theta_p)\left(\frac{\partial \mathbf{d}}{\partial \theta}\right)_p + \frac{(\theta-\theta_p)^2}{2}\left(\frac{\partial^2 \mathbf{d}}{\partial \theta^2}\right)_p + \\ &\frac{(\theta-\theta_p)^3}{3!}\left(\frac{\partial^3 \mathbf{d}}{\partial \theta^3}\right)_p + ... + \frac{(\theta-\theta_p)^n}{n!}\left(\frac{\partial^n \mathbf{d}}{\partial \theta^n}\right)_p \end{aligned} \tag{3.29}$$

3.7.2 Spatial Discretization

3.7.2.1 Shape Functions

The chosen linear shape functions in the software Carat++ in equations 3.16-3.19 combined with equation 3.13 can be rewritten for simplicity's sake in one dimension in equation 3.30:

$$\begin{aligned}\mathbf{d}(\theta) =&\sum_{i=1}^{2} N_i(\theta)\tilde{\mathbf{d}}_i \\ &= \mathbf{d}(\theta=-1)\frac{1-\theta}{2} + \mathbf{d}(\theta=1)\frac{1+\theta}{2} \\ &= \mathbf{d}_1 + (\theta+1)\left(\frac{\partial \mathbf{d}}{\partial \theta}\right)_1 \end{aligned} \tag{3.30}$$

Comparing the used approximation of $\mathbf{d}(\mathbf{x})$ in equation 3.30 with its spatial Taylor series in equation 3.29, the error E of the approximation can directly be derived in equation 3.32. The developing point for the Taylor series is $\theta_p = \theta_1 = -1$.

$$\begin{aligned}E =&\frac{(\theta-\theta_p)^2}{2}\left(\frac{\partial^2 \mathbf{d}}{\partial \theta^2}\right)_p + \\ &\frac{(\theta-\theta_p)^3}{3!}\left(\frac{\partial^3 \mathbf{d}}{\partial \theta^3}\right)_p + ... + \frac{(\theta-\theta_p)^n}{n!}\left(\frac{\partial^n \mathbf{d}}{\partial \theta^n}\right)_p \end{aligned} \tag{3.31}$$

$$E = \mathcal{O}\left((\theta+1)^2\right) = \mathcal{O}\left((\Delta\theta)^2\right) \tag{3.32}$$

The result of equation 3.32 combined with equation 2.1 represents a $\hat{p} = 2$ in space. In line with [127] p. 32 ff, the error for an isoparametric polynomial approximation of order j is $\mathcal{O}(h^{\hat{p}}) = \mathcal{O}(h^{j+1-m})$. h defines the characteristic mesh size and m the magnitude of the mth derivative of the primary variable. This conforms to the statements in [34, 46, 106, 109].

In the present case, using bilinear form functions ($j = 1$), the displacement (primary variable, $m = 0$) convergences with a formal order of $\hat{p} = 2$, while stresses and strains ($m = 1$) converge with a formal order of $\hat{p} = 1$.

3.7.2.2 Geometric Approximation

Membranes are generally curved structures in space. In [106, 127] is shown that the defined formal order of accuracy in the previous section already contains the errors of geometrical approximation as it is an isoparametric approach. Nevertheless, it should be mentioned and assessed separately within this work as it introduces an additional source of error compared to plane structures. As already mentioned, the isoparametric concept uses the same shape functions for the geometrical description and the variable fields. This means, the description of an arc length is approximated with the formal order elaborated in the previous section.

This can exemplarily be confirmed in the example given in section 2.3.5. In this example, π is estimated using polygons for the calculation of the perimeter of a circle. Polygons are equal to linear shape functions. Using an increasing number of edges of the polygon the resulting estimation of π constantly improves towards the exact solution. The reduction of the error and the observed order of accuracy are shown in tables 2.1 and 2.2. Finally, this example confirms the order of accuracy of the arc length approximation of a curve to $\hat{p} = 2$. This knowledge of the accuracy of the arc length approximation in 1D can equally be transferred and is valid for the approximation of the surface integral in 2D as well.

3.7.2.3 Spatial Integration

Following [127], the spatial integration of the discretized terms in equation 3.3 should not affect the solution. This means, the chosen numerical integration should be at least as exact/accurate as the integrand. It is shown in [80, 127] that the integration should be exact to the order $2(j-k)$, where again j is the order of polynomial approximation and k is the highest order of derivatives occurring in the integrand. Therefore, the demanded order of accuracy of the integration is $\mathcal{O}\left(h^{2(j-k)+1}\right)$.

In the present case of using linear shape functions, the integral of the inner forces needs an integration of order $\mathcal{O}\left(h^{2(j-k)+1}\right) = \mathcal{O}\left(h^{2(1-1)+1}\right) = \mathcal{O}\left(h^{1}\right)$ respectively an integration, which is exact to the order $2(j-k) = 2(1-1) = 0$. The outer forces and the inertia term need an integration of order $\mathcal{O}(h^{2(j-k)+1} = \mathcal{O}\left(h^{2(1-0)+1}\right) = \mathcal{O}(h^3)$ respectively an integration, which is exact to the order $2(j-k) = 2(1-0) = 2$.

According to the Gaussian integration presented in [59, 107], the integral is evaluated using so-called Gauss points and weighting functions. A Gaussian integral "is exact for all polynomials of degree $\leq 2 \cdot n_p - 1$" [107]. This defines the number of required Gauss points n_p in every dimension to avoid the introduction of additional errors by numerical integration. Therefore, the required number of Gauss points n_p in each direction of a quadrilateral element can be evaluated using equation 3.33:

$$n_p \geq \frac{2(j-k)+1}{2} \qquad \text{with} \quad n_p \in \mathbb{N} \tag{3.33}$$

Assuming linear shape functions, the integration of the inner forces at least needs $n_p \geq 1$, and the outer forces and the inertia forces at least need $n_p \geq 2$ at every element in every direction in space.

3.7.3 Temporal Discretization

As given in section 3.6.2, the parameters α_f, α_m, β, and γ must be preset for the time discretization using the Newmark method. For a formal second order of accuracy and minimal numerical damping, the choice of parameters is given in equation 3.34 [36, 60].

$$\alpha_f = 0.50, \quad \alpha_m = 0.50, \quad \beta = 0.25, \quad \gamma = 0.50 \tag{3.34}$$

3.7.4 Resulting Formal Order of Accuracy

The present code of Carat++ uses bilinear shape functions (cf. equations 3.16-3.19) for the quadrilateral elements. For the numerical integration, two Gauss points are used for both directions, which is at least the number of required Gauss points given in equation 3.33. The Newmark time integration scheme always uses the parameter choice given in section 3.7.3. Therefore, the formal time and spatial order of accuracy ($\hat{p}_t$ respectively $\hat{p}_s$) of the code is given in equation 3.35:

$$\hat{p}_s = \hat{p}_t = \hat{p} = 2 \tag{3.35}$$

3.8 Application of the MMS

3.8.1 Equilibrium Forces

The equilibrium forces required to reach a prescribed deformation $\mathbf{d} = \hat{\mathbf{d}}$ in the context of the MMS are shown in equations 3.36 and 3.37, avoiding additional outer volume or area forces. For the calculation of the manufactured volume or surface traction forces at the membrane, it is recommended to use a symbolic computation software, such as Maple®or Mathematica®. In order to facilitate understanding, the stress tensors are shown as a function of the target displacements $\hat{\mathbf{d}}$ (e.g., $\mathbf{P} = \mathbf{P}(\hat{\mathbf{d}})$). The general stress tensor components σ^{ij} or P^{ij} can be reduced to the in-plane stresses $n^{\alpha\beta}$ respectively $N^{\alpha\beta}$ in the membrane theory [4]. $\mathbf{n}^{\alpha}|_{\alpha}$ represents the covariant derivative of $\mathbf{n}^{\alpha}$ [4, 123].

$$\hat{\mathbf{q}}_s = \frac{\hat{\mathbf{t}}_s}{b} = \rho \frac{\partial^2 \hat{\mathbf{d}}_c}{\partial t^2} - \nabla \cdot \boldsymbol{\sigma}(\hat{\mathbf{d}}_c) = \rho \frac{\partial^2 \hat{\mathbf{d}}_c}{\partial t^2} - \mathbf{n}^{\alpha}|_{\alpha} = \rho \frac{\partial^2 \hat{\mathbf{d}}_c}{\partial t^2} - \left(n^{\alpha\beta}\mathbf{g}_{\beta}\right)|_{\alpha} \tag{3.36}$$

$$\hat{\mathbf{Q}}_s = \frac{\hat{\mathbf{T}}_s}{B} = \rho \frac{\partial^2 \hat{\mathbf{d}}}{\partial t^2} - \nabla \cdot \mathbf{P}(\hat{\mathbf{d}}) = \rho \frac{\partial^2 \hat{\mathbf{d}}}{\partial t^2} - \mathbf{N}^{\alpha}|_{\alpha} = \rho \frac{\partial^2 \hat{\mathbf{d}}}{\partial t^2} - \left(N^{\alpha\beta}\mathbf{g}_{\beta}\right)|_{\alpha} \tag{3.37}$$

The calculated forces $\hat{\mathbf{q}}_s$ respectively $\hat{\mathbf{Q}}_s$ in equations 3.36 or 3.37 are volume forces. To generate area forces $\hat{\mathbf{t}}_s$ respectively $\hat{\mathbf{T}}_s$ which act on the mid-plane of the thin assumed membrane, they have to be integrated over the thickness b respectively B. As the stresses are assumed to be constant over the thickness, the integrations results in an multiplication with the thickness shown in equations 3.38 and 3.39:

$$\hat{\mathbf{t}}_s = \int_{-\frac{b}{2}}^{\frac{b}{2}} \hat{\mathbf{q}}_s d\theta^3 = \hat{\mathbf{q}}_s b \tag{3.38}$$

$$\hat{\mathbf{T}}_s = \int_{-\frac{B}{2}}^{\frac{B}{2}} \hat{\mathbf{Q}}_s d\theta^3 = \hat{\mathbf{Q}}_s B \tag{3.39}$$

It is emphasized again that the derived forces represent the equilibrium forces of the problem for a desired displacement field $\hat{\mathbf{d}}$. Therefore the mentioned surface or volume force has to be applied to the code regardless of the discretization method of the implemented equilibrium. All developed equilibria are based on the strong form equilibria (equation 3.36 or 3.37). Thus the generality of the method is evident.

3.8.2 Boundary and Initial Conditions

The Dirichlet and Neumann boundary conditions (BC) for the deformed and the initial configuration are set with equations 3.40 respectively 3.41 (cf. section 2.3.3.4 and [123]). $\mathbf{N}$ and $\mathbf{n}$ represent the in-plane normal vector and $\mathbf{T}$ and $\mathbf{t}$ are the traction vectors on the edges of the initial (Γ) and the deformed (γ) configuration respectively. In steady-state simulations, the initial conditions (IC) are set to a non-balanced state, e.g., to a zero value field. In transient problems the IC must be set to the target value of the variable. Therefore, at $t = t_0$ the initial configuration matches the deformed configuration (equation 3.42 for both configurations).

$$\mathbf{d}_c^\gamma = \hat{\mathbf{d}}_c^\gamma \qquad \mathbf{t}^\gamma = \sigma^\gamma \mathbf{n}^\gamma \tag{3.40}$$

$$\mathbf{d}^\Gamma = \hat{\mathbf{d}}^\Gamma \qquad \mathbf{T}^\Gamma = \mathbf{P}^\Gamma \mathbf{N}^\Gamma \tag{3.41}$$

$$\mathbf{d}(\mathbf{X}, t = t_0) = \mathbf{d}_c(\mathbf{x}, t = t_0) = \hat{\mathbf{d}}(\mathbf{X}, t = t_0) = \hat{\mathbf{d}}_c(\mathbf{x}, t = t_0) \tag{3.42}$$

In structural dynamics, a Dirichlet BC is equivalent to a support condition (e.g., displacement equal to zero) and a Neumann BC is equivalent to a edge force at the boundary. In general, the benchmarks defined in the subsequent sections do not apply boundary conditions at the edges of the structure. If they are applied, they are given in the examples separately.

3.8.3 Framework

The framework to assess the simulation code Carat++, using EMPIRE for the application of the MMS, is presented in this section. In total, five software components are used in this framework:

- Carat++ (already presented above)
- Maple®
- mmsClient

- Emperor
- Matlab®

An overview of the framework is given in figure 3.4. The procedure and the sequence of the program executions are shown in figure 3.5.

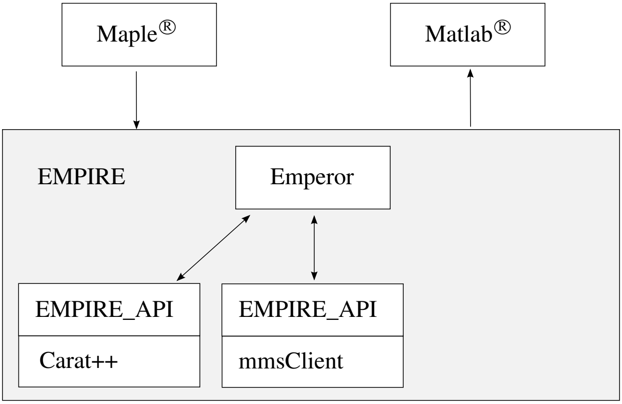


Figure 3.4: Overview of the MMS framework for the assessment of the CSD

3.8.3.1 Maple®

For the calculation of the equilibrium traction forces, the boundary and the initial conditions, equations 3.37, 3.41 and 3.42 are implemented in an own Maple® code.

The analytical derivation of these terms is modeled in terms of the chosen manufactured solution field $\hat{\mathbf{d}}$. Using the chosen initial geometry description and $\hat{\mathbf{d}}$, the kinematic description given in equations 3.4 - 3.7, the Green-Lagrangian strain tensor $\mathbf{E}(\hat{\mathbf{d}})$ in terms of the manufactured solution $\hat{\mathbf{d}}$ is obtained. The linear elastic isotropic material, given in equations 3.8-3.10, leads to the PK2 stress tensor $\mathbf{S}(\hat{\mathbf{d}})$. The stress tensor due to prestress $\mathbf{S}_{ps}$ is added to the stress tensor $\mathbf{S}(\hat{\mathbf{d}})$ due to strains. To convert the available PK2 stress tensors in the desired PK1 stress tensor $\mathbf{P}(\hat{\mathbf{d}})$, the PK2 stresses have to be contracted with the deformation gradient $\mathbf{F}$. To finally calculate $\hat{\mathbf{T}}_s$, the known fields have to be derived as described in equation 3.37.

As $\hat{\mathbf{d}}$ and $\mathbf{P}(\hat{\mathbf{d}})$ are known, the normal $\mathbf{N}^\Gamma$ at the boundary is computed by the initial geometry definition. With this information, all boundary conditions can be computed (cf. equations 3.40-3.42).

3.8.3.2 mmsClient

The mmsClient is a proprietary auxiliary tool to generate the discrete traction loads from $\hat{\mathbf{T}}_s$, which are transferred to Carat++. Additionally, it receives the computed displacement field $\mathbf{d}$ from Carat++ and evaluates its numerical error.

The generated analytical traction force vector $\hat{\mathbf{T}}_s$ and the manufactured solution $\hat{\mathbf{d}}$ from Maple®are stored in a function container of the mmsClient software. Next, the load is sampled point-wise as preparation for the integration and mapping procedures of Emperor.

After solving the benchmark in Carat++, the discrete solution field $\mathbf{d}$ is transferred to the mmsClient. As the manufactured solution field $\hat{\mathbf{d}}$ is also stored in the function container of the mmsClient, the difference between $\mathbf{d}$ and $\hat{\mathbf{d}}$ can be calculated, which represents the field error. To finally get a mesh independent and comparable error information, the difference between $\mathbf{d}$ and $\hat{\mathbf{d}}$ is evaluated on a subset of nodes (cf. figure 3.6), called probes or probe locations. Finally, the representative probe information is stored with double precision in text files for the processing in Matlab®.

3.8.3.3 Emperor

Emperor provides procedures to perform sampling, mapping and integration of fields between the different contributing software in the frame of EMPIRE. Therefore the discrete solutions of equations 5.4 and 5.6 using the Mortar method is performed within Emperor. In the MMS application of Carat++, the same grids are used in the mmsClient and Carat++ to avoid additional mapping errors (cf. chapter 5).
As the sampling and integration procedures are numerical processes, they introduce additional error to the simulation. As similarly stated above for the geometrical approximation, the traction force integration method has to have at least the order of accuracy as the actual discretization of Carat++. As Carat++ is formally second order accurate (cf. section 3.7.4), the force integration must be at least formally second order accurate as well. The used procedures for sampling and integration also use linear shape functions. Therefore, they are formally second order accurate.

3.8.3.4 Matlab®

The software Matlab®is used to finally apply the error norms 2.7 and 2.8 on the resulting field deviations of $\hat{\mathbf{d}}$ from $\mathbf{d}$ at the probe locations. Using these errors and the named error norms, graphs are plotted for the error development in a log-log diagram and a diagram for the observed order of accuracy over grid refinement (e.g., figure 3.11).

3.8.4 Procedure

The procedure applying the MMS for CSD, especially for Carat++, is outlined in the following. The procedure and its balance equation are based on the momentum equilibrium in the initial configuration (cf. equations 3.2, 3.37, 3.41 and 3.42).

1. Creation of a manufactured solution (cf. section 2.3.3) using Maple®
    - Creation of a manufactured field solution $\hat{\mathbf{d}}$
    - Creation of all contributing constants
    - Derivation of the equilibrium traction force $\hat{\mathbf{T}}_s$ using equation 3.39
    - Derivation of the initial condition using equation 3.42
    - Derivation of the boundary conditions using equation 3.41
2. Application of the MMS terms to the code Carat++ by assistance of the mmsClient tool
    - Application of the equilibrium forces $\hat{\mathbf{T}}_s$
    - Application of the initial condition
    - Application of the boundary conditions
3. Performing the simulation in Carat++ with a resulting field $\mathbf{d}$
4. Error evaluation by assistance of the mmsClient tool
    - Sampling of $\mathbf{d}$ at a set of probe locations
    - Error estimation at probes using $\mathbf{d}$ and $\hat{\mathbf{d}}$ in equation 2.7 resp. 2.8
5. Repetition of steps 2-4 with systematic refinement
6. Calculation of the observed order of accuracy p using Matlab®
    - Comparison of the error evaluations at each refinement step
    - Derivation of p with refinement using equation 2.2
7. Comparison of the formal order of accuracy $\hat{p}$ with the observed order of accuracy p
8. Assessment of the code
    - If p matches $\hat{p}$, all tested ordered parts are working as intended
    - If p does not match $\hat{p}$, the error source has to be investigated

For the detailed procedure of refinement for steady state or unsteady analyses, the proposed procedures of section 2.3.3.5 respectively 2.3.3.6 should be considered. The MMS procedure is shown additionally in figure 3.5.

3.8.5 Spatial and Time Resolutions

As already mentioned, the MMS deals with a systematic refinement in space and time. The quantities of grid size and resolution in general is almost arbitrary. The presented numbers are only a suggestion for the reader, where the results shown later are produced.

As stated in sections 2.3.3.5 and 2.3.3.6, the set of sampling points of a single MMS study has to be kept constant. In the context of this work, all nodal and possibly

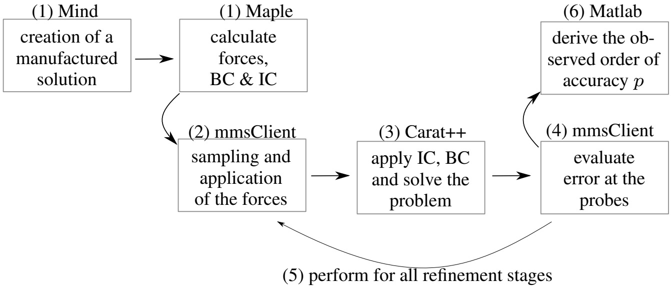


Figure 3.5: The MMS procedure for structural analysis using Carat++

time positions of the coarsest stage are used as sampling points, independently of the individual grid or time resolution.

The coarsest grid, which in parallel defines the sampling points for the complete assessment, is chosen as a quadrilateral grid using 8 by 8 elements. In unsteady solutions 8 timesteps are chosen as coarsest resolution, which in parallel defines the sampling points over time. A sketch of the spatial sampling points in the first stages of grid refinement is illustrated in figure 3.6. The grid is drawn with straight lines, and the probe positions are denoted with crosses.

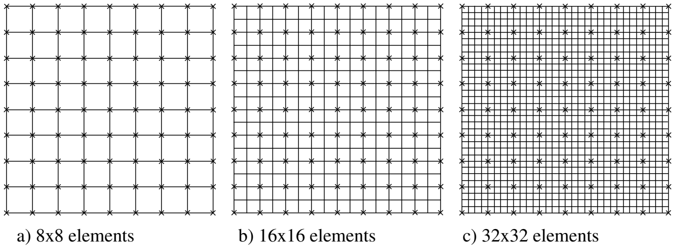

a) 8x8 elements
b) 16x16 elements
c) 32x32 elements

Figure 3.6: Sampling points in space during refinement for the CSD

During refinement, this setup is refined with a factor of $r_s = r_t = r = 2$ with the target to reduce the error with each refinement step by a factor of 4 (cf. section 2.3.3.5 and 2.3.3.6) to observe an order of accuracy of $p = 2$. As shown in the literature, the refinement factor could be reduced to a minimum of $1.1 - 1.2$ for the observation of the order of accuracy. This can be especially important, if one cannot afford further large refinement steps in the asymptotic range of the solution. As already mentioned, in this chapter a constant refinement factor of $r = 2$ is chosen.

3.8.6 Solution Accuracy

Recalling section 2.3.2, the discretization error should be isolated from the total numerical error (cf. figure 2.4). Therefore, the used software Carat++ and all other contributing software (cf. figure 3.5) are using double precision accuracy to minimize the round-off error (cf. figure 3.7). To keep the IICE as small as possible, the solution tolerance in the inf norm of displacements (cf. equation 2.8) of the nonlinear equation system (using a Newton-Raphson procedure) is set to $1 \cdot 10^{-14}$.

3.8.7 Error Map

In this section, an overview of the errors on the CSD simulation is elaborated. On the one hand, occurring errors can be described in a general way. The errors originate from the processes of preprocessing of input data, the solution of the governing equations, and the postprocessing of output data of the simulation. For the application of the present software Carat++, the general error categories and the specific error sources of the software are shown in figure 3.7 on the left-hand side and the right-hand side respectively.

Preprocessing of input data		- Human errors in the preparation of input data - Generation of input data *
Numerical errors due to the solution of the governing equations	Iterative solution of the nonlinear problem	- Discretization error in time - Discretization error in space of the physical equations (momentum, material, kinematics) - Solver for the linear equation system for $\mathbf{d}$ * - Newton-Raphson IICE *
Postprocessing of output data		- Generation of output data * - Human errors in the postprocessing of simulation results

* contains round-off

Figure 3.7: Error map of the presented CSD software Carat++

It is obvious that errors in the generation of input data will influence the total simulation process. This indicates that the generation and application of the MMS procedure (cf. figure 3.5) needs to be at least as accurate as the solution of the simulation itself. In the present case, the actual simulation provides a formal order of accuracy of $p = 2$. Therefore, the generation and application of the MMS forces, the BC, and the IC need to be at least second order accurate as well. Additionally, the output generation and the error evaluation in Matlab needs to be at least equally accurate.

If this is not fulfilled, the total observed order of the simulation will lower to the accuracy of the weakest link of the chain.

3.8.8 Benchmarks

The following benchmark series for CSD is developed as a stairway in complexity. This means, the proposed benchmarks sequence starts with the assessment of restricted functionality of the code and becomes increasingly complex through the benchmark sequence (cf. figure 3.8).

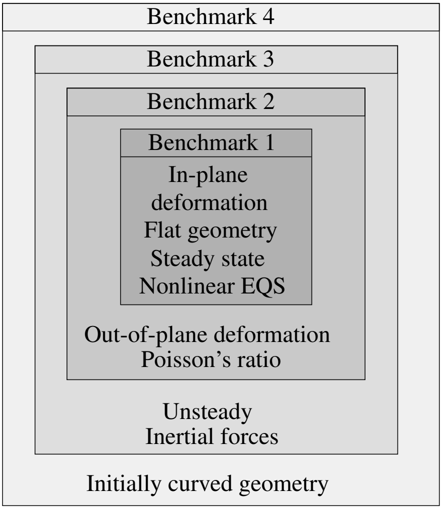


Figure 3.8: Hierarchical benchmark sequence for CSD

All functions and variables of the following examples are listed in a table, so that the reader is able to reconstruct the force term $\hat{\mathbf{Q}}_s$ respectively the traction term $\hat{\mathbf{T}}_s$, the boundary conditions, and the initial conditions. The choice of the BC type (Dirichlet or Neumann) on each edge is left to the reader. Consequently different boundary scenarios can be tested on the same example. All of the input parameters of the benchmarks are defined in the International System of Units (SI) and their derived expressions [81]. Therefore, units for the input numbers of the benchmarks are generally omitted.

Remember: The MMS assessment starts at a preset of governing equations with all its assumptions as a predefined fact. It is not necessary that the used parameters are in the range of applicability of the equilibrium equations from its physical meaning. This means that, e.g., in the first example the membrane has a thickness B of $0.25m$ over a span area of only $1.0m^2$ although this does not make sense in physical applications of the membrane theory model. To get familiar with the elaborated benchmarks, the figures and plots in the first example are explained extensively.

3.8.8.1 Benchmark 1: Plane Membrane, In-Plane Deformation, Steady State

The simplest presented benchmark is a plane rectangular membrane under pure tension in a steady state regime (cf. table 3.1). To keep this first benchmark as simple as possible, the Poisson's ratio ν_s is set to zero to eliminate the Poisson effect. The thickness is set to $B = 0.25m$ and the isotropic prestress is set to $25000N/m^2$. This benchmark assesses the pure normal force action with prestress in the fully geometric nonlinear environment. Figure 3.9 shows the undeformed state of the rectangular membrane with $1.0m \cdot 1.0m$. It is not curved and initially lies in the global x-y plane. The boundaries at $x = 0.0$ and $x = 1.0$ respectively $\theta^1 = 0.0$ and $\theta^1 = 1.0$, are fixed with a support (Dirichlet zero) boundary condition. The boundaries at $y = 0.0$ and $y = 1.0$ respectively $\theta^2 = 0.0$ and $\theta^2 = 1.0$ are fixed in global y and z direction only. Figure 3.10 shows the deformed configuration of the membrane. Along the formula of the desired deformation $d_x = 0.1 \cdot \sin(\theta^1\pi)$, figure 3.10 shows the deformation of the membrane during the simulation. It is an in-plane deformation with a maximum value of $0.10m$ at $\theta^1 = 0.5$ at all θ^2. The systematic refinement during the set of simulations is performed along sections 2.3.3.5 and 3.8.4. Figure 3.11 shows the inf and the E_2 error norms of the different displacement directions $\mathbf{d} = (dx, dy, dz)^T$. The index i references to the infinity, and the index 2 references to the E_2 norm of the individual displacement field error. The negative slope in figure 3.11 in the left panel - with almost a straight line in the more refined area - of the log-log diagram indicates that the error tends towards machine accuracy and therefore, it indicates convergence of the fields. The slope of the error graph is calculated at each refinement step from a pair of errors, using equation 2.2. The pairwise calculated error development is drawn as a graph in figure 3.11 right. The calculated inclination represents the observed order of accuracy p. In order to judge the benchmark execution successful, the observed order of accuracy must reach the formal order of accuracy with refinement. As elaborated in section 3.7.4, the formal order of accuracy of the used software Carat++ is $\hat{p} = 2$.

Table 3.1: Benchmark 1: Overview table

initial configuration	deformation	domain size
$x = \theta^1$	$\hat{d}_x = 0.1 \cdot \theta^1\pi)$	$\theta^1 \in [0, 1]$
$y = \theta^2$	$\hat{d}_y = 0$	$\theta^2 \in [0, 1]$
$z = 0$	$\hat{d}_z = 0$	steady state

material	element properties
$E = 70000$	$B = 0.25$
$\rho = 0$	$S_{ps_{\theta 1}} = 25000$
$\nu_s = 0$	$S_{ps_{\theta 2}} = 25000$

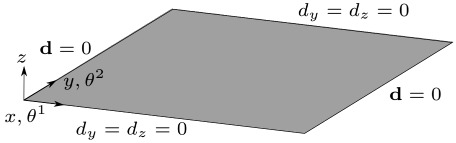


Figure 3.9: Benchmark 1: Initial configuration and boundary conditions

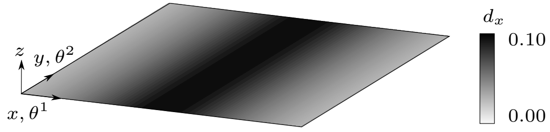


Figure 3.10: Benchmark 1: Deformed configuration

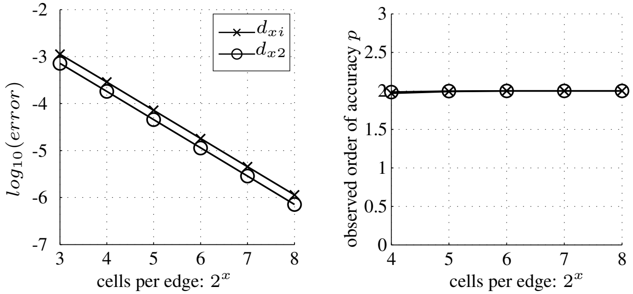


Figure 3.11: Benchmark 1: Spatial convergence plot

As the observed order of accuracy $p \approx 2.0$ matches $\hat{p}$ in both error norms, the execution results in a positive assessment of all ordered functionalities touched by this benchmark. Therefore, these features can be used for further investigations.

3.8.8.2 Benchmark 2: Plane Membrane, Out-of-Plane Deformation, Steady State

Benchmark 2 is a membrane which is deformed out-of-plane. The initial plane is located in the x-y-plane and the deformation is perpendicular on this plane in the direction of the z-axis. The inertial forces are neglected, as the computation is steady state. An overview of the benchmark is presented in table 3.2. As the row of benchmarks is a benchmark stair with increasing complexity, the special focus of the current benchmark is on the additional parts contributing to the simulation compared to the previous ones. This is possible, if the contributing parts/features of the code to the previous benchmark(s) are already positively assessed.

The initial configuration of the membrane is similar to benchmark 1. The difference is that in benchmark 2 all edges are fixed in all directions (compare figures 3.9 and

Table 3.2: Benchmark 2: Overview table

initial configuration	deformation	domain size
$x = \theta^1$	$\hat{d}_x = 0$	$\theta^1 \in [0, 1]$
$y = \theta^2$	$\hat{d}_y = 0$	$\theta^2 \in [0, 1]$
$z = 0$	$\hat{d}_z = 0.25 \cdot \sin(\theta^1 \pi) \cdot \sin(\theta^2 \pi)$	steady state

material	element properties
$E = 1000$	$B = 0.001$
$\rho = 0$	$S_{ps_{\theta 1}} = 5$
$\nu_s = 0.3$	$S_{ps_{\theta 1}} = 5$

3.12). Beside the fact that different numbers are set for the material and the element properties, the Poisson effect now is activated through $\nu_s = 0.3$. The deformation of the membrane is perpendicular to the initial geometry with a peak deformation of $d_z(\theta^1 = \theta^2 = 0.5) = 0.25m$ (cf. figure 3.13).

Compared to the previous benchmark, the second benchmark additionally assesses the geometric transformation through out-of-plane deformation, shear force action, and the full material law with the Poisson effect.
Analogous to the results of benchmark 1, figure 3.14 left panel illustrates the *infinity* and the E_2 error norms of the displacements. The observed order of accuracy p again is calculated using equation 2.2 and is plotted in figure 3.14 right panel. Figure 3.14 presents an observed order of accuracy of $p \approx 2.0$ in the asymptotic range of the solution. In figure 3.14 can be observed that the code is not able to calculate a solution which directly tends to the analytical solution, if the grid is too coarse ($p \ll \hat{p}$). This is only possible, if the simulation is in the asymptotic range of the solution. The asymptotic range of the solution is reached as soon as the observed order of accuracy converged to its asymptotic value.
As stated in section 3.7.4, the formal order of accuracy of the used software Carat++ is $\hat{p} = 2$. As the observed order of accuracy $p \approx 2.0$ matches $\hat{p}$ in both error norms, the execution results in a positive assessment of all ordered functionalities touched by this benchmark. Therefore, these features can be used for further investigations.

3.8.8.3 Benchmark 3: Plane Membrane, Out-of-Plane Deformation, Unsteady

Benchmark 3 is a plane membrane which is deformed out-of-plane from the initial x-y-plane. Inertial forces appear in this benchmark, as the membrane density is existent and the calculation is unsteady. An overview of the benchmark is presented in table 3.3.
The main difference between the previous and the current benchmark is the unsteadiness over time. Therefore, the figures of the initial and the deformed configuration at $t = 1.0s$ (cf. figures 3.15 and 3.16) are similar to the figures 3.12 and 3.13.

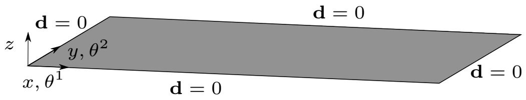


Figure 3.12: Benchmark 2: Initial configuration and boundary conditions

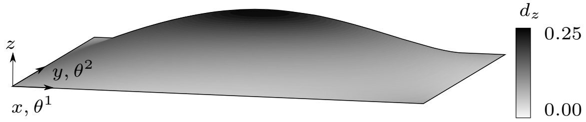


Figure 3.13: Benchmark 2: Deformed configuration

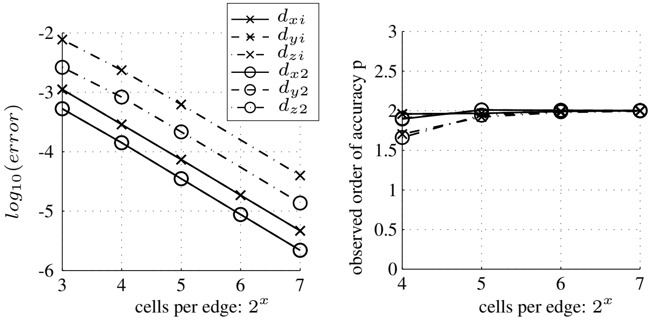


Figure 3.14: Benchmark 2: Spatial convergence plot

By incorporating unsteadiness, two additional things for the assessment strategy have to be decided. As discussed in section 2.3.3.6, there exist different ways for the assessment of unsteady simulations. Table 3.3 exhibits that, a solution field based on a sinus function is chosen in space and time. This indicates that the spatial and the time accuracy is assessed together with the concurrent refinements outlined in section 3.8.4. Figure 3.17 shows the error and the observed order of accuracy plots of the presented benchmark 3. Using a second order accurate method in space and time, the space and time refinement is performed with the same factor of $r = 2$ in space and time. This means exemplarily that 1 timestep is used in the level of 2^3 cells per edge, whereas 4 timesteps are used in the level of 2^5 cells per edge.

In the refined area, the observed order of accuracy reaches $p \approx 2.0$, which matches the formal order of accuracy of $\hat{p} = 2$.

Therefore, the additional mass/inertia contribution and the time discretization are positively assessed and can be used for further investigations.

Table 3.3: Benchmark 3: Overview table

initial configuration	deformation	domain size
$x = \theta^1$	$\hat{d}_x = 0$	$\theta^1 \in [0, 1]$
$y = \theta^2$	$\hat{d}_y = 0$	$\theta^2 \in [0, 1]$
$z = 0$	$\hat{d}_z = 0.25 \sin(\theta^1 \pi) \sin(\theta^2 \pi) \sin(t \frac{\pi}{2})$	$t \in [0, 1]$

material	element properties
$E = 1000$	$B = 0.001$
$\rho = 1000$	$S_{ps_{\theta 1}} = 25$
$\nu = 0.3$	$S_{ps_{\theta 2}} = 25$

3.8.8.4 Benchmark 4: Curved Membrane, Out-of-Plane Deformation, Unsteady

The last elaborated benchmark for the CSD is an initially curved membrane which is deformed out-of-plane in an unsteady calculation (cf. table 3.4).

Table 3.4: Benchmark 4: Overview table

initial configuration	deformation	domain size
$x = \theta^1$	$\hat{d}_x = 0$	$\theta^1 \in [0, 1]$
$y = \theta^2$	$\hat{d}_y = 0$	$\theta^2 \in [0, 1]$
$z = \theta^1 - \theta^1 \theta^1$	$\hat{d}_z = \frac{1}{4} \sin(\theta^1 \pi) \cos(\theta^2 \pi) \sin(\frac{1}{2} \pi t)$	$t \in [0, 2]$

material	element properties
$E = 1000$	$B = 0.001$
$\rho = 1000$	$S_{ps_{\theta 1}} = 25$
$\nu = 0.3$	$S_{ps_{\theta 2}} = 25$

The initial configuration is described with a parabola along θ^1 in z-direction as drawn in figure 3.18. The supports at $\theta^2 = 0$ and $\theta^2 = 1.0$ are only fixed in global y-direction. The deformation on this curved structure is described with a sinus function in space and time. The deformed shape at $t = 1.0s$ is drawn in figure 3.19.

Compared to the previous benchmarks, this benchmark additionally assesses the geometric approximation of the initial configuration and the deformation process with the base of a curved structure. The unsteady assessment is performed using a parallel spatial and time refinement similar to benchmark 3.

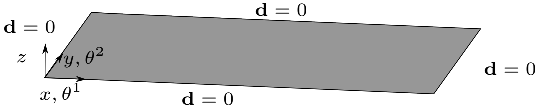


Figure 3.15: Benchmark 3: Initial configuration and boundary conditions

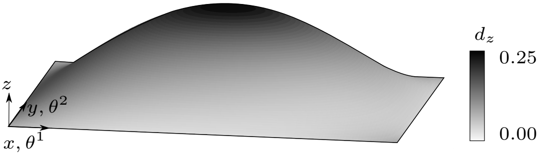


Figure 3.16: Benchmark 3: Deformed configuration at $t = 1.0$

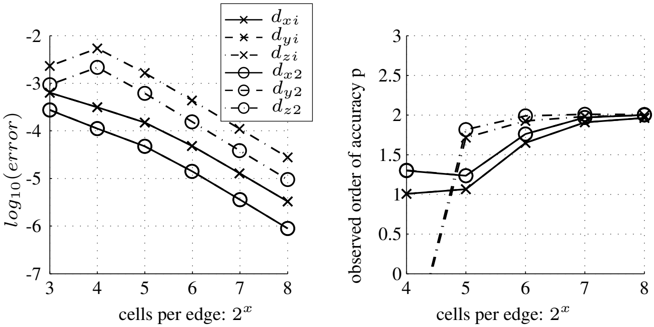


Figure 3.17: Benchmark 3: Spatial and time convergence plot

The error and the observed order of accuracy p development are shown in figure 3.20. In the refined area, the observed order of accuracy reaches $p \approx 2.0$, which matches the formal order of accuracy of $\hat{p} = 2$.

As shown in section 3.7, the geometric approximation of a curved structure is additionally introducing error. This additional error due to geometrical approximation can be seen as a reason, why the errors in benchmark 4 compared to benchmark 3 are larger (compare figures 3.17 and 3.20). Nevertheless, as previously shown, this geometric approximation error also scales with the same formal order of accuracy and should not affect the order of accuracy of the solution. Figure 3.20 depicts in the right panel that the observed order of accuracy reaches $p \approx 2$ in the refined areas, which matches the formal order of accuracy of $\hat{p} = 2$.

Therefore, the additional contribution of the geometrical approximations is positively assessed and can be used for further investigations.

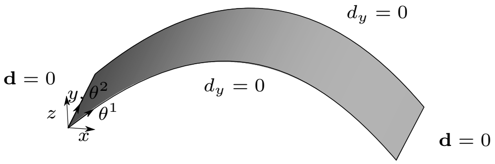


Figure 3.18: Benchmark 4: Initial configuration and boundary conditions

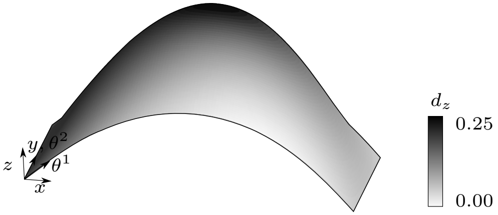


Figure 3.19: Benchmark 4: Deformed configuration at $t = 1.0$

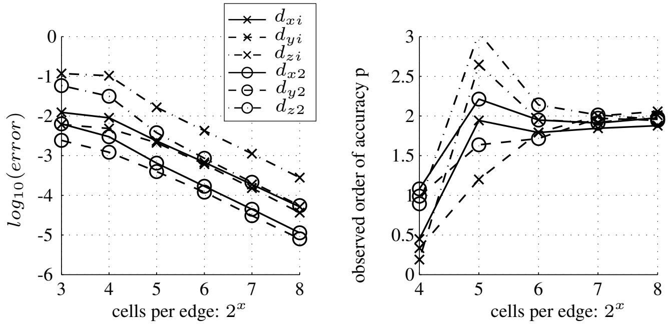


Figure 3.20: Benchmark 4: Spatial and time convergence plot

3.8.9 Statement and Results for the used CSD Environment

Summarizing this chapter and the benchmarks particularly, the code Carat++ is assessed to work as intended. The elaborated benchmark sequence with increasing stages of complexity builds confidence in the functionalities and features of the code. It can be finally determined that the governing equations in the code are solved consistently. The used geometrical nonlinear membrane using linear shape functions and Newmark time integration confirms the formal order of accuracy in space and time with $p \approx \hat{p} = 2$. Consequently all ordered parts of the code are correctly implemented in the software Carat++. Therefore, all assessed parts of the code can be used for further investigations in the context of V&V to finally reach a predictive capability of the software (cf. [9, 57, 75, 79] and chapter 2).

3.9 Adaptation of the CSD Framework

As shown in this chapter, the MMS framework is applied to a particular CSD environment for modeling and simulation of lightweight membrane structures. However, the very general concept of the framework makes it very attractive for the adaptation to other CSD environments or adaptations and extensions of the assessed CSD software.

In the next sections, a few examples of potential adaptations are outlined. The presented examples focus on the required adaptations within the presented MMS environment. The actual modeling and simulation functionalities regarding the solution of the considered field or the internal data handling are not part of the description, as they should be anyway available independent of an MMS assessment.
Furthermore it is obvious, if the environment is changed, the adapted parts need a separate investigation to derive the formal order of accuracy. This is required before the execution of the individual benchmarks, to set the refinement factors correctly (cf. section 2.3.3.6 and 3.8.5).
It is further shown that only moderate changes are necessary to adapt the MMS framework for the assessment of slightly adapted, or even different, CSD software environments. It may be concluded that the MMS framework lives up to its promised flexibility, adaptivity, and modularity.

3.9.1 Adaptation of the PDE Formulation

As described in this chapter, the equilibrium of the structural dynamics is formulated in the so called Total Lagrangian formulation. Consequently the reference geometry information $\mathbf{X}$ during the complete simulation is still the initial geometry and the displacements are projected on the initial geometry. In contrast, the Updated Lagrangian formulation is completely formulated on the current geometry information $\mathbf{x}$. If the MMS framework presented above is supposed to assess a CSD software formulated in Updated Lagrange, the following points have to be adapted:

- The MMS forces have to be adapted to $\hat{\mathbf{t}}_s$ (cf. equation 3.38) instead of $\hat{\mathbf{T}}_s$ in equation 3.39
- The boundary conditions have to be adapted to the current configuration using equations 3.40 instead of equations 3.41.

Those are the only adaptations that have to be conducted in order to ensure a proper performance of all benchmarks after having changed the formulation from Total Lagrangian to Updated Lagrangian.

It is therefore obvious that changing the CSD software to another CSD Total Lagrangian formulated software with the same element needs absolutely no change of the MMS framework.

3.9.2 Adaptation of the Constitutive Equations

Assuming large strains during the deformation of the membrane, one could extend the simulation environment for membranes with a nonlinear material model. As the material is contributing only at the derivation of the traction force $\hat{\mathbf{T}}_s$, the constitutive equations (cf. equations 3.8-3.10), giving the relations between stresses and strains, have to be adapted. This leads to finally different manufactured forces or tractions which have to be applied to the code. Nevertheless, the elaborated benchmarks of the previous chapters can still be used without restriction to assess the environment with the adapted constitutive equations.

3.9.3 Adaptations for a Solid Element

The presented element is a geometrical nonlinear membrane element. Assuming the adaptation to a framework for the assessment of a geometrical nonlinear solid element formulated in the same software, the manufactured body forces $\hat{\mathbf{q}}_s$ respectively $\hat{\mathbf{Q}}_s$ need to be applied to the software. The difference is only in the derivation of these forces. First of all, the regular Lamé parameters need to be used, as the plane stress assumption does not hold for solids. Secondly, the divergence of the stresses outlined in 3.36 respectively 3.37 needs to be fully evaluated. This means that the divergence of the stresses are equal to the covariant derivative of the normal stress, which does not hold for solid elements. Thirdly, if the solid is not able to handle prestress, it can be easily set to zero.

The presented membrane benchmarks therefore can generally be used for the assessment of solids as well. However, the presented benchmarks of this chapter are only partially qualified for the assessment of solids. Therefore, the development of additional benchmarks is recommended, which are adapted to the key features of the element.

CHAPTER 4

Fluid Dynamics

Fluid Dynamics is concerned with the behavior and the flow of fluids in a bounded volume. In the work at hand, a special focus is on the modeling of incompressible turbulent Newtonian fluids based on the Navier-Stokes equations.
The chapter aims at providing an overview of the requirements, the assumptions, and the implementations of the fluid dynamics in the CFD software OpenFOAM®. The notation of the equations generally use vector notations. A detailed explanation of the used operators, such as the gradient and the divergence of a field, can be found in [30]. The notation of the equations partially use the Einstein summation convention. A detailed description can be found, e.g., in [38, 122]. For the application of the MMS in the context of fluid dynamics, a framework is presented. Furthermore, the application of the MMS, the development of a hierarchical benchmark series, and the assessment results are elaborated. Using these results and the derived formal order of accuracy, the CFD code can be assessed for the intended application.

The derivation of the balance equations of the fluid is kept compact. A detailed derivation, containing discussions of different possibilities to model fluid flows using the Finite Volume method can be found in [7, 30, 49, 53, 120, 121].

4.1 General Assumptions

As mentioned in chapter 1, the fluid part of the FSI analysis deals with incompressible and isothermal Newtonian fluids. The simulation can be steady state or unsteady using the Navier-Stokes equations.

For the spatial description of the fluid, the Eulerian and the Arbitrary Eulerian-Lagrangian (ALE) approach is applied. The discretization of the fluid dynamics part is performed using the Finite Volume method.
An overview of the assessed features of the software OpenFOAM®is illustrated in figure 4.1.

CFD - OpenFOAM®
Finite Volume Method
ALE Formulation
Steady state
Unsteady using BDF1, BFD2 and Trapezoidal time discretization rule
Incompressible fluids
Newtonian fluids
URANS turbulence
Grid motion

Figure 4.1: Overview of the CFD software to assess: OpenFOAM®

4.2 Equilibrium

The conservation of mass and the conservation of momentum define the interior equilibrium equations in incompressible fluid dynamics. Assuming $\nu_t = 0$, these equations together are called the incompressible Navier-Stokes equations and are presented in the strong form in vector notation in equations 4.1 and 4.2:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) - \nabla \cdot ((\nu_m + \nu_t) 2\mathbf{D}) + \nabla P = 0 \tag{4.1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{4.2}$$

In equations 4.1 and 4.2, $\mathbf{u}$ is the absolute velocity vector in global directions, ν_m and ν_t is the molecular respectively turbulent viscosity of the fluid, and P is the kinematic pressure defined in equation 4.3:

$$P = \frac{p}{\rho_f} \tag{4.3}$$

p is the pressure, and ρ_f the constant density of the fluid. $\mathbf{D}$ is the symmetric strain rate tensor given in equation 4.12. If $\nu_t \neq 0$, equations 4.1 and 4.2 are called the Unsteady Reynolds-Averaged Navier-Stokes (URANS) equations. The physical interpretation of equation 4.1 is the momentum balance at an infinitesimal small amount of fluid. The first part containing the time derivative corresponds to the mass and is therefore called the inertia force term. The second part is called the convection term, where the fluid mass is convected with its own velocity $\mathbf{u}$. The third part corresponds to the diffusive motions in the fluid; therefore, it is called the diffusion term. The last part is the pressure or pressure gradient term.

For the application of Finite Volume discretization, the momentum and mass conservation, equations 4.1 and 4.2, are solved in an integral sense over a control volume V [30]. The integrals of equation 4.1 and 4.2 therefore result in equations 4.4 and

4.5:

$$\int_V \frac{\partial \mathbf{u}}{\partial t} dV + \int_V \nabla \cdot (\mathbf{u}\mathbf{u}) dV - \int_V \nabla \cdot ((\nu_m + \nu_t) 2\mathbf{D}) dV + \int_V \nabla P dV = 0 \tag{4.4}$$

$$\int_V \nabla \cdot \mathbf{u} dV = 0 \tag{4.5}$$

4.3 Incompressible Fluids

The assumption of incompressibility holds, as the range of velocities of the application is small compared to the speed of sound c. At a temperature of $20^\circ\mathrm{C}$, the speed of sound is $c \approx 343 \frac{m}{s}$ in the air. The ratio of the maximum occurring velocity and the speed of sound gives the Mach number Ma. The incompressibility condition is shown in equation 4.6 [30]:

$$\mathrm{Ma} = \frac{u_{max}}{c} < 0.3 \tag{4.6}$$

The assumption of incompressibility leads to a constant density ρ_f in space and time.

4.4 Conservation of Mass

The differential form of the mass conservation of incompressible flows derived from the Reynolds transport theorem is given by equation 4.7 [6, 20, 30, 49, 53, 120]:

$$\nabla \cdot \mathbf{u} = 0 \tag{4.7}$$

$\mathbf{u}$ describes the velocity vector of the fluid. $\nabla \cdot (\cdot)$ denotes the divergence of $(\cdot)$.

4.5 Conservation of Momentum

The differential form of the momentum conservation of incompressible flows is presented in equation 4.8 respectively 4.9:

$$\frac{d\rho_f \mathbf{u}}{dt} = \nabla \cdot \boldsymbol{\sigma} \tag{4.8}$$

$$\rho_f \frac{\partial \mathbf{u}}{\partial t} + \rho_f \mathbf{u} \cdot \nabla \mathbf{u} = \nabla \cdot \boldsymbol{\sigma} \tag{4.9}$$

$$\tag{4.10}$$

$\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\mathbf{u}$ the absolute fluid velocity, and ρ_f the density of the fluid. $\nabla(\cdot)$ denotes the gradient of $(\cdot)$. Body forces are neglected.

4.6 Constitutive Equations

For the application in wind engineering, the assumption of Newtonian fluids holds. The Cauchy stress tensor $\boldsymbol{\sigma}$, which is the molecular rate of transport of momentum [30], can be expressed as shown in equation 4.11:

$$\boldsymbol{\sigma} = -\left(p + \frac{2}{3}\mu_m \nabla \cdot \mathbf{u}\right)\mathbf{I} + 2\mu_m \mathbf{D} \tag{4.11}$$

p denotes the pressure, $\mu_m = \nu_m \cdot \rho_f$ is the dynamic molecular viscosity, $\mathbf{I}$ is the unity tensor, and $\mathbf{D}$ is the symmetric rate of strain tensor [30] that is defined by equation 4.12:

$$\mathbf{D} = \frac{1}{2}\left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right] \tag{4.12}$$

4.7 Incompressible Navier-Stokes Equations

As a consequence of the assumptions of incompressible and isothermal Newtonian fluids, the conservation of mass (cf. equation 4.7) and momentum (cf. equation 4.9) along with the constitutive equations 4.11 and 4.12 leads to the final form of the governing equations of the incompressible Navier-Stokes equations as shown in equations 4.13 and 4.14:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) - \nabla \cdot (2\nu_m \mathbf{D}) + \nabla P = 0 \tag{4.13}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{4.14}$$

The actual pressure term $\frac{1}{\rho_f}\nabla p$ in equation 4.13 is replaced by the gradient of the kinematic pressure ∇P using equation 4.3.
Each of the four terms in equation 4.13 has a physical meaning. From left to right, this is:

- $\frac{\partial \mathbf{u}}{\partial t}$: the rate of change of $\mathbf{u}$ over time
- $\nabla \cdot (\mathbf{u}\mathbf{u})$: Transport of $\mathbf{u}$ by convection with the velocity $\mathbf{u}$
- $\nabla \cdot (2\nu_m \mathbf{D})$: Transport of $\mathbf{u}$ by molecular diffusion
- Rate of kinematic pressure of P

4.8 Turbulence

The turbulent behavior of fluids is a very complex physical process. The phenomenon and its modeling possibilities are broadly discussed in the literature. A very extensive discussion can be found in [85]. The Boussinesq approximation of turbulent stresses is applied to model turbulence in the context of this work [30, 85, 120]. Following this idea, turbulence effects can be interpreted as viscous stresses on the fluid, analogous to the molecular diffusion. Therefore, the viscous part of the momentum equation is extended by a turbulence viscosity term containing ν_t. The turbulent incompressible Unsteady Reynolds-Averaged Navier-Stokes (URANS) equations are shown in equations 4.15 and 4.16:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u}\mathbf{u}) - \nabla \cdot ((\nu_m + \nu_t) 2\mathbf{D}) + \nabla P = 0 \tag{4.15}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{4.16}$$

The turbulent viscosity ν_t is a product of a subset of equations depending of the chosen turbulence models.

For modeling turbulence effects, there are plenty of options. One way is to separate the models in three main approaches or groups:

- (Unsteady) Reynolds-Averaged Navier-Stokes (RANS or URANS)
- Large-Eddy simulation (LES)
- Direct Numerical simulation (DNS)

A comprehensive discussion of the approaches and their models can be found in [30, 35, 85, 120]. This work concentrates on the turbulence modeling using the RANS/URANS approach, using the so-called $k - \epsilon$ turbulence model to show the general relations between the Navier-Stokes and the turbulence equations and their behavior within an MMS assessment framework.

4.8.1 Turbulence Modeling using the $k - \epsilon$ model

In this section, the standard $k - \epsilon$ turbulence model originally proposed by Launder and Spalding [65] is presented. Alternatives in the context of URANS/RANS, e.g., are the Mixing-length model proposed by Prandtl [85, 86, 120] or the $k - \omega - SST$ model proposed by Menter [69, 85, 120].

As already mentioned above, additional equations must be solved to derive ν_t for the solution of equation 4.15. The turbulent viscosity is computed using equation 4.17:

$$\nu_t = 0.09 \frac{k^2}{\epsilon} \tag{4.17}$$

To solve equation 4.17, the $k - \epsilon$ model solves differential equations shown in equations 4.18 and 4.19 in order to obtain the turbulent kinetic energy k and the dissipation of turbulent kinetic energy ϵ:

$$\frac{\partial k}{\partial t} + \nabla \cdot (\mathbf{u}k) - \nabla \cdot ((\nu_m + \nu_t) \cdot \nabla k) - G_\nu + \epsilon = 0 \tag{4.18}$$

$$\frac{\partial \epsilon}{\partial t} + \nabla \cdot (\mathbf{u}\epsilon) - \nabla \cdot \Big((\nu_m + \frac{\nu_t}{1.3}) \cdot \nabla \epsilon\Big) - 1.44 \frac{\epsilon}{k} G_\nu + 1.92 \frac{\epsilon^2}{k} = 0 \tag{4.19}$$

with

$$G_\nu = 2\nu_t \cdot \mathbf{D} : \mathbf{D} \tag{4.20}$$

Herein, $\mathbf{D}$ is the symmetric rate of strain tensor derived in equation 4.12.
Each of the five terms in both of the presented equations has a physical meaning. From left to right, this is:

- $\frac{\partial(\cdot)}{\partial t}$: the rate of change of $(\cdot)$ over time
- $\nabla \cdot (\mathbf{u}(\cdot))$: Transport of $(\cdot)$ by convection
- $\nabla \cdot ((\nu_m + *) \cdot \nabla(\cdot))$: Transport of $(\cdot)$ by Diffusion
- Rate of Production of $(\cdot)$
- Rate of Destruction of $(\cdot)$

4.9 Discretization

4.9.1 Spatial Discretization

The chosen spatial discretization used in the software OpenFOAM®belongs to the Finite Volume Method (FVM) [30, 49, 120]. Herein, the domain is approximated by a set of non-overlapping finite control volumes V. The initial grid is described by a position vector $\mathbf{X}$.
In the Finite Volume method, the governing equations 4.15 and 4.16 will be integrated over the control volume V, as shown in equations 4.21 and 4.22:

$$\int_V \frac{\partial \mathbf{u}}{\partial t} dV + \int_V \nabla \cdot (\mathbf{u}\mathbf{u})\, dV - \int_V \nabla \cdot ((\nu_m + \nu_t) 2\mathbf{D})\, dV + \int_V \nabla P dV = 0 \tag{4.21}$$

$$\int_V \nabla \cdot \mathbf{u} dV = 0 \tag{4.22}$$

The application of the Finite Volume method on the presented equations in general use the Gauss theorem presented in equation 4.23. Additionally, the Leibniz integration rule, or the Reynolds transport theorem [6, 20, 30, 49, 53, 120] is described in equation 4.24. The results are elaborated for an arbitrary field $(\cdot)$ in equations 4.25 and 4.26. The application for the time derivative part of the Navier-Stokes is presented in equations 4.27-4.29.

$$\int_V \nabla \cdot (\cdot)\, dV = \int_S (\cdot) \cdot \mathbf{n} dS \tag{4.23}$$

$$\frac{d}{dt} \int_{V(t)} (\cdot)\, dV = \int_{V(t)} \left(\frac{d\,(\cdot)}{dt} + \nabla \cdot (\mathbf{u}_g\,(\cdot)) \right) dV \tag{4.24}$$

$$\int_{V(t)} \frac{\partial\,(\cdot)}{\partial t} dV = \frac{d}{dt} \int_{V(t)} (\cdot)\, dV - \int_{V(t)} \nabla \cdot (\mathbf{u}_g\,(\cdot)) dV \tag{4.25}$$

$$\int_{V(t)} \frac{\partial\,(\cdot)}{\partial t} dV = \frac{d}{dt} \int_{V(t)} (\cdot)\, dV - \int_{S(t)} (\cdot)\, \mathbf{u}_g \cdot \mathbf{n} dS \tag{4.26}$$

$$\frac{d}{dt}\int_{V(t)} \mathbf{u} dV = \int_{V(t)} \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u_g u})\right) dV \tag{4.27}$$

$$\frac{d}{dt}\int_{V(t)} \mathbf{u} dV = \int_{V(t)} \frac{\partial \mathbf{u}}{\partial t} dV + \int_{S(t)} (\mathbf{u_g u}) \cdot \mathbf{n} dS \tag{4.28}$$

$$\int_{V(t)} \frac{\partial \mathbf{u}}{\partial t} dV = \frac{d}{dt}\int_{V(t)} \mathbf{u} dV - \int_{S(t)} (\mathbf{u_g u}) \cdot \mathbf{n} dS \tag{4.29}$$

S or $S(t)$ is the surface of the control volume V respectively $V(t)$, and $\mathbf{u}_g$ is the velocity of the moving volume $V(t)$ respectively the grid. Setting $(^\cdot) = 1$, equation 4.26 represents the Geometric Conservation Law (GCL) [20, 28], also known as the Space Conservation Law (SCL) [30]. The GCL must be satisfied for all control volumes and has to be adapted along the time integration method [20, 28, 39, 40, 111]. For simplicity's sake, the time dependent volume $V(t)$ is reduced to V and the time dependent surface $S(t)$ is reduced to S. For a fixed control volume, this means a non-moving grid, $\mathbf{u}_g$ is equal to zero. Using equations 4.23-4.28, equations 4.21 and 4.22 can be rewritten in equations 4.30 and 4.31:

$$\frac{d}{dt}\int_V \mathbf{u} dV + \int_S (\mathbf{u} - \mathbf{u}_g)\mathbf{u} \cdot \mathbf{n} dS - \int_S (\nu_m + \nu_t) 2\mathbf{D} \cdot \mathbf{n} dS + \int_S P\mathbf{n} dS = 0 \tag{4.30}$$

$$\int_S \mathbf{u} \cdot \mathbf{n} dS = 0 \tag{4.31}$$

These equations are also known as the Arbitrary Lagrangian-Eulerian (ALE) description of a fluid with a moving grid [20, 40].
Each term of equation 4.30 becomes approximated in the following sections. For the discrete description of the initial and the deformed geometry ($\mathbf{X}$ respectively $\mathbf{x}$) and the fields $\mathbf{u}$ and P, the same linear shape functions are used. This is called an isoparametric approach, analogous to the CSD in chapter 3.
A function $\phi(\mathbf{x})$ around a fixed reference point $\mathbf{x}_p$ is linearly approximated in space using equation 4.32:

$$\phi(\mathbf{x}) = \phi_p + (\mathbf{x} - \mathbf{x}_p)(\nabla\phi)_p \tag{4.32}$$

with

$$\phi_p = \phi(\mathbf{x}_p) \tag{4.33}$$

The volume integral of $\phi(\mathbf{x})$ over a volume V therefore is shown in equation 4.34:

$$\int_V \phi(\mathbf{x}) dV = \int_V \left[\phi_p + (\mathbf{x} - \mathbf{x}_p)(\nabla\phi)_p\right] dV \tag{4.34}$$

Assuming the point $\mathbf{x}_p$ as the centroid of a control volume by definition, equation 4.34 simplifies to equation 4.35:

$$\int_V \phi(\mathbf{x})dV = \int_V \phi_p dV = \int_V dV \phi_p = V_p \phi_p \tag{4.35}$$

The term $(\mathbf{x} - \mathbf{x}_p)$ cancels out, as its integral gives the distance of $\mathbf{x}_p$ to the centroid of the volume, which is zero by definition. V_p denotes the volume of the control volume.

The surface integral of $\phi(\mathbf{x})$ is approximated with a sum of integrals over all surfaces f using equation 4.36:

$$\int_S \phi(\mathbf{x}) \cdot \mathbf{n} dS = \sum_f \int_f \phi(\mathbf{x}) \cdot \mathbf{n} df = \sum_f \phi_f \cdot \mathbf{S}_f \tag{4.36}$$

with the value of ϕ and the center of the face f

$$\phi_f = \phi(\mathbf{x}_f) \tag{4.37}$$

$\mathbf{S}_f$ is the outward-pointing face area vector of each face f. Equation 4.36 can directly be used to approximate the pressure term of equation 4.31:

$$\int_S P \mathbf{n} dS = \sum_f P_f \mathbf{S}_f \tag{4.38}$$

4.9.1.1 Approximation of the Diffusive Term

The diffusive term of equation 4.30 is approximated using equations 4.32-4.37 and is presented in equation 4.39:

$$\int_S (\nu_m + \nu_t) \nabla \mathbf{u} \cdot \mathbf{n} dS = \sum_f (\nu_m + \nu_t)_f (2\mathbf{D})_f \cdot \mathbf{S}_f \tag{4.39}$$

with the linear approximation of the face gradient of ϕ

$$(\nabla \phi)_f = \lambda (\nabla \phi)_p + (1 - \lambda)(\nabla \phi)_n \tag{4.40}$$

with the neighbor point n and the linear interpolation factor presented in equation 4.41:

$$\lambda = \frac{\mathbf{x}_f - \mathbf{x}_p}{\mathbf{x}_n - \mathbf{x}_p} \tag{4.41}$$

More details about the approximations and the special topics of non-orthogonal treatment is discussed in [30, 49, 53, 120].

4.9.1.2 Approximation of the Convective Term

The convective term of equation 4.30 is approximated using equations 4.32-4.37 and is presented in equation 4.42:

$$\int_S (\mathbf{u} - \mathbf{u}_g)\mathbf{u} \cdot \mathbf{n} dS = \sum_f (\mathbf{u} - \mathbf{u}_g)_f \mathbf{u}_f \cdot \mathbf{S}_f \tag{4.42}$$

A linear approximation for the face value of ϕ leads to the Central Differencing (CD) scheme described in equation 4.43:

$$\phi_f = \lambda\phi_p + (1 - \lambda)\phi_n \tag{4.43}$$

As CD for the approximation of the convective term may result to unphysical oscillations in the solution for convection-dominated problems, the solution in general is not bounded [44, 49, 82].

To guarantee the boundedness of the solution, a numerical diffusion is introduced at the expense of accuracy. One example is the Upwind Differencing (UD) scheme. Herein, the face interpolation of ϕ is approximated using equation 4.44:

$$\phi_f = \begin{cases} \phi_f = \phi_p, & \text{for} \quad (\mathbf{u} - \mathbf{u}_g)_f \cdot \mathbf{S} \geq 0 \\ \phi_f = \phi_n, & \text{for} \quad (\mathbf{u} - \mathbf{u}_g)_f \cdot \mathbf{S} < 0 \end{cases} . \tag{4.44}$$

As the accuracy of UD is mostly not seen as sufficient, there are a lot of intermediate schemes developed that can be interpreted as a mix between UD and CD, which are more accurate then UD, but still bounded. A broad overview of possible convection schemes is given and discussed in [49, 120].

4.9.2 Time Discretization

For the time discretization, the continuous time period $[t_0, T]$ is split into m discrete time increments with a size of Δt. The current time t_m therefore is described with equation 4.45:

$$t_m = t_0 + \Delta t \cdot m \tag{4.45}$$

For the discretization of the time derivative term of equation 4.30, two implicit Backward Differencing Formula (BDF) schemes and the trapezoidal rule are presented briefly. The BDF1 is shown in equation 4.46 and the BDF2 is shown in equation 4.47 for the time step $m+1$. Furthermore, the trapezoidal rule, also called the Crank-Nicolson scheme is presented in equation 4.48.

$$\frac{d}{dt}\int_V \mathbf{u} dV = \frac{d}{dt}(V_p\mathbf{u}) = \frac{1}{\Delta t}\left(V_p^{m+1}\mathbf{u}_p^{m+1} - V_p^m\mathbf{u}_p^m\right) \tag{4.46}$$

$$\frac{d}{dt}\int_V \mathbf{u} dV = \frac{d}{dt}(V_p\mathbf{u}) = \frac{1}{2\Delta t}\left(3V_p^{m+1}\mathbf{u}_p^{m+1} - 4V_p^m\mathbf{u}_p^m + V_p^{m-1}\mathbf{u}_p^{m-1}\right) \tag{4.47}$$

$$\frac{d}{dt}\int_V \mathbf{u} dV = \frac{d}{dt}(V_p\mathbf{u}) = \frac{1}{2\Delta t}\left(V_p^{m+1}\mathbf{u}_p^{m+1} + V_p^m\mathbf{u}_p^m\right) \tag{4.48}$$

All of the other terms of the equilibria in equations 4.30 and 4.31 are determined in the time step level $m + 1$ as all of the used schemes are implicit time integration schemes.

4.9.3 Grid Motion

The motion of the grid is defined by the solution of an auxiliary equation with the input of the displacement of the boundary. An extensive discussion of the possibilities and the solution of the grid motion can be found in [50]. It is very common to describe the grid motion with the spring analogy. The grid points are the joints and the edges of the cells can be interpreted as springs [50]. Therefore, a steady Laplacian equation is set for the grid motion as shown in equation 4.49:

$$\nabla \cdot (\Gamma_g \nabla \mathbf{d}_g) = 0 \tag{4.49}$$

Γ_g is the diffusivity or the stiffness of the grid. In the discretized form, equation 4.49 transforms to 4.50:

$$\int_V \nabla \cdot (\Gamma_g \nabla \mathbf{d}_g) \, dV = \int_S (\Gamma_g \nabla \mathbf{d}_g) \cdot \mathbf{n} dS = 0 \tag{4.50}$$

As $\mathbf{d}_g$ is the grid displacement field, it directly defines the current position of a domain point $\mathbf{x} = \mathbf{X} + \mathbf{d}_g$ on the basis of the initial configuration $\mathbf{X}$. The grid velocity $\mathbf{u}_g$ is the derivative of the grid displacement $\mathbf{d}_g$. The grid velocity $\mathbf{u}_g$ has to be discretized along the chosen time discretization method to fulfill the above mentioned SCL. For the given time discretization schemes, the grid velocities are calculated using equations 4.51-4.53 for the BDF respectively the CN:

$$\mathbf{u}_g^{m+1}|_{BDF1} = \frac{d\mathbf{d}_g}{dt}|_{BDF1} = \frac{\mathbf{d}_g^{m+1} - \mathbf{d}_g^m}{\Delta t} \tag{4.51}$$

$$\mathbf{u}_g^{m+1}|_{BDF2} = \frac{d\mathbf{d}_g}{dt}|_{BDF2} = \frac{3\mathbf{d}_g^{m+1} - 4\mathbf{d}_g^m + \mathbf{d}_g^{m-1}}{2\Delta t} \tag{4.52}$$

$$\mathbf{u}_g^{m+1}|_{CN} = \frac{d\mathbf{d}_g}{dt}|_{CN} = \frac{\mathbf{d}_g^{m+1} + \mathbf{d}_g^m}{2\Delta t} \tag{4.53}$$

4.10 Solution Process

4.10.1 Boundary Conditions

As the discretized domain is limited in space, boundary conditions (BC) for all primary variables in general are set. This work concentrates on the numerical boundary conditions and does not regard physical boundary conditions like no-slip boundaries (cf. [30, 49, 120]). One type of numerical BC is a Dirichlet condition, which prescribes the value of a variable at the boundary. The other type is a Neumann condition, which prescribes the gradient of the variable in the normal direction of the boundary. The consistent implementation of the Dirichlet and the Neumann BC and their influence on the individual parts of the governing equations is shown in [49, 53].

4.10.2 Interior Equations

As the solution of the governing equations is complicated by the lack of an independent equation of the pressure, the so-called pressure-velocity coupling is constructed using equations 4.1 and 4.2. The pressure-velocity coupling is performed using the projection method originally developed by Chorin and Temam [14, 15, 110]. The projection results in the so-called Poisson equation to compute the pressure. If it is derived theoretically, it results to the form in equation 4.54 [30].

$$\nabla \cdot \nabla P = -\nabla \cdot \nabla \cdot (\mathbf{uu}) \tag{4.54}$$

Famous schemes for the pressure velocity coupling projection methods are the steady and unsteady SIMPLE scheme [83, 120], the PISO [47, 48], and their derived schemes. In the Finite Volume application, a Poisson equation similar to the presented equation 4.54 is derived. To maintain consistency and stability among the used approximations, it is recommended to derive the pressure equation directly from the discretized momentum and continuity equations rather than to approximate equation 4.54 [30, 49, 53]. A consistent derivation of an equation for the pressure P is shown in detail in [30, 49].

4.11 Formal Order of Accuracy

The discretization method, the chosen spatial and time approximations, the numerical integration, and the solution process define the formal order of accuracy [30, 53, 120]. The formal order of accuracy $\hat{p}$ of an approximated expression can be determined 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. The formal order can be reached as long as no other errors with smaller convergence rate occur. A detailed discussion can be found in [30, 53].

4.11.1 Taylor Series Expansion of ϕ

The basis of an assessment of the formal order of convergence $\hat{p}$ of a variable ϕ is the comparison of its Taylor series expansion with its approximation. The Taylor series of ϕ is given over time and space in equations 4.55 and 4.56, respectively. The spatial Taylor series expansion is simplified to one dimension.

$$\begin{aligned} \phi(t+\Delta t) =& \phi(t^{m+1}) = \phi^m + \Delta t \left(\frac{\partial \phi}{\partial t}\right)^m + \frac{\Delta t^2}{2}\left(\frac{\partial^2 \phi}{\partial t^2}\right)^m + \\ & \frac{\Delta t^3}{3!}\left(\frac{\partial^3 \phi}{\partial t^3}\right)^m + ... + \frac{\Delta t^n}{n!}\left(\frac{\partial^n \phi}{\partial t^n}\right)^m \end{aligned} \tag{4.55}$$

$$\begin{aligned} \phi(x) =& \phi_p + (x-x_p)\left(\frac{\partial \phi}{\partial x}\right)_p + \frac{(x-x_p)^2}{2}\left(\frac{\partial^2 \phi}{\partial x^2}\right)_p + \\ & \frac{(x-x_p)^3}{3!}\left(\frac{\partial^3 \phi}{\partial x^3}\right)_p + ... + \frac{(x-x_p)^n}{n!}\left(\frac{\partial^n \phi}{\partial x^n}\right)_p \end{aligned} \tag{4.56}$$

4.11.2 Spatial Discretization

Comparing the used approximation of $\phi(x)$ in equation 4.32 with its Taylor series in equation 4.56, the error E of the approximation can directly be derived in a one dimensional view in equation 4.58:

$$E = \frac{(x-x_p)^2}{2}\left(\frac{\partial^2\phi}{\partial x^2}\right)_p + \frac{(x-x_p)^3}{3!}\left(\frac{\partial^3\phi}{\partial x^3}\right)_p + ... + \frac{(x-x_p)^n}{n!}\left(\frac{\partial^n\phi}{\partial x^n}\right)_p \tag{4.57}$$

$$E = \mathcal{O}((x-x_p)^2) \tag{4.58}$$

The result of equation 4.58 represents combined with equation 2.1 a formal order of accuracy in space of $\hat{p}_s = 2$.

The volume integral of $\phi(\mathbf{x})$ using the second order approximation of $\phi(\mathbf{x})$ is given in equation 4.59:

$$\begin{aligned}\int_V \phi(\mathbf{x})dV = \int_V \phi_p + (\mathbf{x}-\mathbf{x}_p)(\nabla\phi)_p dV = \\ \int_V \phi_p dV + \int_V (\mathbf{x}-\mathbf{x}_p)(\nabla\phi)_p dV\end{aligned} \tag{4.59}$$

By definition of the used code OpenFOAM®, the variables are located at the cell centroids. For a centroid of a volume V, equation 4.60 always holds:

$$\int_V (\mathbf{x}-\mathbf{x}_p)dV = 0 \tag{4.60}$$

Therefore, the second order accurate volume integral simplifies to the form of equation 4.35, denoted by equation 4.61:

$$\int_V \phi(\mathbf{x})dV = \int_V \phi_p dV = V_p\phi_p \tag{4.61}$$

The surface integral of ϕ at a surface f with the surface center value ϕ_f follows the same way as the volume integral using equation 4.60. Therefore, it simplifies in a second order accurate way in equation 4.62.

$$\int_f \phi(\mathbf{x})\cdot\mathbf{n}df = \phi_f\cdot\mathbf{S} \tag{4.62}$$

4.11.3 Time Discretization

As given above, the Taylor series for a field ϕ over time is defined for a time step m and $m-1$ with the base of time step $m+1$ in equation 4.63 respectively 4.64:

$$\phi(t^m) = \phi^{m+1} - \Delta t \left(\frac{\partial \phi}{\partial t}\right)^{m+1} + \frac{\Delta t^2}{2}\left(\frac{\partial^2 \phi}{\partial t^2}\right)^{m+1} - \frac{\Delta t^3}{3!}\left(\frac{\partial^3 \phi}{\partial t^3}\right)^{m+1} + ... + \frac{\Delta t^n}{n!}\left(\frac{\partial^n \phi}{\partial t^n}\right)^{m+1} \tag{4.63}$$

$$\phi(t^{m-1}) = \phi^{m+1} - 2\Delta t \left(\frac{\partial \phi}{\partial t}\right)^{m+1} + \frac{2^2\Delta t^2}{2}\left(\frac{\partial^2 \phi}{\partial t^2}\right)^{m+1} - \frac{2^3\Delta t^3}{3!}\left(\frac{\partial^3 \phi}{\partial t^3}\right)^{m+1} + ... + \frac{2^n\Delta t^n}{n!}\left(\frac{\partial^n \phi}{\partial t^n}\right)^{m+1} \tag{4.64}$$

Comparing the used approximation of the time derivative of ϕ with the Taylor series of equations 4.63 and 4.64, the error of the time derivative approximation can be estimated for the three time discretization schemes in equation 4.65-4.67.

$$E_{BDF1} = \frac{\Delta t}{2}\left(\frac{\partial^2 \phi}{\partial t^2}\right)^{m+1} + \mathcal{O}(\Delta t^2) = \mathcal{O}(\Delta t) \tag{4.65}$$

$$E_{BDF2} = \Delta t^2 \left(\frac{\partial^3 \phi}{\partial t^3}\right)^{m+1} + \mathcal{O}(\Delta t^3) = \mathcal{O}(\Delta t^2) \tag{4.66}$$

$$E_{CN} = \frac{\Delta t^2}{2}\left(\frac{\partial^2 \phi}{\partial t^2}\right)^{m+1} + \mathcal{O}(\Delta t^3) = \mathcal{O}(\Delta t^2) \tag{4.67}$$

The solution process of the time integration using the projection method in the unsteady SIMPLE or the PISO procedure needs to be investigated as well. Following the error analysis of the PISO algorithm, the time error introduced by a partitioned solution process can be evaluated by the so-called one-step error analysis as described in [37, 47, 48, 55, 67]. Therefore the order of the introduced error reduces linearly with the number of iterations starting at $\mathcal{O}(\Delta t)$. One iteration step reduces the error to a level of $\mathcal{O}(\Delta t^2)$. This holds analogously for the unsteady SIMPLE procedure. This means, the projection methods are preserving the order of accuracy of the time discretization itself, shown in equations 4.65-4.67 [30, 53].

A closer look at the implementation of the BDF2 scheme exhibits a weak point of multistep schemes. The BDF2 needs the field informations of two recent time steps, though this information is not available at the begin of a simulation. The current implementation therefore uses for the very first time step a BDF1 discretization, which unfortunately lowers the complete order of the implemented BDF2 to $\hat{p}_t^{BDF2} = 1$.

4.11.4 Resulting Formal Order of Convergence

The present code of OpenFOAM®uses linear shape functions for the approximation in space (cf. equations 4.32) using hexahedral volumes. For the numerical integration, second order still holds using the discussed surface and volume of equations 4.35 and 4.36. The formal order of spatial discretization lowers to first order at skew grids [53].
The BDF1 time integration scheme formally shows first order, and the BDF2 and the CN schemes formally show second order of accuracy with the drawback of unboundedness of the solution. The currently implemented starting condition of the BDF2 scheme lowers its formal accuracy to $\hat{p}_t^{BDF2} = 1$.
Therefore, the formal time and spatial order of accuracy ($\hat{p}_t$ respectively $\hat{p}_s$) of the code is given in equations 4.68-4.72:

$$\hat{p}_s^{rectangular} = 2 \tag{4.68}$$

$$\hat{p}_s^{skew} = 1 \tag{4.69}$$

$$\hat{p}_t^{BDF1} = 1 \tag{4.70}$$

$$\hat{p}_t^{BDF2} = 1 \tag{4.71}$$

$$\hat{p}_t^{CN} = 2 \tag{4.72}$$

4.12 Application of the MMS

4.12.1 Equilibrium Source Terms

The equilibrium sources $\hat{\mathbf{s}}_f$ required to reach the manufactured fluid velocities $\mathbf{u} = \hat{\mathbf{u}}$ and pressure $P = \hat{P}$ in the context of the MMS are elaborated on throughout this section. Assuming a laminar flow, equations 4.73 and 4.74 define the set of sources for the momentum $\hat{\mathbf{s}}_f^{mom}$ and the mass conservation $\hat{\mathbf{s}}_f^{mass}$ with $\nu_t = 0$:

$$\hat{\mathbf{s}}_f^{mom} = \frac{\partial \hat{\mathbf{u}}}{\partial t} + \nabla \cdot (\hat{\mathbf{u}}\hat{\mathbf{u}}) - \nabla \cdot ((\nu_m + \nu_t) 2\hat{\mathbf{D}}) + \nabla \hat{P} \tag{4.73}$$

$$\hat{\mathbf{s}}_f^{mass} = \nabla \cdot \hat{\mathbf{u}} \tag{4.74}$$

$\hat{\mathbf{s}}_f^{mom}$ represents the source term for the momentum conservation in equation 4.21 and $\hat{\mathbf{s}}_f^{mass}$ represents the source term for the mass conservation in equation 4.22. If the code uses projection methods for the solution respectively the time advancement of the Navier-Stokes equations, the actual mass conservation is not solved directly. But as it is the basis PDE for the Poisson equation in the projection, $\hat{\mathbf{s}}_f^{mass}$ represents the source for the Poisson equation 4.54 as well.

Assuming a turbulent MMS flow using the presented $k-\epsilon$ RANS turbulence model, k, ϵ, and ν_t are variable fields of the simulation as well. Therefore, manufactured solutions have to be created and defined to solve the equation for the turbulent viscosity (cf. equation 4.17) shown in equation 4.75:

$$\hat{\nu}_t = 0.09 \frac{\hat{k}^2}{\hat{\epsilon}} \tag{4.75}$$

As equation 4.75 defines the relation between k, ϵ, and ν_t, manufactured solutions have to be defined for two of them such that the third one can be directly derived from 4.75 to avoid additional source terms for equation 4.17 respectively 4.75. Using the manufactured solutions of the variables, the source terms of the turbulence equations (cf. equations 4.18 and 4.19) $\hat{\mathbf{s}}^k$ and $\hat{\mathbf{s}}^\epsilon$ are derived using equations 4.76 - 4.79

$$\hat{\mathbf{s}}_f^k = \frac{\partial \hat{k}}{\partial t} + \nabla \cdot (\hat{\mathbf{u}}\hat{k}) - \nabla \cdot \Big((\nu_m + \nu_t) \cdot \nabla \hat{k}\Big) - \hat{G}_\nu + \hat{\epsilon} \quad (4.76)$$

$$\hat{\mathbf{s}}_f^\epsilon = \frac{\partial \hat{\epsilon}}{\partial t} + \nabla \cdot (\hat{\mathbf{u}}\hat{\epsilon}) - \nabla \cdot \Big((\nu_m + \frac{\nu_t}{1.3}) \cdot \nabla \hat{\epsilon}\Big) - 1.44\frac{\hat{\epsilon}}{\hat{k}}\hat{G}_\nu + 1.92\frac{\hat{\epsilon}^2}{\hat{k}} \quad (4.77)$$

with equations 4.75, 4.78, and 4.79:

$$\hat{G}_\nu = 2\nu_t \cdot \hat{\mathbf{D}} : \hat{\mathbf{D}} \quad (4.78)$$

$$\hat{\mathbf{D}} = \frac{1}{2}\left[\nabla \hat{\mathbf{u}} + (\nabla \hat{\mathbf{u}})^T\right] \quad (4.79)$$

It is emphasized again that the calculated sources $\hat{\mathbf{s}}_f$ represent the equilibrium terms of the problem for the created or given manufactured fields.
These sources $\hat{\mathbf{s}}_f$ therefore have to be applied to the code indifferently of the discretization method of the implemented method for the pressure velocity coupling. All developed equilibria are based on the strong form equilibrium (equations 4.15 and 4.16 and possibly 4.18 respectively 4.19). Furthermore, the source terms are independent of the grid motion, as the grid motion is just a product of discretization and does not exist in the strong formulation of the momentum equation. Thus, the generality of the method for all kind of fluid treatment once more is evident. The grid motion can be seen just as an auxiliary condition, as long as it should not be assessed separately. If the grid motion algorithms should be assessed by the MMS as well, a manufactured solution $\hat{\mathbf{d}}_g$ has to be introduced. The source term for the grid motion $\mathbf{s}_f^d$ generates with the base of equation 4.49 and is outlined in equation 4.80:

$$\hat{\mathbf{s}}_f^{d_g} = \nabla \cdot \Big(\Gamma_g \nabla \hat{\mathbf{d}}_g\Big) \quad (4.80)$$

4.12.2 Boundary and Initial Conditions

The Dirichlet and Neumann boundary conditions (BC) are determined by equations 4.81 respectively 4.82 (cf. section 2.3.3.4). In steady state simulations, the initial conditions (IC) are set to a non-balanced state (e.g., to zero). In transient problems the IC must be set to the target value of the variable. Therefore, at $t = t_0$ the initial fields match the manufactured solution fields.

$$\phi^\gamma = \hat{\phi}^\gamma \quad (4.81)$$

$$\left(\frac{\partial \phi}{\partial \mathbf{x}} \cdot \mathbf{n}\right)^\gamma = \left(\frac{\partial \hat{\phi}}{\partial \mathbf{x}} \cdot \hat{\mathbf{n}}\right)^\gamma \quad (4.82)$$

$$\phi(\mathbf{x}, t = t_0) = \hat{\phi}(\mathbf{x}, t = t_0) \quad (4.83)$$

4.12.3 Framework

The framework to assess the CFD simulation code OpenFOAM®is presented in this section. In total, three software components are used in this framework:

- OpenFOAM®
- Maple®
- Matlab®

An overview of the framework is illustrated in figure 4.2. The procedure and the sequence of the program executions is shown in figure 4.3.

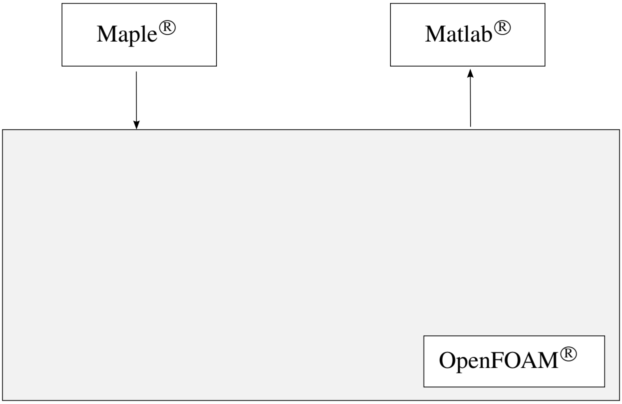


Figure 4.2: Overview of the MMS framework for the assessment of the CFD

4.12.3.1 Maple®

For the calculation of the source terms $\hat{\mathbf{s}}_f$, the boundary and the initial conditions elaborated in equations 4.73-4.83 are implemented in a Maple®code.
The analytical derivation of the manufactured solutions is modeled in terms of the chosen manufactured solution fields $\hat{\phi}$. In detail, $\hat{\phi}$ represents the fields of $\hat{P}$, $\hat{\mathbf{u}}$, and possibly $\hat{\nu}_t$, $\hat{k}$, and $\hat{\epsilon}$. The normal vector of the boundary $\hat{\mathbf{n}}^\gamma$ can be derived from the initial definition of the domain boundary $\mathbf{X}$. If the grid boundary is moving, the geometry and the normal vector are unsteady over time as well. Therefore, the normal vector $\hat{\mathbf{n}}^\gamma$ can be derived from the current position of the boundary of $\mathbf{x}^\gamma$.

4.12.3.2 OpenFOAM®

Beside the already presented implementations of the solution concept of the incompressible Navier-Stokes equations, a few operations for the application of the MMS

are implemented in OpenFOAM®. Therefore, the derived source terms $\hat{\mathbf{s}}_f$ calculated in Maple® are applied directly in the code and added to the individual balance equation. As the balance equations are formulated as integral equations, the sources have to be integrated second order accurate along equation 4.61 as well.
After solving the benchmark in OpenFOAM®, the discrete solution fields ϕ are compared to the also stored manufactured solutions $\hat{\phi}$. To finally create a grid independent and comparable error information, the difference between ϕ and $\hat{\phi}$ is evaluated on a subset of points, called probes or probe locations.

4.12.3.3 Matlab®

The software Matlab® is used to finally apply the error norms 2.7 and 2.8 on the resulting field deviations of $\hat{\phi}$ from ϕ at the probe locations. Using these errors and the named error norms, graphs are plotted for the error development in a log-log diagram and a diagram for the observed order of accuracy (e.g., figure 4.14).

4.12.4 Procedure

The procedure applying the MMS for CFD, in particular for OpenFOAM®, is outlined in the following. The procedure and its balance equation are based on the incompressible Navier-Stokes equations 4.15-4.20.

1. Creation of manufactured solutions (cf. section 2.3.3) using Maple®
    - Creation of the manufactured field solutions $\hat{\phi}$
    - Creation of all contributing constants
    - Derivation of the source term fields $\hat{\mathbf{s}}_f$ using equations 4.73-4.79
    - Derivation of the initial conditions using equation 4.83
    - Derivation of the boundary conditions using equations 4.81 and 4.82
2. Application of the MMS terms to the code OpenFOAM®
    - Application of the source term fields $\hat{\mathbf{s}}_f$ to the right hand side of the individual equation
    - Application of the initial condition
    - Application of the boundary conditions
3. Performing the simulation in OpenFOAM® with the resulting fields ϕ
4. Error evaluation
    - Sampling of the fields ϕ at a set of probe locations
    - Field error estimation at probes using ϕ and $\hat{\phi}$
5. Repetition of steps 2-4 with systematical refinement
6. Calculation of the observed order of accuracy p using Matlab®
    - Error norm application to the field errors using equation 2.7 resp. 2.8
    - Comparison of the error evaluations at each refinement step
    - Derivation of p with refinement using equation 2.13

7. Comparison of the formal order of convergence $\hat{p}$ to the observed order of convergence p

8. Assessment of the code

   - If p matches $\hat{p}$, all ordered functionalities are working as intended
   - If p does not match $\hat{p}$, the error source has to be investigated

For the detailed procedure of refinement for steady state or unsteady analyses, the proposed procedure of section 2.3.3.5 respectively 2.3.3.6 should be considered.

The MMS procedure is additionally shown in figure 4.3.

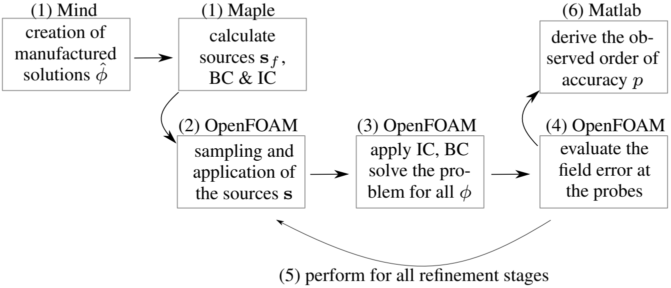


Figure 4.3: The MMS procedure for fluid dynamics using OpenFOAM®

4.12.5 Spatial and Time Resolutions

As already mentioned in chapter 2, the MMS deals with systematic refinement in space and time. The quantities of grid size and resolution in general is almost arbitrary. The depicted numbers are only a suggestion for the reader, where the later shown results are produced. For the reason of limited computational resources, the benchmarks for CFD generally are given for two dimensional computations.

As elaborated in sections 2.3.3.5 and 2.3.3.6, the set of sampling points of a single MMS assessment has to be kept constant. In the context of this work, all cell midpoints and time positions of the coarsest stage are used as sampling points, independently of the individual grid or time resolution.
The coarsest grids in CFD are chosen as quadrilateral grids using 8 by 8 elements. Their midpoints define the 64 sampling points of each simulation during refinement In unsteady simulations 8 timesteps are chosen as the coarsest resolution, which in parallel defines the sampling points over time. A sketch of the sampling points in the first stages of grid refinement is illustrated in figure 4.4. The grid is drawn with straight lines, and the probe positions are denoted with crosses.
Compared to the CSD in chapter 3, the setup of the refinement factors are more complex. In chapter 3, the formal order of accuracy in space and time always was

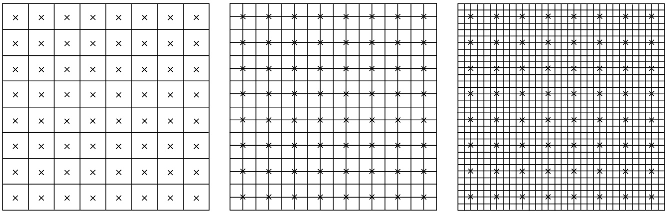

a) 8x8 elements b) 16x16 elements c) 32x32 elements

Figure 4.4: Sampling points in space during refinement for the CFD

$\hat{p} = 2$. Therefore, the space and time refinement were always chosen with the same factor.

Having different formal orders of accuracy, different refinement factors r must be set to compensate this lack. Therefore, the defined target is to reduce the error within each refinement step by a factor of 4 (cf. section 2.3.3.5 and 2.3.3.6), which is the case for second order accuracy using $r = 2$. In the present case, using, e.g., a BDF1 time discretization and a rectangular grid, the spatial formal order of accuracy is $\hat{p}_s^{rectangular} = 2$, and the time formal order of accuracy is $p_t^{BDF1} = 1$. To assess both accuracies together, the refinement over time has to be adapted along its formal order of accuracy to a $r_t = 4$. If the result shows a $p = 2$, the second order of accuracy in space and the first order of accuracy over time is successfully assessed [57].

4.12.6 Solution Accuracy

Recalling section 2.3.2, the discretization error should be isolated from the total numerical error (cf. figure 2.4). Therefore, the used software OpenFOAM®and all other contributing software (cf. figure 4.3) are using double precision accuracy to minimize the round-off error. To keep the IICE as small as possible, the solution tolerance in a non-normalized version of the E_2 norm of velocities and pressures (cf. equation 2.8) of the unsteady SIMPLE projection method is set to $1.0 \cdot 10^{-11}$.

4.12.7 Error Map

In this section, an overview of the errors on the CFD simulation is outlined. The errors can be described in a general way [73]. The errors arise in the processes of preprocessing of input data, the errors due the solution of the governing equations, and the postprocessing of output data of the simulation. For the application of the present software OpenFOAM®, the general sources of error and the specific error sources of the software are shown in figure 4.5 left hand side and the right hand side respectively. It is obvious that errors in the generation of input data will influence the total simulation process. This indicates that the generation and application of the MMS procedure (cf. figure 4.3) needs to be at least as accurate as the solution of the

Preprocessing of input data		- Human errors in the preparation of input data - Generation of input data *
Numerical errors due to the solution of the governing equations	Iterative solution of the nonlinear problem	- Discretization in time - Discretization of the grid motion - Solver for the linear equation system $\mathbf{d}_g$ * - Discretization of the momentum equation - Discretization of the mass equation (Poisson's equation) * - Solver for the linear equation system for $\mathbf{u}$ and P * - Unsteady SIMPLE IICE * - Discretization of the turbulence equation - Solver for the linear equation system for $\nu_t(k, \epsilon, ...)$ *
Postprocessing of output data		- Generation of output data * - Human errors in the postprocessing of simulation results

* contains round-off

Figure 4.5: Error map of the presented CFD software OpenFOAM®

simulation itself. In the present case, the actual simulation provides a formal order of accuracy between $\hat{p} = 1$ and $\hat{p} = 2$. Therefore, the generation and application of the MMS source terms, the BC, and the IC are at least second order accurate. Additionally, the output generation and the error evaluation in Matlab needs to have at least the same accuracy. If the accuracy of the MMS input or output processes are not sufficiently accurate, the total observed order of the simulation will lower to the accuracy of the weakest link of the chain.

4.12.8 Benchmarks

The following benchmark series for CFD is developed as a stairway in complexity. This means, the proposed benchmarks sequence starts with the assessment of restricted functionality of the code and becomes increasingly complex through the benchmark sequence (cf. figure 4.6).

All functions and variables of the following examples are listed in a table such that the reader is able to construct each source term $\hat{\mathbf{s}}_f$, the boundary and the initial conditions, BC respectively IC. The selection of the BC type (Dirichlet or Neumann) on each edge is left to the reader. Therefore different boundary scenarios can be assessed on the same example. The only restriction is that at least one Dirichlet BC has to be set for each ϕ using incompressible flows. All of the input parameters of the benchmarks are defined in the International System of Units (SI) and their derived expressions [81]. Therefore, units for the input numbers of the benchmarks are generally omitted.

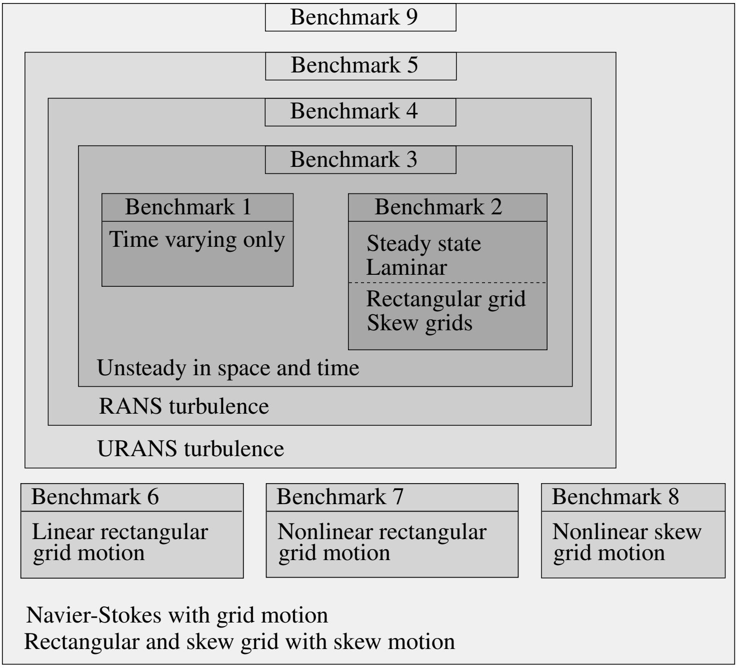


Figure 4.6: Hierarchical benchmark sequence for CFD

4.12.9 Source Term Application

For the application of the MMS on the turbulent incompressible Navier-Stokes equations with grid motion, all governing equations in general need a manufactured source term $\hat{\mathbf{s}}_f$. Sometimes, the code to be assessed gives no access to some equations for the addition of a source term. Therefore, the manufactured solutions $\hat{\phi}$ have to be designed such that $\hat{\phi}$ directly fulfills the non-accessible balance equations, which is equal to a zero source term. In the present case of OpenFOAM®, all equations can be accessed. As already mentioned in section 4.12.1, only two turbulent fields are chosen (e.g., $\hat{k}$ and $\hat{\nu}_t$) and the third is derived using equation 4.75. If all three fields were arbitrary chosen, a source term for equation 4.75 would be necessary.

If the mass conservation respectively the Poisson equation is not accessible or should not be accessed, the velocity field $\hat{\mathbf{u}}$ must satisfy the condition $\nabla \cdot \hat{\mathbf{u}} = 0$. As this construction in general is not a trivial task, a procedure for the construction of complex divergence-free velocity fields is proposed [26]. For the sake of flexibility of the manufactured solution fields, the velocity fields of the proposed benchmarks in general are not divergence free.

4.12.10 Laminar Navier-Stokes Equations

The laminar Navier-Stokes equations are given in equations 4.13 and 4.14. The individual chosen boundary conditions are shown in figure 4.7. Herein, D means a Dirichlet boundary condition and N means a Neumann boundary condition along equations 4.81 and 4.82. N^0 is the special case of a zero Neumann boundary condition, also known as zero-gradient BC.

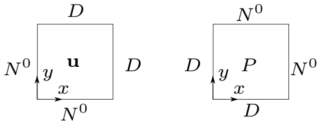


Figure 4.7: Laminar benchmarks: Boundary conditions for $\mathbf{u}$ and P

4.12.10.1 Benchmark 1: Time Assessment

The first benchmark is an exclusive exercise for the time derivative term of the unsteady Navier-Stokes equations. All other terms cancel out by the definition of the fields, as their derivatives in space are zero. Table 4.1 gives the overview of this benchmark.

Table 4.1: Benchmark 1: Overview table

fields	material	domain
$\hat{u}_x = \sin(50 \cdot \pi t)$	$\nu_m = 0.5$	$X_1 = x \in [0, 1]$
$\hat{u}_y = -\sin(50 \cdot \pi t)$		$X_2 = y \in [0, 1]$
$\hat{\mathbf{u}}_g = \mathbf{0}$		$t \in [0, 0.01]$
$\hat{P} = \sin(50 \cdot \pi t)$		

The manufactured field definition of the velocities is divergence-free. This is shortly shown in equation 4.84

$$\nabla \cdot \hat{\mathbf{u}} = \frac{\partial \hat{u}_x}{\partial x} + \frac{\partial \hat{u}_y}{\partial y} = 0 \tag{4.84}$$

The amplitude of the velocities and the kinematic pressure is one, and the first quarter, means the first 90° of each sinus wave is simulated within the simulation time (cf. figure 4.8).

The results produced by the presented framework are illustrated in figures 4.14-4.16. The left parts of the named figures show the log-log plot of the error norms of the fields over refinement for the current benchmark example. The right parts of the named figures show the observed order of accuracy p with refinement. As this benchmark is only varying over time, a coarse grid is chosen for all time refinement steps.

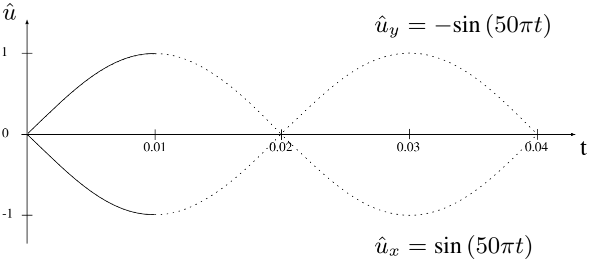


Figure 4.8: Benchmark 1: Flow velocity $\hat{\mathbf{u}}$ over time

4.12.10.2 Benchmark 2: Spatial Assessment

For the individual assessment of the spatial terms respectively steady state simulations, the manufactured solutions are chosen without time variation. This means, the term $\frac{\partial \hat{\mathbf{u}}}{\partial t}$ is vanishing from the momentum equation. The choice of the variables and the manufactured field solutions are presented in table 4.2. The velocity fields and the pressure are shown exemplarily in figure 4.9. The chosen $\nu_m = 0.5$ is much bigger than the viscosity of the desired applications in air or water. But a high viscosity guarantees the balance of the diffusion compared to the convection. As elaborated in chapter 2, this balance assures that the discretization errors, produced by the different terms of the governing equations, have the same magnitude and can therefore be captured by the order of accuracy test.

Table 4.2: Benchmark 2: Overview table

fields	material	domain
$\hat{u}_x = \sin(x^2 + y^2) + 0.001$	$\nu_m = 0.5$	$X_1 = x \in [0, 1]$
$\hat{u}_y = \cos(x^2 + y^2) + 0.001$		$X_2 = y \in [0, 1]$
$\hat{\mathbf{u}}_g = \mathbf{0}$		$t \in [0, 0.01]$
$\hat{P} = \sin(x^2 + y^2) + 2.0$		

In this benchmark, different shapes of the grid are assessed. For this assessment, three representative grids are under investigation (cf. figures 4.10):

- Rectangular grid, cf. figure 4.10a
- Grid skewness in the interior domain only, cf. figure 4.10b
- Skew grid, cf. figure 4.10c

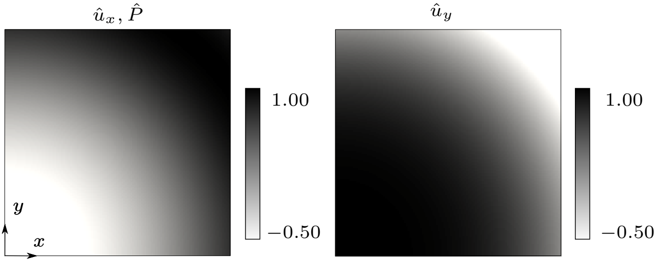
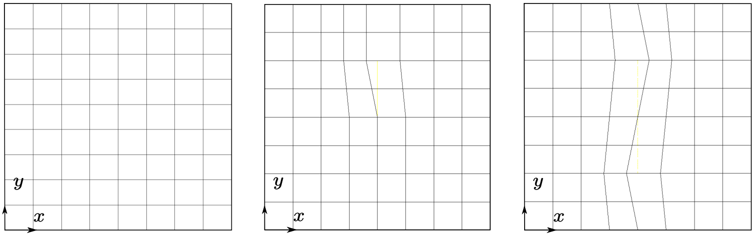


Figure 4.9: Benchmark 2: Field plot



(a) rectangular grid (b) interior grid skew (c) skew grid

Figure 4.10: Coarsest grid with different skewness

The results of the MMS assessment of the presented framework for the three types of grids are illustrated in figures 4.17-4.19.
The results show that the observed order of accuracy lowers towards $p = 1$ as soon as the grid becomes skew. It seems furthermore that the accuracy of the complete field is most sensitive to skewness at the boundaries of the domain. It is not reaching clearly $p = 1$, as the lower accurate implemented terms, which are sensitive to grid skewness, are small compared to the main terms in the resolved grid area [49].

4.12.10.3 Benchmark 3: Spatial and time Assessment

In this benchmark, all of the manufactured solution fields are defined variable in space and time to assess the time and the spatial discretization together. As discussed in sections 2.3.3.6 and 4.12.5, the resolutions of space and time discretization have to be adapted along the formal orders of accuracy to rigorously assess both discretization. For the assessment of the time and spatial accuracy together, the refinement in space is chosen as $r_s = 2$ and over time as $r_t = 4$ for the BDF schemes. For the CN time discretization, the refinement factors are set to $r_s = r_t = 2$, as its accuracy of $\hat{p}_t = 2$ was confirmed in Benchmark 1.

The choice of the variables and the manufactured field solutions are outlined in table 4.3. The velocity fields and the pressure are shown exemplarily in figure 4.11.

The results of the benchmark of the rectangular grid (cf. figure 4.10a) are shown in the figures 4.20-4.22.

Table 4.3: Benchmark 3: Overview table

fields	material	domain
$\hat{u}_x = x \sin\left(\frac{\pi t}{2}\right) \cos(\pi x^2 y^2) \sin(\pi x^2 y^2)$	$\nu_m = 0.5$	$X_1 = x \in [0,1]$
$\hat{u}_y = -y \sin\left(\frac{\pi t}{2}\right) \cos(\pi x^2 y^2) \sin(\pi x^2 y^2)$		$X_2 = y \in [0,1]$
$\hat{\mathbf{u}}_g = \mathbf{0}$		$t \in [0,1]$
$\hat{P} = \sin\left(\frac{\pi t}{2}\right) \cos\left(\pi(1-x)(1-y)\right)$		

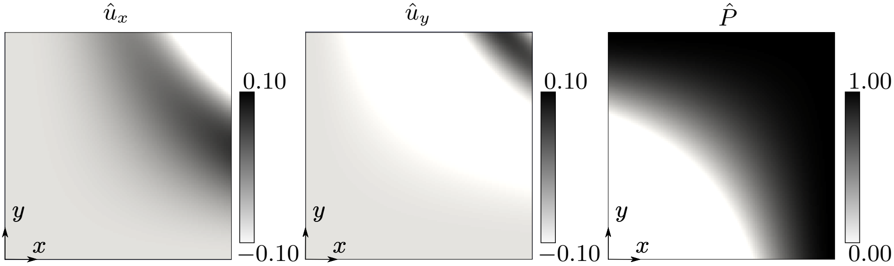


Figure 4.11: Benchmark 3: Fields $\hat{\mathbf{u}}$ and P at $t = 1.0$

4.12.10.4 Conclusion

The results of the benchmarks 2-3 show within the presented framework an observed order of accuracy in space for a rectangular grid of $p_s^{rectangular} \approx 2$, and for a skew grid of $p_s^{skew} \approx 1$. This corresponds to the formal order of spatial accuracy for the two kinds of grids, elaborated in equations 4.68 and 4.69. The benchmarks 1 and 3 confirm with $p \approx 1$ the formal order of accuracy for both BDF schemes of $\hat{p}_t^{BDF} = 1$. The Crank Nicolson scheme confirms with $p \approx 2$ its formal order of accuracy of $\hat{p}_t^{CN} = 2$.

4.12.11 Turbulent Navier-Stokes Equations

The turbulent URANS equations are represented by the equations 4.15 and 4.16. The presented $k - \epsilon$ model, outlined in equations 4.17-4.20, is assessed in the following benchmarks of this section. As for the turbulent parameters discretized equations are solved, their convergence can also be assessed. Consequently besides the convergence of the velocities $\mathbf{u}$ and the pressure P, the convergence of k and ϵ are assessed. The boundary conditions of the fields are exemplarily chosen as shown in figure 4.12.

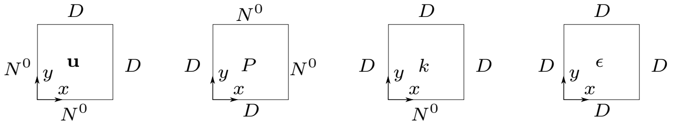


Figure 4.12: Turbulent benchmarks: Boundary conditions

Remembering the flexibility of the presented MMS method, this benchmark could also be used to assess, e.g., Prandtl's mixing length model or a $k-\omega$ turbulence model with the presented fields of the following two sections. The only change is in the individual source term of the subset of turbulence equations. As ν_t is created as a manufactured field, nothing changes therefore for the solution of the interior equations 4.17-4.20, which again underlines the generality and flexibility of the MMS.

4.12.11.1 Benchmark 4: Steady State

The steady state benchmark example for the RANS equations with turbulence is based on the laminar steady state benchmark number 2 with additional turbulence. The field and variable definitions are presented in table 4.4. The results for all fields are shown in figure 4.23.

Table 4.4: Benchmark 4: Overview table

fields	material	domain
$\hat{u}_x = \sin(x^2+y^2)+0.001$	$\nu_m = 0.5$	$X_1 = x \in [0,1]$
$\hat{u}_y = \cos(x^2+y^2)+0.001$		$X_2 = y \in [0,1]$
$\hat{\mathbf{u}}_g = \mathbf{0}$		$t \in [0, 0.01]$
$\hat{P} = \sin(x^2+y^2)+2.0$		
$\hat{k} = 0.1 \cdot \sin(x+y)^2 + 0.1$		
$\hat{\nu}_t = 0.1 \cdot \cos(x+y)^2 + 1.0$		
$\hat{\epsilon} = 0.09 \cdot \frac{k^2}{\nu_t}$		

4.12.11.2 Benchmark 5: Unsteady

The unsteady benchmark example for the URANS equations with turbulence is based on the laminar unsteady benchmark number 3 with additional turbulence. The field and variable definitions are presented in table 4.5. It should be noted that ν_t, k and ϵ must not be negative. Therefore, the sin-functions are lifted with a constant offset to set the minimum value of k and ϵ to zero. The produced results with the presented framework for the different time discretization and differently chosen refinement factor combinations of r_s and r_t are shown in figures 4.24, 4.25, and 4.26.

Table 4.5: Benchmark 5: Overview table

fields	material	domain
$\hat{u}_x = x\sin\left(\frac{\pi t}{2}\right)\cos(\pi x^2 y^2)\sin(\pi x^2 y^2)$	$\nu_m = 0.5$	$X_1 = x \in [0,1]$
$\hat{u}_y = -y\sin\left(\frac{\pi t}{2}\right)\cos(\pi x^2 y^2)\sin(\pi x^2 y^2)$		$X_2 = y \in [0,1]$
$\hat{\mathbf{u}}_g = \mathbf{0}$		$t \in [0, 1.0]$
$\hat{P} = \sin\left(\frac{\pi t}{2}\right)\cos\left(\pi(1-x)(1-y)\right)$		
$\hat{k} = 0.1 \cdot \sin\left(\frac{\pi t}{2}\right) \cdot \sin(x+y)^2 + 0.1$		
$\hat{\nu}_t = 0.1 \cdot \sin\left(\frac{\pi t}{2}\right) \cdot \cos(x+y)^2 + 1.0$		
$\hat{\epsilon} = 0.09 \cdot \frac{k^2}{\nu_t}$		

4.12.11.3 Conclusion

Compared to the benchmarks 1-3 of this chapter, the additional RANS/URANS turbulence does not influence the accuracy of the primary variables $\mathbf{u}$ and P. The URANS turbulence fields are also converging with at least an observed order of accuracy of $p_t = 1$. In space, k and ϵ show an order of accuracy of $p_s = 2$ for rectangular grids, which corresponds to the behavior of the velocity field $\mathbf{u}$. As a note for the convergence of the turbulent fields, it should be mentioned that the turbulence equations are only solved once at the end of each timestep. Therefore the unsteady results of k, ϵ, and ν_t are not fully converged within the current timestep. Furthermore, the fields $\mathbf{u}$ and P are therefore solved using the turbulent ν_t from the old timestep. This arrangement lowers the order of accuracy despite a possible improvement of the time integration for the turbulent parameters for the sake of efficiency of the code. As one can e.g. see in figure 4.26, in the region of very high refinement, there is an additional drop in accuracy observed. This can be interpreted as an additional error source, that drops the accuracy of the turbulent parameters to a value smaller than one. Despite this fact, the primary variables $\mathbf{u}$ and P are solved consistently in second order space and at least first order in time.

4.12.12 Grid Motion

As mentioned above, the grid motion influences the solution of the Navier-Stokes equations. The grid velocity $\mathbf{u}_g$ in the convection term is a part of the balance equation 4.30. As presented above, the grid motion is determined by a secondary equation for the grid displacement $\mathbf{d}_g$. Afterwards the grid velocity $\mathbf{u}_g$ is calculated using the grid displacement $\mathbf{d}_g$ along equations 4.51-4.53.

The grid displacement $\mathbf{d}_g$ is determined solving equation 4.49 respectively 4.50. As equation 4.50 also constitutes a discretized PDE, an order of accuracy test using the

MMS is performed for $\mathbf{d}_g$.

It has to be mentioned that the probe positions are no longer unique in moving grids during refinement.
The following two options for the probes are investigated:

- Keep the probe locations fixed in space
- Move the probe locations with the moving grid

The first option guarantees that the results are really comparable of all simulations, as the probes are at the same positions fixed in space.
The second option guarantees to have valid results at all probe points, as all of them are always inside the domain, independently of the size of the grid deformation. Comparisons are made during the development of this work. Both options show the same order of accuracy in all examined tests. The following results are produced using the second option for the handling of the probes, as the number of valid probes is constant.

Three test benchmarks are shown in the following sections:

- \- Linear grid motion
- \- Nonlinear rectangular grid motion
- \- Nonlinear skew grid motion

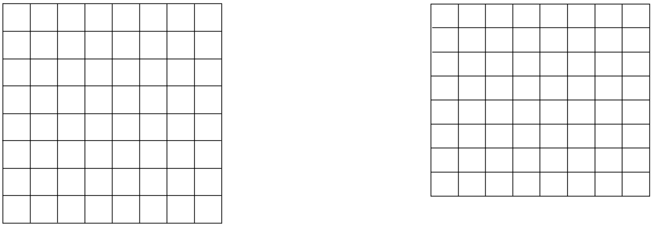
These three are chosen to separate the functionalities and to identify the properties of the discretization. The initial grid and the deformed grids at the final time step are shown in figure 4.13.

4.12.12.1 Benchmark 6: Linear Grid Motion

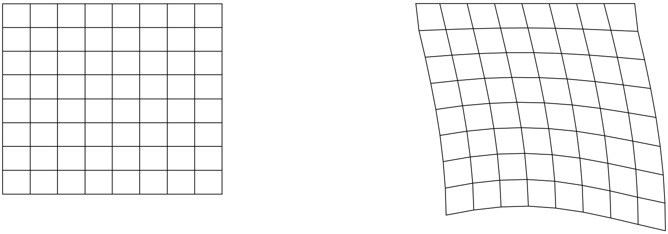
The linear grid motion benchmark is designed to be solved until machine accuracy respectively the final IICE by the used linear shape functions.
The manufactured solutions and the diffusion of the grid motion are outlined in table 4.6. The resulting error is shown in figure 4.27.

Table 4.6: Benchmark 6: Overview table

fields	material	domain
$\hat{d}_{gx} = 0$	$\Gamma_g = 0.5$	$X_1 = x \in [0, 1]$
$\hat{d}_{gy} = 0.5 \cdot (1 - y) \cdot \sin(\pi t)$		$X_2 = y \in [0, 1]$
$\hat{\mathbf{u}} = \hat{P} = 0$		$t \in [0, 0.1]$



(a) Initial grid of benchmarks $6-8$ (b) Final grid of benchmark 6



(c) Final grid of benchmark 7 (d) Final grid of benchmark 8

Figure 4.13: Initial grid (a) and grid motion at the final time step of benchmarks 6-8 in subfigures b-d

4.12.12.2 Benchmark 7: Nonlinear Rectangular Grid Motion

As a second step, a nonlinear grid motion is applied on a rectangular grid, which stays rectangular during the deformation. The manufactured solutions and the diffusion of the grid motion are presented in table 4.7. The resulting error and the observed order of accuracy is shown in figure 4.28.

Table 4.7: Benchmark 7: Overview table

fields	material	domain
$\hat{d}_{gx} = 0$	$\Gamma_g = 1.0$	$X_1 = x \in [0, 1]$
$\hat{d}_{gy} = 0.5 \cdot (1-y)^2 \cdot \sin(\pi t)$		$X_2 = y \in [0, 1]$
$\hat{\mathbf{u}} = \hat{P} = 0$		$t \in [0, 0.1]$

4.12.12.3 Benchmark 8: Nonlinear Skew Grid Motion

As the third step, a nonlinear grid motion is applied on a rectangular grid, which becomes skewed during deformation. The manufactured solutions and the diffusion of the grid motion are outlined in table 4.8. The resulting error and the observed order of accuracy is shown in figure 4.29.

Table 4.8: Benchmark 8: Overview table

fields	material	domain
$\hat{d}_{gx} = \cos(0.5\pi y)\sin(\pi t)$	$\Gamma_g = 1.0$	$X_1 = x \in [0, 1]$
$\hat{d}_{gy} = 0.5\sin(\pi x)\cos(\frac{\pi y}{2})\sin(\pi t)$		$X_2 = y \in [0, 1]$
$\hat{\mathbf{u}} = \hat{P} = 0$		$t \in [0, 0.05]$

4.12.12.4 Conclusion

The convergence of the solution of the grid motion confirms the formal order of accuracy for the spatial solution. The linear motion can be solved grid-independent to machine accuracy, the nonlinear rectangular motion is observed to converge with $p_s = 2$ and the nonlinear skew motion is observed to converge with $p_s = 1$. This coincides with the experience from the Navier-Stokes solution and the formal order analysis of the spatial terms in section 4.11.4.

4.12.13 Navier-Stokes Equations with Grid Motion

Benchmarks for the Navier-Stokes equations, containing grid motion, are presented in this section. Despite the fact of grid motion, the basic equation is still the same as the equations without grid motion (cf. equations 4.13 and 4.14). At a first glance, this may seem confusing, but only the discretized equations with and without grid motion are different (cf. section 4.9).

4.12.13.1 Benchmark 9

The manufactured solutions and the variables of the grid motion are outlined in table 4.9. The resulting error and the observed order of accuracy is shown in figure 4.29.

In this benchmark, the fluid velocity relative to the grid is kept constant. This actually does not fulfill the manufactured solution requirements for the MMS. However, as this is already assessed from the previous benchmarks, this is a sufficient test for the combination of Navier-Stokes and the grid motion equations. The results of the benchmark using a rectangular initial grid and $r_s = 2$ are shown in the E_2 norm for the CN ($r_t = 2$) and the BDF2 time integration using $r_t = 2$ and $r_t = 4$ in figures 4.30-4.32.

Similar to the previous benchmarks, the first order of accuracy in space for skew grids is confirmed. Therefore, the time refinement factors r_t are generally set equal to the refinement factor in space $r_s = 2$ as an observed order of accuracy of $p = 1$ is expected. The results for the skew initial grids are shown in figures 4.33 and 4.34.

Table 4.9: Benchmark 9: Overview table

fields	material	domain
$\hat{u}_x = 1 + \hat{u}_{gx}$	$\nu_m = 0.5$	$X_1 = x \in [0, 1]$
$\hat{u}_y = 2 + \hat{u}_{gy}$	$\Gamma_g = 1.0$	$X_2 = y \in [0, 1]$
$\hat{u}_{gx} = \pi \cdot \cos\left(\frac{\pi}{2} \cdot y\right) \cdot \cos(\pi \cdot t)$		$t \in [0, 1]$
$\hat{u}_{gy} = 0.5\pi \cdot \sin(\pi \cdot x) \cdot \cos\left(\frac{\pi}{2} \cdot y\right) \cdot \cos(\pi \cdot t)$		
$\hat{P} = 2 + 100\cos(x + y) \cdot \sin(\pi \cdot t)$		
$\hat{\mathbf{d}}_g = \int_t \hat{\mathbf{u}}_g dt$		

4.12.13.2 Conclusion

For the solution of the Navier-Stokes equations with grid motion it can be concluded that the spatial solution converges along the steady state solutions without grid motion. In terms of observed order of accuracy, this means $p_s^{skew} = 1$ and $p_s^{rectangular} = 2$. The code has "problems" solving the unsteady Navier-Stokes equations in combination with the grid motion using the trapezoidal method. This can be seen in figures 4.30 and 4.33. Here, the pressure does not converge at all over refinement. Using the BDF time discretization, the solutions convergence for all fields with $p \geq 1$, which corresponds to the formal orders of accuracy.

4.12.14 Statement and Results for the used CFD Environment

Summarizing this chapter and the benchmarks particularly, the code OpenFOAM® is assessed to work mostly as intended. The only problem arised during a benchmark combining the Navier-Stokes solution with grid motions using the trapezoidal rule for the time discretization. This defines a task or a challenge for the code developer to improve the code. At this point it is mentioned again that the MMS is not only competent for the confirmation of code functionalities. It is very helpful to precisely identify and localize mistakes in the code.

All other features of the code, containing rectangular and skew grid handling, laminar and turbulent simulations, work as intended and confirm the formal orders of accuracies derived above. The elaborated benchmark sequence with its increasing stages of complexity build confidence in the functionalities and features of the code. It can be concluded that the governing equations in the code are solved consistently, with the only restriction of the problem mentioned above. Therefore, the positively assessed parts of the code can be used for further investigations in the context of V&V to finally reach a predictive capability of the software (cf. [9, 57, 75, 79] and chapter 2).

4.13 Adaptation of the CFD Framework

As shown above in this chapter, the MMS framework is applied to a particular CFD environment for modeling and simulation of laminar and turbulent incompressible fluid dynamics. However, the very general concept of the framework makes it very attractive for the adaptation to other CFD environments or adaptations and extensions of the assessed CFD software.

In the next sections, a few examples of potential adaptations are outlined. The given examples focus on the required adaptations within the presented MMS environment. The actual modeling and simulation functionalities regarding the solution of the considered field or the internal data handling are not part of the description, as they should be anyway available independent of an MMS assessment.

Furthermore it is obvious, if the environment is changed, the adapted parts need a separate investigation to derive the formal order of accuracy. This is required before the execution of the individual benchmarks, to set the refinement factors correct (cf. section 4.12.5).

It will be shown that only moderate changes are necessary to adapt the MMS framework for the assessment of slightly adapted, or even different CFD software environments. It may be concluded that the MMS framework lives up to its promised flexibility, adaptivity, and modularity.

4.13.1 Adaptation of the Turbulence Model

The demand for the assessment of further turbulence models in CFD is obvious. Therefore, the framework of chapter 4 is slightly adapted. Changing the turbulence model to another model, which is based in the Boussinesq approximation (this means, a ν_t exists for the influence of the turbulence on the actual Navier-Stokes equations), just needs the source term derivation based on the chosen equations of the turbulence models parallel to the equations 4.75-4.79. As the turbulent viscosity ν_t is generally defined in the presented CFD benchmarks of this chapter, the benchmarks can directly be used for the assessment of the adapted environment. Additional suggested MMS benchmarks and the assessment of different RANS and URANS turbulence models are given in [23–25].

The adaptation of the turbulence model to an LES sub-grid scale model behaves in a similar manner. The only challenge to be tackled is that the equations solving the turbulence viscosity are dependent on a filter length. In CFD, the filter length of the unresolved scales is either set equal to the current cell size or the user can specify it by a preset. As an example, the Smagorinsky-Lilly model is mentioned [102]. To assess the implementation nevertheless with the presented MMS framework, two different possibilities are mentioned shortly:

If the value of the filter length can be preset, it has to stay constant over the refinement study an a stringent assessment can be performed.

If the grid represents the filter width directly, the individual cell size enters the analytical equations as a constant. This finally leads to a manufactured solution, which is grid dependent. From the perspective of this work, it cannot be estimated, if this would really assess the consistency of the implementations in a stringent way, as a changing analytical equation actually means changing the physical problem.

The theoretical convergence having infinity grid points would therefore result in a vanishing filter, solving the actual Navier-Stokes analogously to a DNS.

4.13.2 Adaptation of the Discretization Method

Assuming the change from a Finite Volume in CFD to a Finite Element formulation, no changes are necessary in the MMS environment. As already mentioned, a very important topic is the accessibility to the equations and the ability to add the derived source terms $\hat{\mathbf{s}}_f$ (cf. section 4.12.1) in the new software component.

Beside changing the theory of the discretization, one could also think of the change of the volume discretization. For example tetrahedral or polyhedral elements could be used to discretize the fluid domain instead of the presented rectangular elements.

The mentioned changes in this section only require the generation of new grids; this means, the manufactured solutions and the benchmarks can directly be applied for all types of volume elements.

4.14 Results

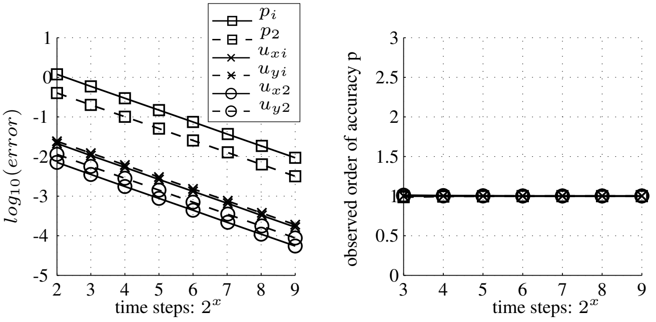


Figure 4.14: Benchmark 1: Time convergence plot using BDF1

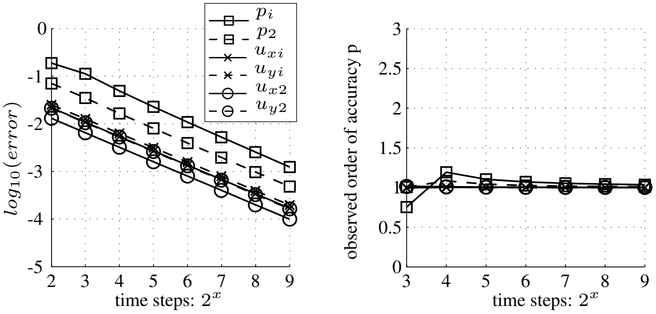


Figure 4.15: Benchmark 1: Time convergence plot using BDF2

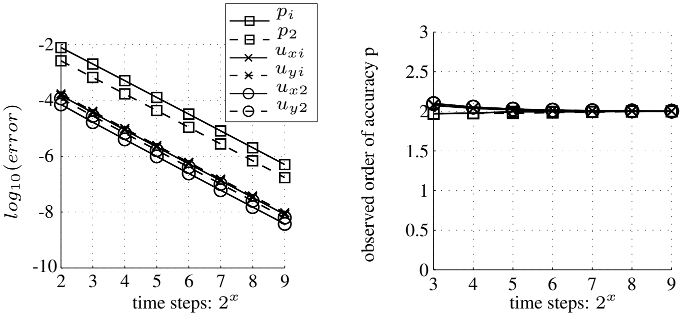


Figure 4.16: Benchmark 1: Time convergence plot using CN

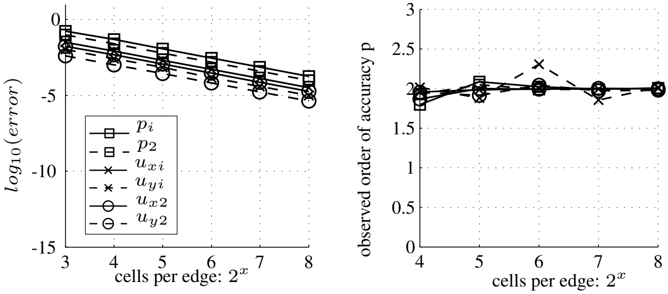


Figure 4.17: Benchmark 2: Spatial convergence plot on the rectangular grid

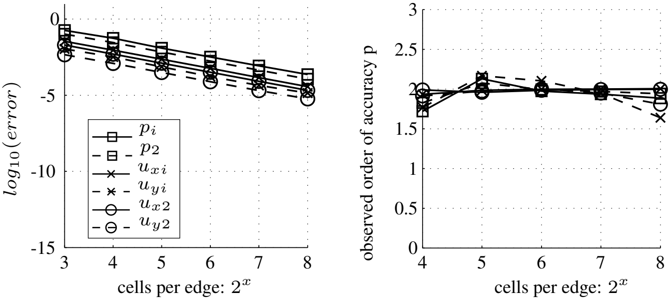


Figure 4.18: Benchmark 2: Spatial convergence plot on the interior skew grid

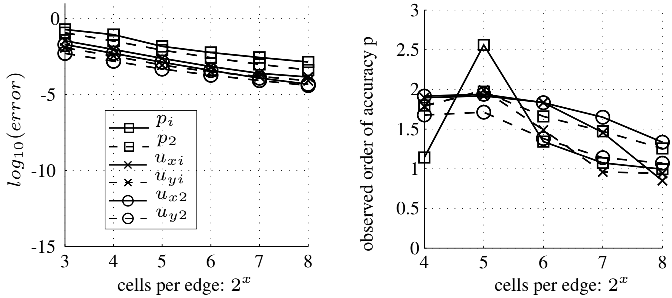


Figure 4.19: Benchmark 2: Spatial convergence plot on the skew grid

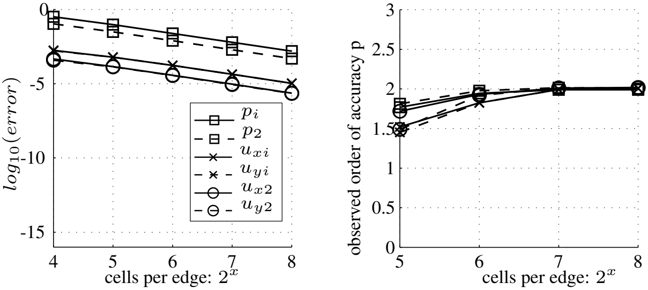


Figure 4.20: Benchmark 3: Convergence plot on the rectangular grid using BDF1 with $r_t = 4$

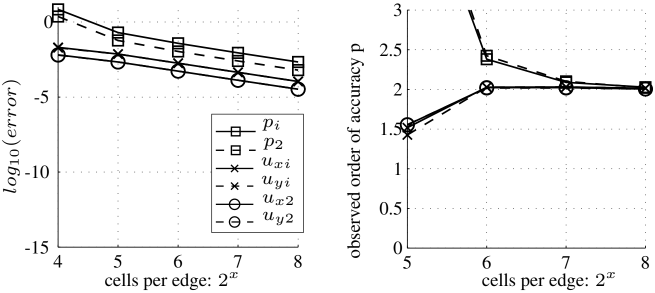


Figure 4.21: Benchmark 3: Convergence plot on the rectangular grid using BDF2 with $r_t = 4$

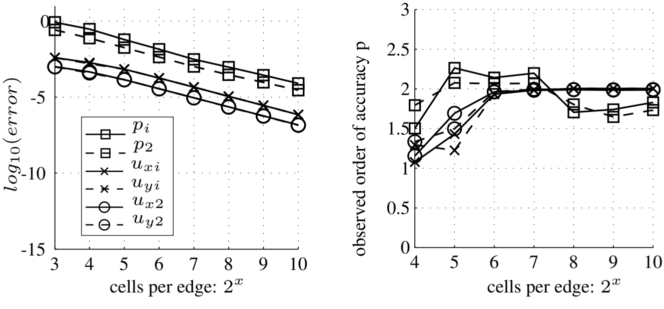


Figure 4.22: Benchmark 3: Convergence plot on the rectangular grid using CN with $r_t = 2$

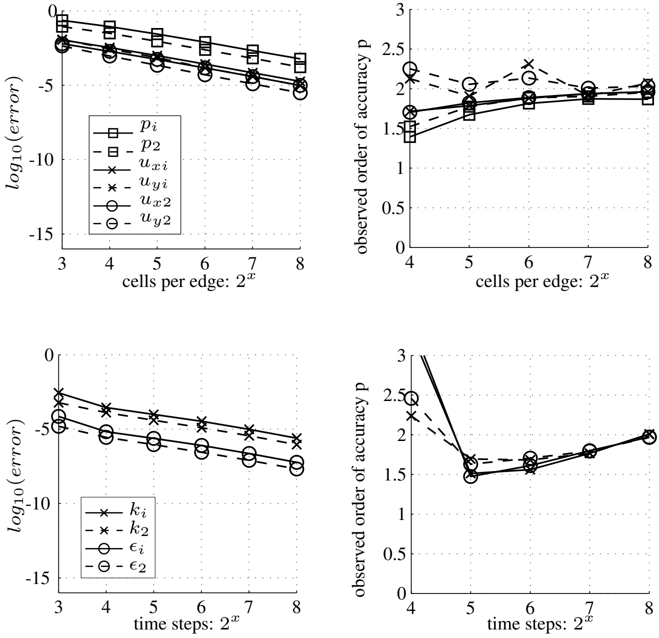


Figure 4.23: Benchmark 4: Spatial convergence plot on the rectangular grid

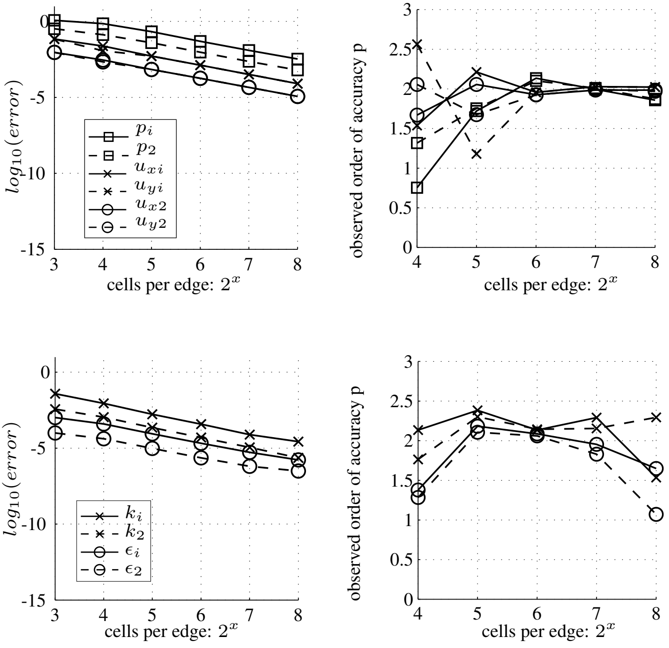


Figure 4.24: Benchmark 5: Convergence plot on the rectangular grid using CN with $r_t = 2$

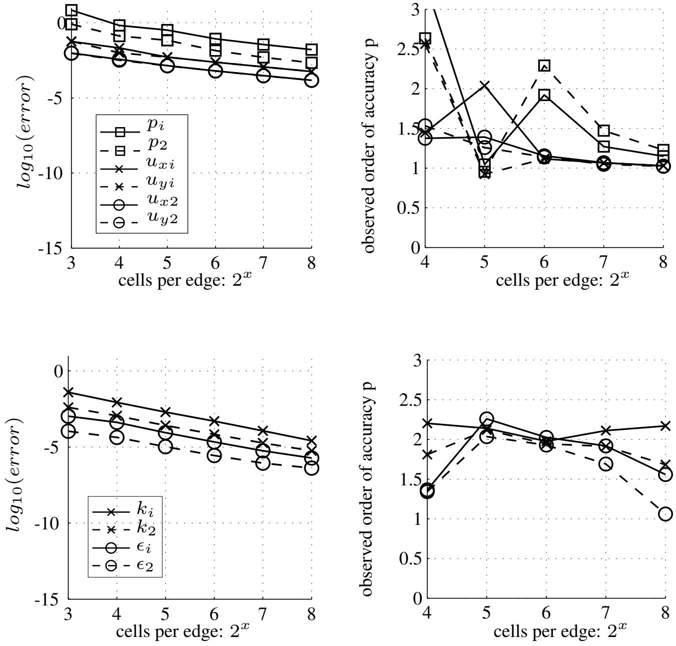


Figure 4.25: Benchmark 5: Convergence plot on the rectangular grid using BDF2 with $r_t = 2$

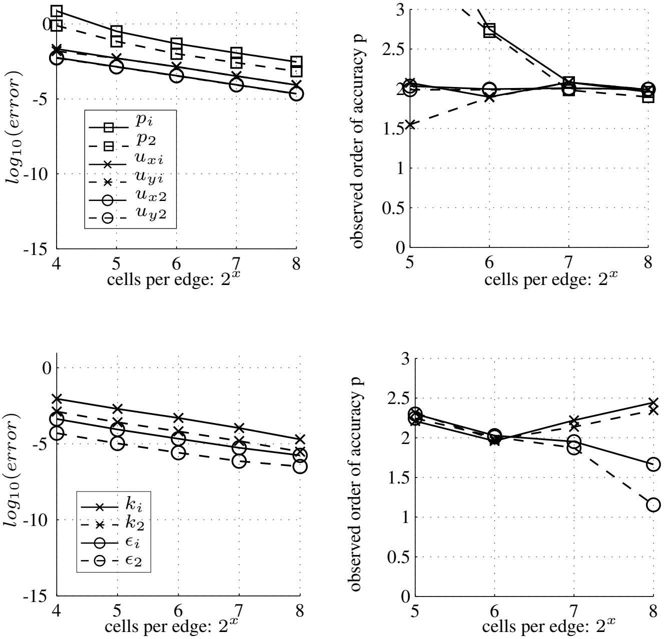


Figure 4.26: Benchmark 5: Convergence plot on the rectangular grid using BDF2 with $r_t = 4$

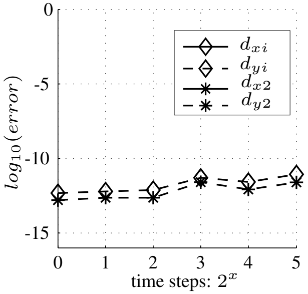


Figure 4.27: Benchmark 6: Convergence plot of a linear grid motion of a rectangular grid

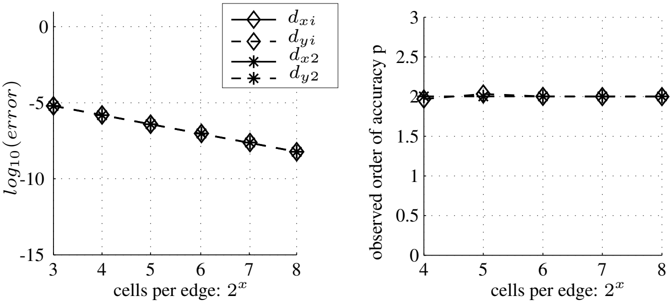


Figure 4.28: Benchmark 7: Convergence plot of a nonlinear grid motion of a rectangular grid

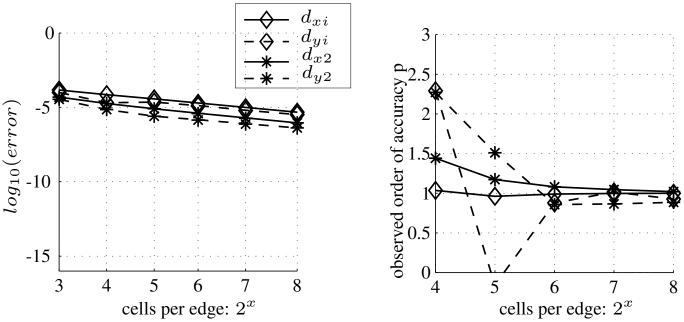


Figure 4.29: Benchmark 8: Convergence plot of a nonlinear skew grid motion of a rectangular grid

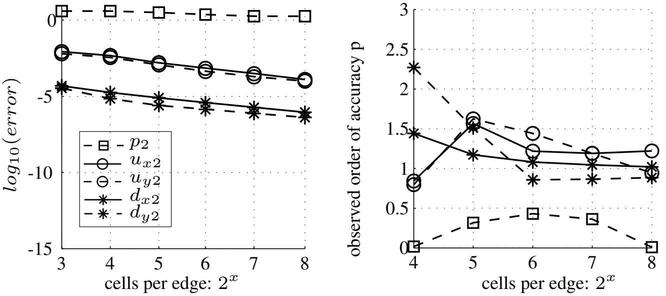


Figure 4.30: Benchmark 9: Convergence plot of a nonlinear skew grid motion of a rectangular grid using CN with $r_t = 2$

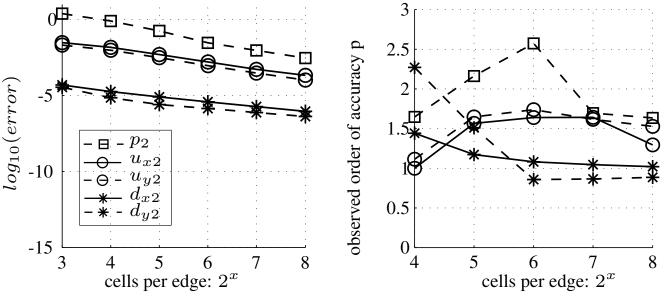


Figure 4.31: Benchmark 9: Convergence plot of a nonlinear skew grid motion of a rectangular grid using BDF2 with $r_t = 2$

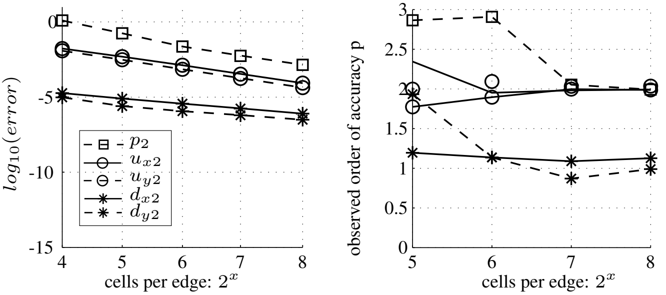


Figure 4.32: Benchmark 9: Convergence plot of a nonlinear skew grid motion of a rectangular grid using BDF2 with $r_t = 4$

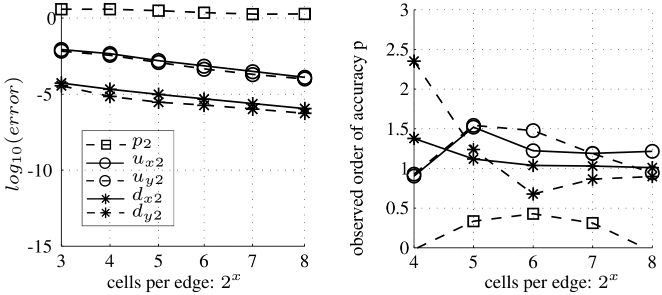


Figure 4.33: Benchmark 9: Convergence plot of a nonlinear skew grid motion of a skew grid using CN with $r_t = 2$

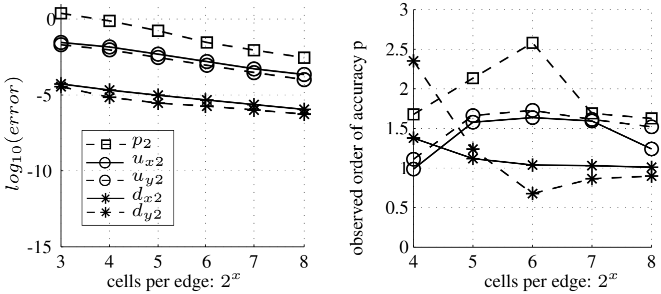


Figure 4.34: Benchmark 9: Convergence plot of a nonlinear skew grid motion of a skew grid using BDF2 with $r_t = 2$

CHAPTER 5

Coupling and FSI

Fluid-Structure Interaction (FSI) is concerned with the interaction of structure with its surrounding fluid flows. Large interactions require joint simulations of the two fields.
The particular application of interest of lightweight membrane structures in the wind leads to large dynamic deflections and interaction with the wind, which therefore requires a coupled FSI analysis.

This chapter aims at providing an overview of the requirements, the assumptions and the implementations of the coupling procedures and of the complete FSI process of the available environment using the toolbox EMPIRE. In the presented partitioned FSI environment, the CSD and CFD software fields, which already have been presented separately in chapters 3 and 4 respectively, are coupled together. The topic of non-matching grids at the common interface between the two fields is a main focus of this chapter.

For the application of the MMS in the context of field coupling and FSI, a framework for the assessment of the environment is presented. Furthermore, the application of the MMS, giving a hierarchical benchmark sequence and produced results are elaborated. Using the results and the derived formal order of accuracy, the complete FSI environment is assessed for the intended application.

The derivation of the balance equation at the common interface of the fluid and the structure is kept relatively short. A detailed derivation, containing discussions of different possibilities for the interface modeling can be found in [18, 27, 36, 40, 54, 62–64, 70, 101, 125].

5.1 FSI Overview

FSI environments usually are distinguished in monolithic and partitioned approaches. As this distinction is not unambiguously used in the literature, this work leans to the definitions in [70]: Monolithic means that the (two) fields and their coupling are treated in a system of aggregated equations. This means, all fields and their interactions can be solved synchronously. In contrast in partitioned approaches, the equations from the fields are segregated and their solution therefore is treated asynchronously. This asynchrony in the solution always induces a lag between the two fields.

The named lag can be minimized, using iterative, or strongly-coupled partitioned procedures. Iterative means that in a single time step the problem is sub-iterated until convergence, which almost represents the aggregated system of equations from the monolithic approach [70, 99]. If the named lag can be accepted, no sub-iterations are performed which is the case in loose, or loosely-coupled partitioned procedures. In this case, both field equations are only solved once in a time-step.

5.2 Target Application

In the context of this thesis the target application of the Fluid-Structure Interaction (FSI) environment is the modeling and simulation of lightweight and flexible elastic membrane structures in atmospheric wind. The flexibility, the relatively small self weight, and the load carrying behavior of lightweight structures require a simulation environment for large deformations of the structures. As soon as the structural deformations in the wind regime no longer are sufficiently small to be neglected, the interaction of the fluid and the structure should not be performed independently. In order to model the behavior in a more realistic manner, the interaction of the fluid and the structure must be taken into account in an FSI analysis.

5.3 Restrictions for the Application

As it is not possible to completely test and assess all parts of codes and code combinations [73], the target application has to be defined. This definition, or rather restriction, defines the functionality to be tested in a benchmark sequence in the following sections.

The restrictions concerning the FSI functionality due to the available environment are outlined in the following list:

- partitioned FSI with surface coupling
- Dirichlet-Neumann partitioning using force elimination
- Gauss-Seidel communication

An overview of the partitioned FSI environment EMPIRE and its assessed features of the coupling software Emperor, is illustrated in figure 5.1 respectively figure 5.2. The shown EMPIRE_API, added to the software Carat++ and OpenFOAM® provides the communication with the coupling software Emperor.

5.4 Requirements to the Coupling Interface

As named above, a Dirichlet-Neumann decomposition is chosen for the partitioning of the FSI process [117]. This means for the specific case of the chosen environment that on the one hand the displacements (Dirichlet), and on the other hand the

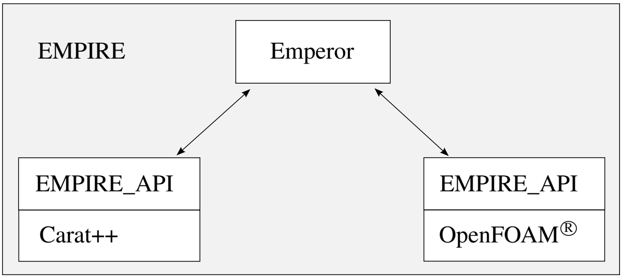


Figure 5.1: Overview of the FSI environment of EMPIRE

Coupling - EMPIRE & Emperor
Finite Element Method Dirichlet-Neumann coupling Gauss-Seidel communication Partitioned FSI Loose and iterative coupling

Figure 5.2: Overview of the assessed features of EMPIRE and Emperor

traction forces (Neumann) are used as surface coupling variables at the common interface [64]. Therefore, the requirements for the interface are the functionality of surface transfer of geometry, and the sampling and mapping of forces and displacements in a conservative resp. consistent way. In particular, the partitioned approach addresses the topic of non-matching grids at the common interface.

5.5 Equilibrium and Compatibility

5.5.1 Geometry

The FSI problem is defined in a domain Ω, which consists of the two non-overlapping sub-domains Ω_s and Ω_f.
The sub-domain Ω_s as the structural domain and Ω_f as the fluid domain share the common interface Γ, which is related to the initial configuration. Boundary conditions are defined at the boundary of the structure Γ_s, and the boundary of the fluid Γ_f. Therefore, the common interface is defined as $\Gamma = \Gamma_s \cap \Gamma_f$.

5.5.2 Single Field Equilibria

The structural and the fluid fields are each solved in their actual way as presented in the chapters 3 and 4. Therefore they can be seen as block box solvers inside the FSI process.

The equations of balance are repeated in their differential form for the sake of completeness. The equation of balance of momentum in the CSD software Carat++ defined on Ω_s is denoted in equation 5.1 (cf. equation 3.2).

$$-\rho_s \frac{\partial^2 \mathbf{d}}{\partial t^2} + \nabla \cdot \mathbf{P} + \rho_s \mathbf{Q} = 0 \tag{5.1}$$

The incompressible and laminar Navier-Stokes equations in the CFD software OpenFOAM®defined on Ω_f are presented in equations 5.2 and 5.3 (cf. equations 4.13 and 4.14).

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) - \nabla \cdot (2\nu_m \mathbf{D}) + \nabla P = 0 \tag{5.2}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{5.3}$$

5.5.3 Coupling Conditions

The two separate fields are coupled together at the common interface Γ. The Dirichlet coupling condition uses the field informations of the velocities $\mathbf{u}$, the pressure P, and the grid deformation $\mathbf{d}_g$ of the fluid, and the displacement $\mathbf{d}$ of the structure. Assuming attached surfaces at Γ, the displacement of the structure represents the boundary deformation of the fluid at Γ as well, as shown in equation 5.4.

$$\mathbf{d}_g(\mathbf{X}^\Gamma, t) - \mathbf{d}(\mathbf{X}^\Gamma, t) = 0 \tag{5.4}$$

The Dirichlet coupling condition depicted in equation 5.4 is also called the compatibility or the kinematic condition at the interface Γ. The integral form of this equation, using Galerkin weighted residuals with the test function $\tilde{\mathbf{d}}$, is presented in equation 5.5:

$$\int_\Gamma \mathbf{d}_g(\mathbf{X}^\Gamma, t)\tilde{\mathbf{d}} d\Gamma - \int_\Gamma \mathbf{d}(\mathbf{X}^\Gamma, t)\tilde{\mathbf{d}} d\Gamma = 0 \tag{5.5}$$

The Neumann coupling condition is the balance of traction forces at the common interface, also called the dynamic continuity condition. The traction force equilibrium is formulated and defined with respect to the current configuration. Therefore the balance of traction forces act on the interface of the current configuration γ. For incompressible fluids, this condition is shown in equation 5.6 with equations 5.7 and 5.8:

$$\mathbf{t}_s^\gamma + \mathbf{t}_f^\gamma = 0 \tag{5.6}$$

with

$$\mathbf{t}_f^\gamma = \boldsymbol{\sigma}_f \cdot \mathbf{n}_f^\gamma = (-\rho_f P \mathbf{I} + 2\nu_m \rho_f \mathbf{D}) \cdot \mathbf{n}_f^\gamma = 0 \tag{5.7}$$

$$\mathbf{t}_s^\gamma = \boldsymbol{\sigma}_s \cdot \mathbf{n}_s^\gamma \tag{5.8}$$

In equation 5.6, the tractions are related to the current configuration. Therefore they are based on the Cauchy stresses $\boldsymbol{\sigma}$ (cf. equations 5.7 and 5.8). As the equilibrium forces at the structure in the used formulation are related to the initial configuration, the forces actually have to be applied related to the PK1 stresses $\mathbf{P}$ (cf. equation 5.1).

As elaborated in [45], the resultant force of a traction vector over the surface is independent of the formulation. Therefore, using Nanson's formula [45] and the relations of the stress tensors (cf. section 3.5), the balance of traction force integrals is outlined in equation 5.9 [45]. Capital and lower case letters still are related to the initial configuration resp. the current configuration.

$$\mathbf{t}(\mathbf{x}^{\gamma}, t)d\gamma = \mathbf{T}(\mathbf{X}^{\Gamma}, t)d\Gamma \tag{5.9}$$

This leads to the applied integral form of the force equilibrium equation 5.6 developed in equations 5.10-5.12:

$$\int_{\gamma} \mathbf{t}_s^{\gamma} d\gamma + \int_{\gamma} \mathbf{t}_f^{\gamma} d\gamma = 0 \tag{5.10}$$

$$\int_{\Gamma} \mathbf{T}_s^{\Gamma} d\Gamma + \int_{\gamma} \mathbf{t}_f^{\gamma} d\gamma = 0 \tag{5.11}$$

$$\int_{\Gamma} \mathbf{T}_s^{\Gamma} d\Gamma + \int_{\gamma} (-\rho_f P\mathbf{I} + 2\nu_m \rho_f \mathbf{D}) \cdot \mathbf{n}_f^{\gamma} d\gamma = 0 \tag{5.12}$$

For the later given application of the Galerkin approach, the integrals of equation 5.12 are unified with respect to $d\Gamma$. Therefore, the traction of the fluid is translated/converted to the initial configuration in equations 5.13-5.16, using equations 5.17 and 5.18.

$$\int_{\Gamma} \mathbf{T}_s^{\Gamma} d\Gamma + \int_{\gamma} \mathbf{t}_f^{\gamma} d\gamma = 0 \tag{5.13}$$

$$\int_{\Gamma} \mathbf{T}_s^{\Gamma} d\Gamma + \int_{\Gamma} \mathbf{t}_f^{\gamma} \det(\mathbf{F}) d\Gamma = 0 \tag{5.14}$$

$$\int_{\Gamma} \mathbf{T}_s^{\Gamma} d\Gamma + \int_{\Gamma} \frac{\mathbf{T}_f^{\Gamma}}{\det(\mathbf{F})} \det(\mathbf{F}) d\Gamma = 0 \tag{5.15}$$

$$\int_{\Gamma} \mathbf{T}_s^{\Gamma} d\Gamma + \int_{\Gamma} \mathbf{T}_f^{\Gamma} d\Gamma = 0 \tag{5.16}$$

with

$$\frac{d\gamma}{d\Gamma} = \det(\mathbf{F}) \tag{5.17}$$

$$\mathbf{T}_f^{\Gamma} = \det(\mathbf{F}) \mathbf{t}_f^{\gamma} \tag{5.18}$$

$\mathbf{F}$ is the deformation gradient of the interface grid. It can be derived analogously to equation 3.6. The Galerkin weighted residual approach is finally applied to equation 5.16 and results in equation 5.19:

$$\int_{\Gamma} \mathbf{T}_s^{\Gamma} \tilde{\mathbf{d}} d\Gamma + \int_{\Gamma} \mathbf{T}_f^{\Gamma} \tilde{\mathbf{d}} d\Gamma = 0 \tag{5.19}$$

5.6 Discretization

In this section, the discretization of the interface conditions and of the solution process over time is presented. It has to be mentioned, that the software is completely designed for non-matching grids. A matching grid scenario is included in the treatment of non-matching grids; therefore no distinction is made in the discretization.

The discretization of the structural and the fluid fields have already been discussed in sections 3.6 and 4.9.

5.6.1 Spatial Discretization

The discretization of the integral mapping equations 5.5 and 5.19 is performed using the Mortar method [29, 87]. A discussion of different possibilities of interpolation methods and Lagrange multiplier methods for FSI can be found in [29]. Using a field discretization of all variables with linear space functions, the fields are approximated for a variable $\Phi(x)$ in equation 5.20:

$$\Phi(x) \approx \sum_i N_i(\mathbf{X}^\Gamma)\Phi_i(t) \tag{5.20}$$

$\Phi_i(t)$ are the individual nodal results at each node i. The current environment uses linear shape functions $N_i(\mathbf{X}^\Gamma)$ for the fluid and the structure, therefore the definition is given in equations 3.16 - 3.19. Equation 5.21 results from the combination of equations 5.20 and 5.5:

$$\int_\Gamma \sum_i N_i^f \mathbf{d}_{gi}\tilde{\mathbf{d}} d\Gamma - \int_\Gamma \sum_i N_i^s \mathbf{d}_i \tilde{\mathbf{d}} d\Gamma = 0 \tag{5.21}$$

The applied Mortar method uses for the test functions $\tilde{\mathbf{d}}$ the same linear shape functions as the fluid field with amplitude values of $d_j = 1.0$. Therefore, the test functions simplify to equation 5.22.

$$\tilde{\mathbf{d}} = \tilde{\mathbf{d}}(\mathbf{X}^\Gamma) = N_j^f(\mathbf{X}^\Gamma) \tag{5.22}$$

Consequently, equation 5.21 results in equation 5.23:

$$\int_\Gamma \sum_i N_j^f N_i^f \mathbf{d}_{gi} d\Gamma - \int_\Gamma \sum_i N_j^f N_i^s \mathbf{d}_i d\Gamma = 0 \tag{5.23}$$

Using the clipping procedure shown in [87, 118], an elementwise integration is performed.

To fulfill the balance of traction forces at the interface, equation 5.6 also needs to be discretized.
Equation 5.19 uses the same shape function and further discretization procedure as equation 5.21. Therefore it results in equation 5.24:

$$\int_\Gamma \sum_i N_j^f N_i^f \mathbf{T}_{si}^\Gamma d\Gamma + \int_\Gamma \sum_i N_j^f N_i^s \mathbf{T}_{fi}^\Gamma d\Gamma = 0 \tag{5.24}$$

The derived integral equations are computed using Gaussian integration analogously to the structural dynamics in section 3.7.2.3.

5.6.2 Time Advancement

The single fields of CSD and CFD use the time discretization and integration that has been introduced in the respective chapters. For the CFD, the BDF2 scheme is used without exception in this chapter.

A model process for a Gauss-Seidel communication pattern using fix-point iteration [64, 101] is shown in figure 5.3. In this figure, an iterative Gauss-Seidel scheme is shown for the time progression reaching t_{n+1} starting from a time t_n. S_1 herein represents the CSD, and S_2 the CFD solution process. This could be the other way around, as well. The arrows denote the transfer of discrete field information using the Mortar mapping method presented above. More precisely, an arrow from S_1 to S_2 is a displacement mapping according to equation 5.4 respectively 5.23, and an arrow from S_2 to S_1 is a traction force mapping according to equation 5.6 respectively 5.24. The increasing numbering of the arrows denote the sequence of the operations during the process. The Gauss-Seidel solution process is iterated until the IICE convergence criterion of the iterative coupling process is reached at the final iteration k. As already mentioned, the Dirichlet-Neumann partition of the FSI uses force elimination. The force elimination implies that the computed forces at the fluid surface are directly mapped to the structural surface (e.g., within arrow number 1,5, 9, and 13 in figure 5.3). Using iterative coupling, the computed displacements of the structure are transfered to the fluid using an Aitken under-relaxation (e.g., within arrows number 3,7,11, and 15 in figure 5.3) [36, 64].

Using a loose coupling process as an alternative instead, arrow number 5 directly enters S_2 at $t = t^{n+1}$ (cf. figure 5.4). In consequence no sub-iterations k and no under-relaxation are performed between S_1 and S_2.

5.7 Formal Order of Accuracy

The errors of an FSI simulation in space and time result from three different parts outlined in the following list:

- - Errors from the CSD
- - Errors from the CFD
- - Errors from the interface operations

Therefore, the formal order of the complete FSI process consist of the formal orders of the three named parts. The formal order of accuracy of the CSD software Carat++ and the CFD software OpenFOAM®have been discussed in detail in sections 3.7 respectively 4.11. The formal order of accuracy in space due to the mapping procedures and the formal order of accuracy over time due to the time progression are presented in the following sections.

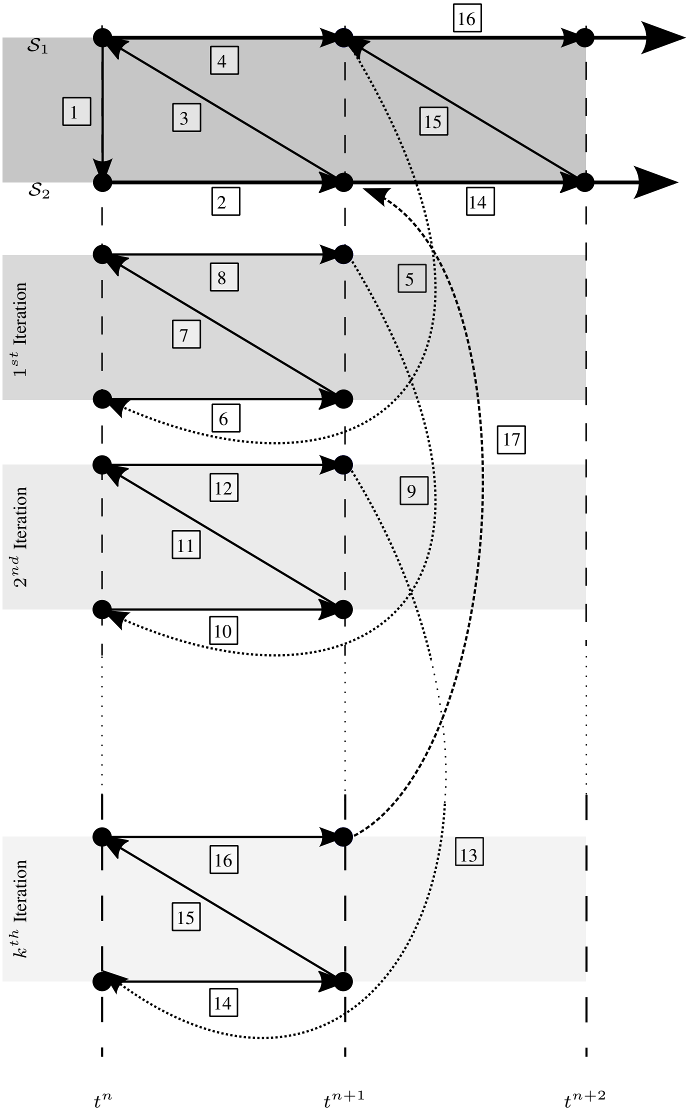


Figure 5.3: Gauss-Seidel communication pattern for iterative coupling [101]

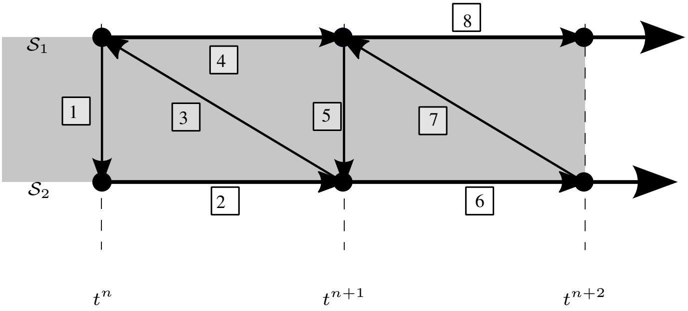


Figure 5.4: Gauss-Seidel communication pattern for loose coupling

5.7.1 Spatial Approximation at the Interface

The Mortar method used in the available environment uses linear shape functions for the approximation of the geometry, the solution fields, and the test functions. The surface integration is performed using element-wise Gaussian integration corresponding to 3.7.2.3. For non-matching grids, the consistent clipping procedure is used [87]. In consequence the spatial approximation of the mapping procedure of equations 5.23 and 5.24 is formally second order accurate ($\hat{p}_s = 2$).

Besides the spatial approximations of the mapping procedures, the evaluation of the surface traction forces at the interface needs to be mentioned (cf. equation 5.7). The common interface in the context of FSI normally is represented by a non-permeable moving wall. The boundary conditions for the pressure and the absolute velocities are modeled as a zero gradient Neumann pressure BC and a Dirichlet velocity BC, with a relative velocity to the wall equal to zero. Consequently the fluid at the wall moves with the same velocity as the wall itself. For the interface traction force calculation (cf. equation 5.7), the pressure and the velocity gradient at the interface need to be extrapolated. This extrapolation is performed using an upwind discretization from the cell midpoint to the surface. As presented in section 4.9.1.2, the upwind procedure is formally first order accurate ($\hat{p}_s = 1$).

5.7.2 Time Advancement of the FSI Process

For the time advancement, the error depends on the coupling procedure, whether loose or strong coupling are applied (cf. section 5.1).

For a strongly coupled process, the error at the interface is controlled by the criterion for the convergence of the iterative process (cf. figure 5.3). The nature of this error is equivalent as the error of IICE already discussed in the previous chapters. This error must be kept very low in order to finally observe the discretization error as the leading error of the simulation (cf. figure 2.4). As long as this is guaranteed, no additional noticeable error is introduced by the IICE. Following the error analysis of the PISO procedure as a partitioned solution of a problem [37, 47, 48, 55], the time error introduced by a partitioned solution process can be evaluated by the so-called one-step error analysis shown in [67]. Therefore the order of the introduced error is reducing linearly with the number of iterations starting at $\mathcal{O}(\Delta t)$. To give an example this means that for two iteration steps the error is reduced to $\mathcal{O}(\Delta t^3)$. This knowledge can save a lot of time for an efficient simulation. Beside, the error of a loose coupling process is directly identified as $\mathcal{O}(\Delta t)$. An overview of all contributing formal orders of accuracy is outlined in table 5.1. Spatial rectangular means that the grid stays rectangular during the whole simulation; therefore it is a special case of the general spatial behavior. As mentioned above, the time accuracy of the FSI time advancement depends on the iterations of the iterative coupling process and is at least first order accurate.

Table 5.1: Formal orders of accuracy $\hat{p}$ of the FSI process

	Spatial general	Spatial rectangular	Time
Structure	2	2	2
Fluid	1	2	1
Coupling & Time Advancement	2	2	>1
Overall FSI Process	1	2	1

In table 5.1 it can clearly be seen that again the weakest link of the chain sets the formal order of accuracy for the overall FSI process. Therefore, the weaker accuracy of the used CFD software in non-orthogonal grids and the time discretization lowers the formal order of accuracy of the total FSI process in general to $\hat{p} = 1$ in space and time. The higher accuracy of the CSM and the coupling processes do not improve the results and the accuracy in the asymptotic range of the solution, as the leading error is determined by the weakest part. From this observation it is obvious that a homogenization of the formal orders $\hat{p}$ in space and time would maximize the solution accuracy with respect to computational costs.

5.8 Application of the MMS

5.8.1 Equilibrium Source Terms

In general, it is possible to enrich both interface equations (cf. equations 5.4 and 5.6) with a source or a force term analogously to the source/force terms for the equations of momentum and mass conservation of the structure and the fluid respectively. In the present thesis the choice has been made that the compatibility condition at the common interface (equation 5.4) should be fulfilled during the MMS simulation. Consequently both grids are fixed together during the simulation and no source term will be applied for this equation.

Therefore, only the balance of forces equation will be enriched with a manufactured force term at the interface. It is shown later in section 5.9.4 that it is in general possible to fulfill both interface equations by a smart choice of the field variables in the fluid and the structure. As the current environment provides a very comfortable access to the interface equations, this effort is not taken for the sake of flexibility of the manufactured solutions. The manufactured traction force term of equation 5.6 is shown in 5.25:

$$\hat{\mathbf{t}}^{\gamma}_{FSI} = \hat{\mathbf{t}}^{\gamma}_{s} + \hat{\mathbf{t}}^{\gamma}_{f} \tag{5.25}$$

Using the strong form of equation 5.16, the manufactured traction force term can be re-written in terms of the initial configuration in equation 5.26:

$$\hat{\mathbf{T}}^{\Gamma}_{FSI} = \hat{\mathbf{T}}^{\Gamma}_{s} + \hat{\mathbf{T}}^{\Gamma}_{f} \tag{5.26}$$

$\hat{\mathbf{T}}^{\Gamma}_{s}$ is the manufactured traction at Γ of the structure. It is directly computed analogously the manufactured traction for the structural dynamics part of chapter 3 in equation 3.39. $\hat{\mathbf{T}}^{\Gamma}_{f}$ represents the manufactured traction of the fluid, which is evaluated with $\hat{\mathbf{u}}$ and $\hat{\mathbf{P}}$ using equation 5.27 (cf. chapter 4):

$$\hat{\mathbf{T}}^{\Gamma}_{f} = \left(-\rho_f \hat{P}\mathbf{I} + 2\nu_m \rho_f \hat{\mathbf{D}}\right) \cdot \hat{\mathbf{n}}^{\gamma}_{f} \cdot \det(\hat{\mathbf{F}}) \tag{5.27}$$

$\hat{\mathbf{T}}^{\Gamma}_{FSI}$ can be interpreted as the additional traction term of the interface to fulfill the equilibrium of the traction forces at Γ. It is termed 'additional' since it is combined with the surface traction of the fluid $\hat{\mathbf{T}}^{\Gamma}_{f}$. It should be noted that for the design of the MMS benchmarks, the source term $\hat{\mathbf{T}}^{\Gamma}_{FSI}$ should not dominate the actual force from the fluid $\hat{\mathbf{T}}^{\Gamma}_{f}$ in order to really assess the full FSI procedure. If $\hat{\mathbf{T}}^{\Gamma}_{FSI}$ dominates the total force acting on the structure, errors due to the computation or mapping of the fluid forces would be overpowered and therefore these errors would possibly not be detected.

5.8.2 Framework

The framework to assess the partitioned FSI environment using EMPIRE is presented in this section. In total, six software components are used in this framework:

- Carat++ (already presented in chapter 3)
- OpenFOAM®(already presented in chapter 4)
- Maple®
- mmsFsiClient
- Emperor
- Matlab®

An overview of the framework is presented in figure 5.5. The procedure and the sequence of the program executions is shown in figure 5.6.

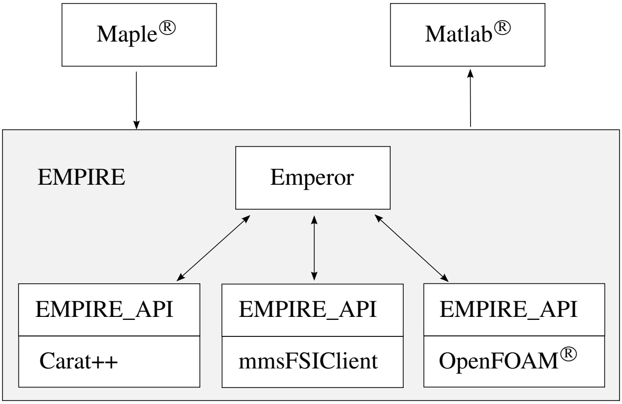


Figure 5.5: Overview of the MMS framework for the assessment of the partitioned FSI environment

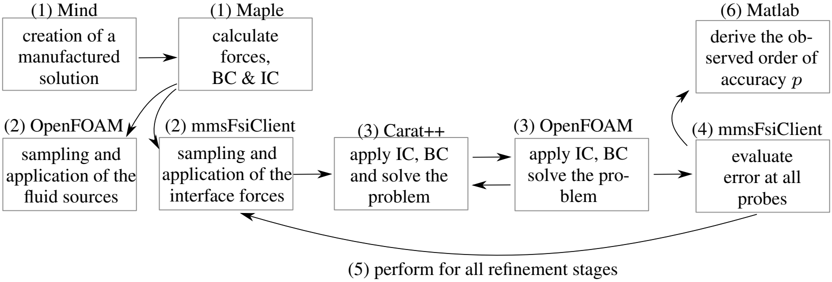


Figure 5.6: The MMS procedure for partitioned FSI analysis

5.8.2.1 Maple®

The Maple® code consists of three parts that are explained in the following.

The first part is the derivation of the manufactured solution of $\hat{\mathbf{d}}$ for the structural solution. This manufactured solution contains the boundary conditions, the initial conditions, and the required force term $\hat{\mathbf{T}}_s$ to compute $\hat{\mathbf{d}}$ asymptotically using Carat++. If the total membrane surface is attached to Γ, the derived source term of the CSD (cf. equation 3.39) matches $\hat{\mathbf{T}}_s^{\Gamma}$ ($\hat{\mathbf{T}}_s = \hat{\mathbf{T}}_s^{\Gamma}$). The detailed procedure is derived in chapter 3.

The second part is the derivation of the manufactured solution of $\hat{\mathbf{u}}$ and $\hat{P}$ for the fluid solution. This contains the boundary conditions, the initial conditions, and the

required source terms $\hat{\mathbf{s}}_f$ to compute $\hat{\mathbf{u}}$ and $\hat{P}$ asymptotically using OpenFOAM®. The detailed procedure is derived in chapter 4.
The third part is the derivation of the manufactured FSI traction $\hat{\mathbf{T}}^{\Gamma}_{FSI}$ at the common interface to satisfy the balance of forces at Γ (cf. equation 5.6) using equation 5.26. The last traction term $\hat{\mathbf{T}}^{\Gamma}_{f}$ is computed using equation 5.27. As already mentioned above the kinematic condition is directly fulfilled by the definition that the grids are attached to each other at Γ. Therefore, no source term is necessary for the compatibility condition.

5.8.2.2 mmsFsiClient

The mmsFsiClient is an own created auxiliary tool to apply the generated loads $\hat{\mathbf{T}}^{\Gamma}$ to Carat++ and to receive the computed displacement field $\mathbf{d}$. Additionally, it manages the FSI procedure and its time advancement approach (cf. section 5.6.2). For the application of source terms for Carat++, a mapping with matching grids is used to avoid the introduction of an additional mapping error.
The procedure for the application (sampling and mapping) of the forces to Carat++ is already shown in the explanations of the mmsClient in section 3.8.3.2.

As the FSI process is managed by this software, the mapping of the interface fields $\mathbf{d}^{\Gamma}$ and $\mathbf{T}^{\Gamma}$ is in general performed for non-matching grids. Additionally, the implementation and the management of loose and iterative coupling is performed inside this software.

The sampling procedure, the integration, and the mapping at non-matching grids introduce additional errors to the FSI simulation. As similarly presented in chapter 3, these schemes have to be at least the order of accuracy of the actual discretization of the single fields. As (at least) Carat++ is formally second order accurate in space and time (cf. section 3.7.4 and table 5.1), the schemes at the interface must at least be formally second order accurate.

The evaluation of the structural field errors is performed the same way as already described for the mmsClient in section 3.8.3.2 inside this software.

5.8.2.3 Emperor

Emperor provides procedures to perform sampling, mapping and integration of fields between the different contributing software components. In our case the discrete solutions of equations 5.4 and 5.6 using the Mortar method are performed inside Emperor.

5.8.2.4 Matlab®

The software Matlab® is used to finally apply the error norms 2.7 and 2.8 on the resulting deviations of the fields $\mathbf{d}$, $\mathbf{T}^{\Gamma}_{f}$, $\mathbf{u}$ and P at the probe locations. Using these errors and the named error norms, graphs are illustrated for the error development in a log-log diagram along with a diagram for the observed order of accuracy (cf. figure 5.20).

5.8.3 Procedure

The procedure applying the MMS for the partitioned FSI process is elaborated in the following. The balance equations are based on the momentum equilibria of the fluid and the structure (cf. equations 5.2, 5.3, and 5.1) and the interface equations of the Dirichlet-Neumann coupling (cf. equations 5.4 and 5.6).
A detailed description of the refinement strategies for steady state or unsteady analyses has been presented in 2.3.3.5 respectively 2.3.3.6.
The procedure is summarized in the following list and is additionally shown in figure 5.6.

1. Creation of a manufactured solution (cf. section 2.3.3) using Maple®
    - Creation of manufactured field solutions $\hat{\mathbf{d}}$, $\hat{\mathbf{u}}$ and $\hat{P}$
    - Creation of all contributing constants
    - Derivation of the traction forces $\hat{\mathbf{T}}_s$ resp. $\hat{\mathbf{T}}_s^{\Gamma}$ using equation 3.37
    - Derivation of the interface traction forces $\hat{\mathbf{T}}_{FSI}^{\Gamma}$ using equation 5.26
    - Derivation of the sources $\hat{\mathbf{s}}_f$ using equations 4.73-4.79
    - Derivation of the initial condition using equations 3.42 and 4.83
    - Derivation of the boundary conditions using equations 3.41, 4.81-4.82
2. Application of the MMS terms
    - Application of the interface forces $\hat{\mathbf{T}}_{FSI}^{\Gamma}$ at the mmsFsiClient
    - Application of the sources $\hat{\mathbf{s}}_f$ in OpenFOAM®
    - Application of the initial condition in Carat++ and OpenFOAM®
    - Application of the boundary conditions in Carat++ and OpenFOAM®
3. Performing the FSI simulation
4. Field error evaluation
    - Sampling of $\mathbf{d}$, $\mathbf{T}_f^{\Gamma}$, $\mathbf{u}$ and P at the probe locations
    - Error estimation with $\hat{\mathbf{d}}$, $\hat{\mathbf{T}}_f^{\Gamma}$, $\hat{\mathbf{u}}$ and $\hat{P}$ using equations 2.7 resp. 2.8
5. Repetition of steps 2-4 with systematical refinement
6. Calculation of the observed order of accuracy p for the fields $\mathbf{d}$, $\mathbf{T}_f^{\Gamma}$, $\mathbf{u}$ and P using Matlab®
    - Comparison of the error evaluations at each refinement step
    - Derivation of p with refinement using equation 2.13
7. Comparison of the formal order of convergence $\hat{p}$ to the observed order of convergence p
8. Assessment of the FSI environment
    - If p matches $\hat{p}$, all ordered functionalities are working as intended
    - If p does not match $\hat{p}$, the error source has to be investigated

5.8.4 Spatial and Time Resolutions

As already mentioned in chapter 2, the MMS deals with systematic refinement in space and time. The quantities of grid size and resolution in general is almost arbitrary. The presented numbers are only a suggestion for the reader. With these numbers the later shown results have been produced. For the reason of limited computational resources, the benchmarks for CFD generally are elaborated for two dimensional computations. To guarantee the functionality in 3D, the 2D domain of the examples is rotated in space and the tests are repeated. Finally, benchmark 3 of this chapter represents a fully 3D FSI benchmarks to complement the benchmark sequence.

As elaborated in sections 2.3.3.5 and 2.3.3.6, the set of sampling points of a single MMS study for the fields $\mathbf{d}$, $\mathbf{T}_f^\Gamma$, $\mathbf{u}$ and P has to be kept constant. In the context of this work, all nodal and time positions of the coarsest stage are used as sampling points, independently of the individual grid or time resolution.
The coarsest CSD grid, which in parallel defines the sampling points for the complete assessment, is chosen as a quadrilateral grid using 8 by 8 elements (cf. figure 3.6). As the complete structure is attached to the interface, the error of the surface traction forces is evaluated at the probe locations of the CSD as well.
The cell midpoints of the coarsest CFD grid using 8 by 8 cells, define their sampling points in space (cf. figure 4.4). In unsteady simulations, 8 timesteps are chosen as coarsest resolution, which in parallel defines the sampling points over time.
The refinement in space and time is constantly chosen to $r_s = r_t = r = 2$.

5.8.5 Solution Accuracy

Recalling section 2.3.2, the discretization error should be isolated from the total numerical error (cf. figure 2.4). Therefore, all software components (cf. figure 5.6) are using double precision accuracy to minimize the round-off error. In order to keep the IICE as small as possible, the solution tolerance in a non-normalized version of the E2 norm of velocities and pressures (cf. equation 2.8) of the unsteady SIMPLE projection method is set to $1.0 \cdot 10^{-11}$. The solution tolerance in the inf norm of displacements in Carat++ (cf. equation 2.8) of the nonlinear equation system (using a Newton-Raphson procedure) is set to $1.0 \cdot 10^{-14}$. In iterative coupling procedures, an interface residuum tolerance in the non-normalized E2 norm is chosen to $1.0 \cdot 10^{-10}$.

5.8.6 Error Map

In this section, an overview of the errors in the FSI simulation is elaborated. On the one hand, the general types of errors are described. The errors arise in the processes of preprocessing of input data, during the solution of the governing equations, and in the postprocessing of output data of the simulation. For the application of the present software Carat++, the general sources of error and the specific error sources of the software are shown in figure 5.7 left and right.
It is obvious that errors in the generation of input data will influence the total simulation process. This indicates that the generation and application of the MMS procedure (cf. figure 5.6) need to be at least as accurate as the solution of the simulation itself. In the present case, the actual simulation of the fluid, the structure and the coupling processes provide a formal order of accuracy between $p = 1$ and $p = 2$. Therefore, the generation and application of the MMS sources and forces, the BC, and the IC need to be at least second order accurate as well. Additionally, the output generation, and the error evaluation in Matlab need to be at least second order accurate, too. If this is not fulfilled, the total observed order of accuracy of the simulation will lower to the accuracy of the weakest link of the chain.

Preprocessing of input data		- Human errors in the preparation of input data - Generation of input data *
Numerical errors due to the solution of the governing equations	Loop for Iterative coupling	- Mapping of $\mathbf{d}^\Gamma$ (solver for the linear equation system for $\mathbf{d}_f^\Gamma$) * - Solution of the CFD * - Mapping of $\mathbf{t}^\Gamma$ (solver for the linear equation system for $\mathbf{t}_s^\Gamma$) * - Solution of the CSM * - Gauss-Seidel IICE *
Postprocessing of output data		- Generation of output data * - Human errors in the postprocessing of simulation results

* contains round-off

Figure 5.7: Error map of the presented FSI environment

5.8.7 Benchmarks

The following benchmark sequence for FSI is developed as a stairway in complexity. The proposed benchmarks suite starts with the assessment of restricted functionality of the code and increases more and more in complexity through the benchmark sequence (cf. figure 5.8). All functions and variables of the following examples are listed in tables, such that the reader is able to construct the individual force and source terms, and the boundary as well as the initial conditions. The choice of the BC type (Dirichlet or Neumann) on the boundaries in general is left to the reader. The only restriction is set for the boundaries at the common interface Γ. Here, a non-permeable wall with a no-slip condition is set for the velocities, i.e. the absolute velocity is equal to the grid velocity ($\mathbf{u}^\Gamma = \mathbf{u}_g^\Gamma$). The same holds for the displacements at the common interface Γ.

The input parameters of the benchmarks are all defined in the International System of Units (SI) and their derived expressions [81]. Therefore, units for the input numbers of the benchmarks are generally omitted.

5.8.8 Benchmark 1: Sampling and Mapping Operations

As described above for (vector) field sampling exactly the same operations are performed for all parameters, be it force, displacement or velocity field. Therefore, one analytical manufactured field solution represents a representative benchmark for the mapping of all types of the used fields.

As the sampling and mapping are purely geometric operations, a steady solution of

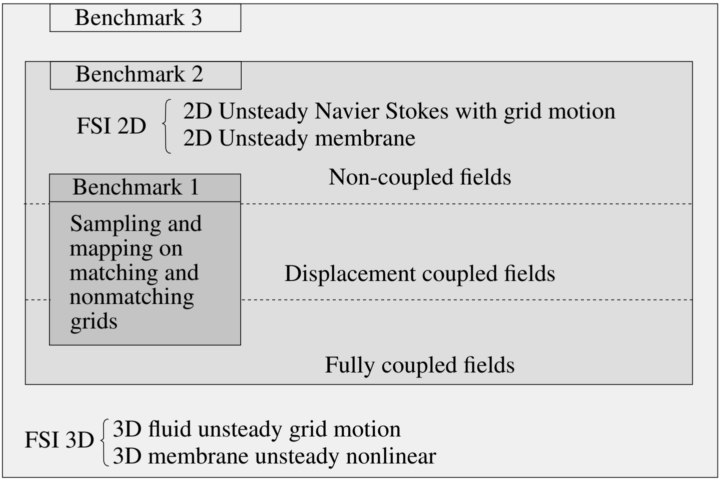


Figure 5.8: Hierarchical benchmark sequence for FSI

all types of fields used in the FSI analysis is sufficient to be benchmarked. For this assessment, only a part of the FSI process is necessary. In detail, the solution of the fluid is cut off, and the interface source force $\hat{\mathbf{T}}^{\Gamma}_{FSI}$ is used to apply the traction force $\hat{\mathbf{T}}$.
Table 5.2 gives the overview of this benchmark of traction sampling, integration, and mapping on an initially curved geometry.

Table 5.2: Benchmark 1: Overview table

init. config.	field	domain
$x = \theta^1$	$\hat{T}_x = 4 \cdot \sin\left(\theta^1 \pi\right) \cdot \cos\left(\theta^2 \pi\right)$	$\theta^1 \in [0..1]$
$y = \theta^2$	$\hat{T}_y = \sin\left(\theta^1 \pi\right) + \cos\left(\theta^2 \pi\right)$	$\theta^2 \in [0..1]$
$z = \theta^1 - \theta^{1^2}$	$\hat{T}_z = 16 \cdot \sin^2\left(\theta^1 \pi\right) \cos^2\left(\theta^2 \pi\right)$	

The traction forces $\hat{\mathbf{T}}$ of the benchmark are shown in the top view of the geometry in figure 5.9.

Using this benchmark, different options for the assessment are possible:

- Matching grids
- Non-matching grids with one very fine grid
- Non-matching grids with a similar grid resolution

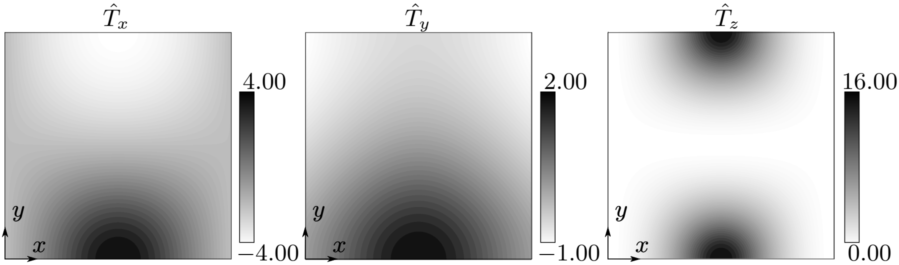


Figure 5.9: Traction field $\hat{\mathbf{T}}$ of benchmark 1

5.8.8.1 Matching Grids

The matching grid assessment contains on both sides of the mapping exactly the same grid. Therefore, both are refined the same way as well. As the grid points are coincident, the discretization error due to mapping must vanish. The log-log plot of this assessment is shown in figure 5.19. The numerical error completely is in the range of machine accuracy which confirms the expectation of vanishing discretization error.

As the IICE condition for the interface equations are defined in the non-normalized E2 norm, the error produced by the IICE is increasing with the number of cells due to refinement.

5.8.8.2 Non-Matching Grids with one very fine Grid

In this section, the receiving grid is chosen to be a superfine grid and the other grid is systematically refined. The results are illustrated in figure 5.20.

5.8.8.3 Non-Matching Grids with a similar Grid Resolution

The second option to assess non-matching grids is to generate two grids with a similar number of elements with non-matching grid points/elements. This is performed using different grids with comparable grid resolutions. As the general procedure is starting with a grid resolution of 8 by 8 elements, the coarser grid now starts with a grid resolution of 7 by 7 elements and is also refined by a factor of $r_s = 2$. The results are illustrated in figure 5.21.

5.8.8.4 Statement and Results

The assessed sampling, mapping, and integration schemes at the interface level assessed in this section are positively assessed and give an observed order of accuracy of $p \approx 2$, which corresponds to the formal order of accuracy of $\hat{p} = 2$. Thus one can conclude that no remarkable additional error is introduced in the FSI process due to mapping (cf. figure 5.7).

5.8.9 Benchmark 2: The way to FSI

This benchmark has the task to design a fully coupled benchmark for the assessment of the FSI environment. As shortly mentioned above, the structure and the fluid are coupled by the structural displacements $\mathbf{d}$ and the surface traction forces $\mathbf{T}^{\Gamma}$ at the common interface Γ. This implies for a fully coupled FSI process, the computed displacements are transferred to the fluid as a boundary motion and the computed boundary surface traction forces of the fluid are transferred to the structure as a boundary surface load. In terms of manufactured solutions, the processes S_1 and S_2 of one Gauss-Seidel iteration (cf. figure 5.3) are defined in the following list of equations:

$$S_2 = S_2(\hat{\mathbf{s}}_f^{mom}, \hat{\mathbf{s}}_f^{mass}, \mathbf{d}) \tag{5.28}$$

$$S_1 = S_1(\hat{\mathbf{T}}_s, \hat{\mathbf{T}}_{FSI}^{\Gamma}, \mathbf{T}_f^{\Gamma}) \tag{5.29}$$

Here the coupling of the two fields can clearly be identified since $\mathbf{d}$ is the computed displacement solution of S_1, and $\mathbf{T}_f^{\Gamma}$ is the computed surface traction solution of S_2.

Still following the concept of increasing complexity for the assessment of the FSI environment, the process starts with a simulation of uncoupled fields towards a fully coupled simulation in the following three subsections.

The benchmark overview is presented in table 5.3:

Table 5.3: Benchmark 2: Overview table

fields	material	domain
$\hat{d}_x = 0$	$\rho_s = 1 \cdot 10^6$	$x \in [0,1]$
$\hat{d}_y = \sin(\pi x) \cdot \sin(\pi t) \cdot t^3$	$E = 1 \cdot 10^6$	$y \in [0,1]$
$\hat{u}_{gx} = 0$	$B = 1 \cdot 10^{-3}$	$t \in [0,1]$
$\hat{u}_{gy} = \frac{\partial \hat{d}_y}{\partial t}$	$S_{ps1,2} = 25$	
$\hat{u}_x = 0$	$\nu_s = 0.3$	
$\hat{u}_y = u_{gy}$	$\rho_f = 1000$	
$\hat{P} = -0.5 + 10 \cdot \cos(x+y) \cdot \sin(\pi t) \cdot \left[y - \hat{d}_y\right]^2$	$\nu_m = 1 \cdot 10^{-5}$	
	$\Gamma_g = 1.0$	

The tractions in y direction of the benchmark are shown in figures 5.10-5.11. The initial and the deformed structure is shown in figure 5.12.
The fluid and the structure have their common boundary Γ initially at $\mathbf{x}^{\Gamma} = \mathbf{x}(x, y = 0)$. The set of boundary conditions are shown in figure 5.13. The current benchmark is an initially flat and prestressed membrane in the x-z-plane. As it is actually a 2D benchmark, the manufactured fields in z-direction are kept constant and therefore do not influence the solution. The membrane is deformed in y-direction. As the fluid itself has already been assessed in chapter 4, the fluid fields are kept relatively

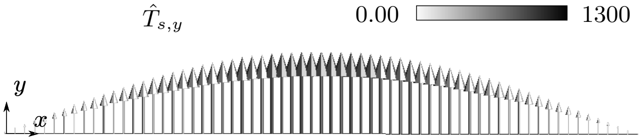


Figure 5.10: Traction field $\hat{T}_{s,y}$ at $t = 0.2$ of benchmark 2

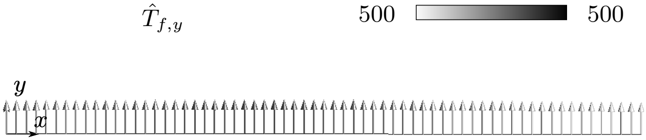


Figure 5.11: Traction field $\hat{\mathbf{T}}_{f,y}$ at $t = 0.2$ of benchmark 2

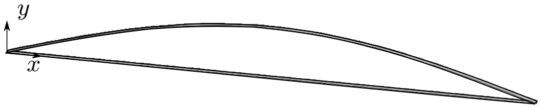


Figure 5.12: Initial and final geometry of the structure of benchmark 2 - scaled with a factor of 20

simple. The main interest herein is in the surface coupling between the fluid and the structure. These fields are sufficiently complex to be considered a valid benchmark for the FSI environment. This benchmark also satisfies the demand that the fluid tractions $\hat{\mathbf{T}}_f^\Gamma$ have the same magnitude as the demanded structural tractions $\hat{\mathbf{T}}_s$.

5.8.9.1 Benchmark 2: Non-Coupled Fields

In this section, the fluid and the structural fields are solved completely separated. This is necessary in order to assess whether the created benchmark can in general be solved by the individual software components of the fluid and the structure. Therefore, in this section, equations 5.28 and 5.29 change to 5.30 and 5.31:

$$S_2 = S_2(\hat{\mathbf{s}}_f^{mom}, \hat{\mathbf{s}}_f^{mass}, \hat{\mathbf{d}}) \tag{5.30}$$

$$S_1 = S_1(\hat{\mathbf{T}}_s, \hat{\mathbf{T}}_{FSI}^\Gamma, \hat{\mathbf{T}}_f^\Gamma) = S_1(\hat{\mathbf{T}}_s) \tag{5.31}$$

As the boundary displacements of the fluid $\mathbf{d}$ and the surface traction of the structure $\mathbf{T}_s$ are completely defined as a manufactured solution $\hat{\mathbf{d}}$ respectively $\hat{\mathbf{T}}_s$, the solution fields are no longer dependent or coupled. As no coupling dependency exists, the results are identical using a loose coupling or an iterative coupling procedure. Figures 5.22-5.24 show the plots of convergence for all variable fields.

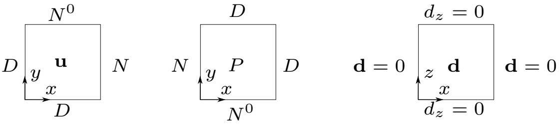


Figure 5.13: Benchmark 2: Boundary conditions

5.8.9.2 Benchmark 2: Coupled Displacement Field

In the next consecutive step the computed displacements of Carat++ are applied respectively mapped to OpenFOAM®. The traction forces are still replaced by the manufactured tractions identical to the previous section. Therefore, this step only represents a one-way coupling of displacements from the structure to the fluid. In terms of the solution of the fields S_1 and S_2, the equations change to 5.32 and 5.33:

$$S_2 = S_2(\hat{\mathbf{s}}_f^{mom}, \hat{\mathbf{s}}_f^{mass}, \mathbf{d}) \tag{5.32}$$

$$S_1 = S_1(\hat{\mathbf{T}}_s, \hat{\mathbf{T}}_{FSI}^{\Gamma}, \hat{\mathbf{T}}_f^{\Gamma}) = S_1(\hat{\mathbf{T}}_s) \tag{5.33}$$

The process of the fluid solution S_2 therefore is solved in its final way within the FSI process (cf. equations 5.32 and 5.28).

The results for loose and iterative coupling are shown in figures 5.25-5.27 respectively figures 5.28-5.30.

In this case, the loose coupling and the iterative coupling must lead to identical results for the structural displacements $\mathbf{d}$ despite the coupling of displacements. This expectation is based on the fact that S_1 (Carat++) solves for $\mathbf{d}$ independently of the computed fluid traction. Therefore Carat++ in every iteration step provides the same computed solution $\mathbf{d}$ as in a loose coupling procedure (compare the similarity of figures 5.25 and 5.28).

5.8.9.3 Benchmark 2: Fully Coupled

The third step in the increasing complexity for the FSI environment is the complete coupling of the fields identical to a regular FSI simulation. In terms of the procedures of S_1 and S_2, this reads as the initial equations 5.28 and 5.29, for the sake of completeness repeated in equations 5.34 and 5.35.

$$S_2 = S_2(\hat{\mathbf{s}}_f^{mom}, \hat{\mathbf{s}}_f^{mass}, \mathbf{d}) \tag{5.34}$$

$$S_1 = S_1(\hat{\mathbf{T}}_s, \hat{\mathbf{T}}_{FSI}^{\Gamma}, \mathbf{T}_f^{\Gamma}) \tag{5.35}$$

Compared to the previous section, the surface tractions computed by OpenFOAM® are now mapped to Carat++, i.e. the interface traction force field $\mathbf{T}_{s,total}^{\Gamma}$ is applied to Carat++ (cf. equation 5.36).

$$\mathbf{T}_{s,total}^{\Gamma} = \hat{\mathbf{T}}_{FSI}^{\Gamma} - \mathbf{T}_f^{\Gamma} \tag{5.36}$$

In consequence $\mathbf{T}^{\Gamma}_{total}$ is a traction force, consisting on the one hand of a manufactured traction $\hat{\mathbf{T}}^{\Gamma}_{FSI}$ and on the other hand of a discrete computed traction $\mathbf{T}^{\Gamma}_{f}$.

The results for loose and iterative coupling are shown in figures 5.31-5.33 respectively 5.34-5.36.

5.8.9.4 Benchmark 2: Fully Coupled Non-Matching Grids 1 (NMG1)

In this section, the benchmark is applied to non-matching grids. Here, the CSD grid is always refined one more step than the CFD grid. Consequently at the coarsest level the CFD grid has 8 elements per edge whereas the corresponding CSD grid has 16 elements per edge. Both are then refined with a factor of $r = 2$ in space and time.

The results of the iterative coupled simulations are shown in figures 5.37-5.39.

5.8.9.5 Benchmark 2: Fully Coupled Non-Matching Grids 2 (NMG2)

In this section, the benchmark is again applied to non-matching grids. In contrast to the previous section, the CFD grid now is always refined one more step than the CSD grid. In consequence at the coarsest level the CSD grid has 8 elements per edge whereas the corresponding CFD grid has 16 elements per edge. Both are then refined with a factor of $r = 2$ in space and time. Therefore this section can be seen as the counterpart to the previous section.

The results of the iterative coupled simulations are shown in figures 5.40-5.42.

5.8.9.6 Benchmark 2: Fully Coupled Non-Matching Grids 3 (NMG3)

In this section, the benchmark is again applied to non-matching grids. The grid generation matches the grids of the non-matching mapping benchmark shown in section 5.8.8.3. The coarsest grid of the CSD is chosen to 8 elements per edge, and the coarsest grid of the CFD is chosen to 7 elements per edge. Therefore, almost no grid points coincide during the refinement. Both grids are refined with a factor of $r = 2$ in space and time.

The results of the iterative coupled simulations are shown in figures 5.43-5.45.

5.8.9.7 Statement and Results

The presented Benchmark 2 gives a full 2D benchmark for the partitioned FSI environment. The results are satisfying, as the observed order of accuracy in space and time matches the formal order of accuracy of $\hat{p} = 1$ in space and time.

5.8.10 Benchmark 3: FSI 3D

In order to complete the benchmark sequence for FSI a fully three dimensional benchmark is used to prove the expandability of the previously discussed two dimensional benchmarks. Following the concept of increasing complexity for the

assessment of the FSI environment, this benchmark presents the closure of the designed benchmark sequence for the FSI environment.
The benchmark overview is presented in table 5.4. The fluid and the structure have their common boundary Γ at $\mathbf{x}^{\Gamma} = \mathbf{x}(x, y = 0, z)$. The set of boundary conditions is shown in figure 5.14. The elaborated benchmark is an initially flat and prestressed membrane in the x-z-plane which is deformed in y-direction and the fluid is fully developed in 3D. It should be noted that this benchmark also satisfies the demand for the same magnitude of the computed fluid tractions $\mathbf{T}_f^{\Gamma}$ and the prescribed structural tractions $\hat{\mathbf{T}}_s$.

Table 5.4: Benchmark 3: Overview table

fields	material	domain
$\hat{d}_x = 0$	$\rho_s = 5 \cdot 10^5$	$x \in [0,1]$
$\hat{d}_y = \sin(\pi x) \cdot \sin(\pi t) \cdot t^2$	$E = 1 \cdot 10^6$	$y \in [0,1]$
$\hat{d}_z = 0$	$B = 1 \cdot 10^{-3}$	$z \in [0,1]$
$\hat{u}_{gx} = \frac{\partial \hat{d}_x}{\partial t} = 0$	$S_{ps1,2} = 25$	$t \in [0, 0.2]$
$\hat{u}_{gy} = \frac{\partial \hat{d}_y}{\partial t}$	$\nu_s = 0.3$	
$\hat{u}_{gz} = \frac{\partial \hat{d}_z}{\partial t} = 0$		
$\hat{u}_x = u_{gx} + \sin(x+y+z) \cdot [y - \hat{d}_y]$	$\rho_f = 1000$	
$\hat{u}_y = u_{gy}$	$\nu_m = 0.5$	
$\hat{u}_z = u_{gz} + \sin(x+y+z) \cdot [y - \hat{d}_y]$	$\Gamma_g = 1.0$	
$\hat{P} = (\cos(x+y) + \sin(z)) \sin(\pi t) \cdot \left[y - \hat{d}_y\right]^2 - \frac{1}{2}$		

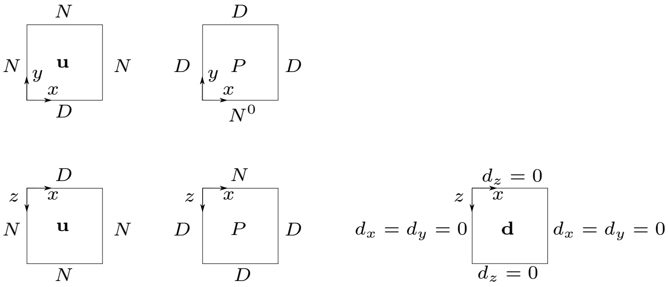


Figure 5.14: Benchmark 3: Boundary conditions

To guarantee the compatibility condition at the common interface Γ (cf. equation 5.4), the fluid velocities in x and z direction are set such that this requirement is fulfilled. This procedure is adapted from [73]. As $\hat{d}_y$ describes the position of the interface at every time in the global y-direction, the term $[y-\hat{d}_y]$ modifies the actual fields. With 'actual', e.g., the $\sin(x+y+z)$ field of the velocities $\hat{u}_x$ and $\hat{u}_z$, is meant. For $y=\hat{d}_y$ the term in brackets is canceled out and therefore the velocity at the bottom boundary becomes equal to the grid velocity field, which in the present case is zero for $\hat{u}_x$ and $\hat{u}_z$. In terms of boundary conditions, the multiplication with the named bracket term creates a zero Dirichlet condition for the relative velocities at the interface Γ. Furthermore, the square of the term in brackets, $[y-\hat{d}_y]^2$, creates a zero Neumann boundary condition of the 'actual' field as well, besides the already mentioned zero Dirichlet boundary condition. In the current benchmark, a zero Dirichlet condition is set for the relative velocities and a zero Gradient condition is set for the pressure at the interface. As this is equal to the regular modeling of a non-permeable interface wall for FSI, this benchmark can be considered representative for FSI problems.

The traction forces of the benchmark are shown in figures 5.15-5.16. The deformed structure is shown in figure 5.17. The fluid velocities $\hat{\mathbf{u}}$ and the pressure $\hat{P}$ are shown in figure 5.18. In figure 5.18 one can see that the (absolute) velocities in x- and z-direction are zero at γ (bottom), and the y-velocity of the bottom is equal to the velocity of γ and therefore equal to the velocity of the structure.

The results of the error evaluation of this benchmark are illustrated in figures 5.46-5.48.

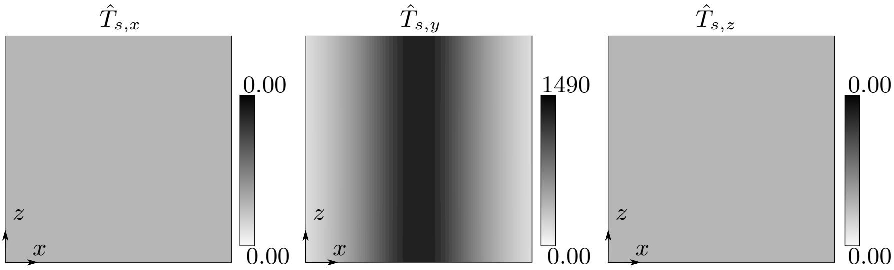


Figure 5.15: Traction field $\hat{\mathbf{T}}_s$ at $t=0.2$ of benchmark 3

5.8.11 Statement and Results for the used FSI Environment

Summarizing this chapter, the elaborated benchmarks complement the previous benchmarks of CSD, CFD, and the interface coupling procedures to a rigorous benchmark sequence for the complete partitioned FSI environment. The challenging topic of non-matching grids in partitioned FSI analyses has extensively been assessed.

It can be stated that all solution fields of the benchmarks are converging. The partitioned FSI environment for incompressible flows is assessed to work - at least in large parts - as intended. The limitation of the qualitiy of the produced results and

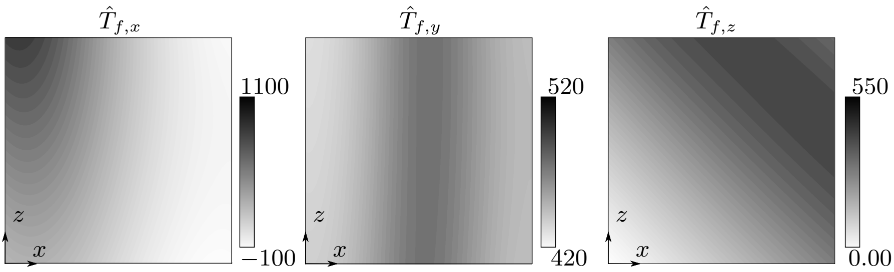


Figure 5.16: Traction field $\hat{\mathbf{T}}_f$ at $t = 0.2$ at the structure of benchmark 3

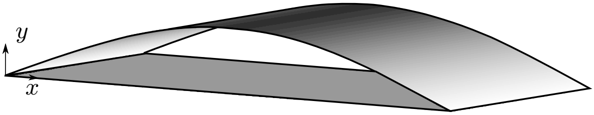


Figure 5.17: Initial and deformed shape of the structure at $t = 0.2$ of benchmark 3 - scaled with a factor of 5

the performance of the environment is set by the CFD software (cf. chapter 4). The time advancement and the traction force computation at the common interface have been mentioned in particular. The order of accuracy for a full FSI simulation therefore lowers to one. Despite the fact that the CSD solver is actually second order in space and time, the overall order of accuracy is oriented to the weakest member of the simulation. This (formal and observed) first order load computation of $\mathbf{T}_f^\Gamma$ is applied to the CSD with the same order of accuracy of one; therefore the total CSD is restricted to the performance of its load representation. Consequently the accuracies of the CSD and therefore the total FSI simulation results lower to an observed order of $p = 1$ as well.

The identification and localization of deficiencies define a task or a challenge for the code developer on the way to improve the software environment. At this point it is mentioned again that the MMS is not only well suited for the confirmation of code functionalities but also very helpful for the precise identification and localization of errors or even mistakes in the software code.
Besides the loss of accuracy it can be concluded that the sampling, the mapping, and the integration at the interface work as intended, i.e. second order accurately. The CSD software itself is confirmed to work second order accurately in space and time in the FSI environment, as long as it receives from the fluid second order accurate tractions. This has been presented in the non-coupled and the one-way coupled cases of benchmark 2 (cf. sections 5.8.9.1 and 5.8.9.2 with figures 5.22-5.30)

It can be concluded that all governing equations of the assessed FSI environment are solved consistently. Therefore, the positively assessed parts of the environment can be used for further investigations in the context of V&V in order to finally reach a

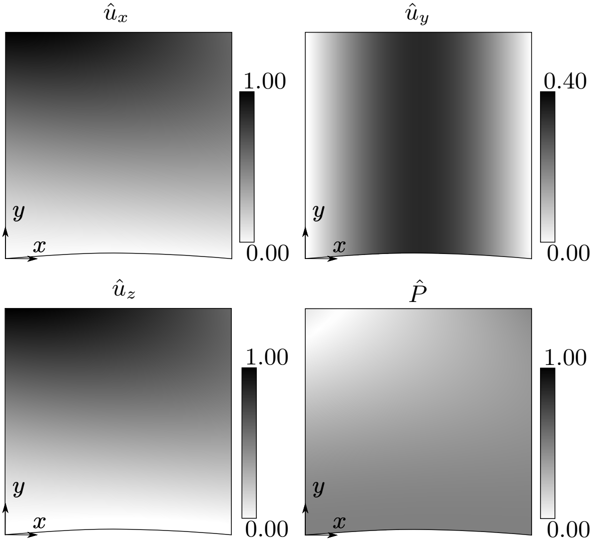


Figure 5.18: Fluid velocities $\hat{\mathbf{u}}$ and pressure $\hat{P}$ at $t = 0.2$ and $z = 0.5$ of benchmark 3

predictive capability of the FSI environment to complement the experimental wind tunnel (cf. [9, 57, 75, 79] and chapter 2).

5.9 Adaptation of the FSI Framework

As elaborated in this chapter, the MMS framework is applied to a particular partitioned FSI environment for the modeling and simulation of lightweight membranes interacting with an incompressible fluid flow. Beyond that, the general concept of the framework makes it very attractive for adaptations to assess other FSI environments or extensions of the assessed FSI environment.

In the next sections, a few examples of potential adaptations are outlined. These examples focus on the required adaptations within the presented MMS environment. The actual modeling and simulation functionalities regarding the solution of the considered field or the internal data handling are not part of the description, as they should be anyway available independent of an MMS assessment.

It is evident that if the environment is changed, the modified parts require separate investigation in order to derive the formal order of accuracy. This investigation is necessary prior to the execution of the individual benchmarks, in order to correctly set and adjust the refinement factors (cf. section 5.8.4).

It will be shown that only moderate changes are necessary in order to adapt the MMS framework for the assessment of slightly adapted, or even completely different, FSI software environments.

It may be concluded that the MMS framework lives up to its promised flexibility, adaptivity, and modularity.

5.9.1 Adaptation for Monolithic FSI

The definition of partitioned FSI compared to a monolithic FSI approach is presented above in section 5.1. As the definition of a partitioned approach with iterative coupling is representing the monolithic approach, the topic of the assessment of monolithic FSI environments is similar to the presented partitioned one. Assuming the same basic balance and interface equations of the monolithic approach as the presented one, the assessment framework processes stay the same. Regarding the procedure shown in figure 5.6, step number 3 containing Carat++ and OpenFOAM, is just replaced by the monolithic FSI simulation environment.
Besides the change of the actual FSI environment (e.g., the discretization and the solution method), no changes are required for the MMS framework to assess the monolithic FSI environment with the elaborated benchmark sequences presented in chapters 3, 4, and 5.

5.9.2 Adaptation of the Mapping Schemes

If the mapping procedure is changed or adapted to another surface coupling scheme, no changes at all would be necessary, as long the underlying equations of the Dirichlet-Neumann surface coupling are not changed in their differential form (cf. equations 5.4 and 5.6). As soon as the interface conditions change, the framework has to adapt the source terms according to the underlying equations.

5.9.3 Adaptation of the Time Advancement

As already explained in the previous sections of this chapter, the solution of the FSI problem should be independent of the solution strategy. Therefore, the same MMS setup has been used in the presented FSI benchmarks, independent of the choice of loose or iterative coupling schemes. This behavior also holds for the choice of a different communication pattern (e.g., a Jacobian based FSI communication instead the Gauss-Seidel communication [101]).

5.9.4 Creating Force-Free Benchmarks at the Interface

If the interface equation of surface traction forces is not accessible or should not be accessed, equation 5.6 must be fulfilled by the definition of the manufactured fields. This means that the structure and the fluid must be in equilibrium at the common interface Γ. If this requirement is fulfilled the term $\hat{\mathbf{T}}^{\Gamma}_{FSI}$ of equation 5.26 vanishes. One possible way to create benchmarks without additional interface traction forces would be the definition of the pressure with respect to the demanded structural force $\hat{\mathbf{T}}^{\Gamma}_{s}$. Using equations 5.18 and 5.7, the equilibrium forces at the interface Γ can be written as shown in equation 5.37.

$$\left(-\rho_f \hat{P}^{\gamma}\mathbf{I} + 2\nu_m \rho_f \hat{\mathbf{D}}\right) \cdot \hat{\mathbf{n}}^{\gamma}_{f} \cdot \det(\hat{\mathbf{F}}) + \hat{\mathbf{T}}^{\Gamma}_{s} = 0 \tag{5.37}$$

With the assumption of $\nu_m = 0$ at the interface Γ, equation 5.37 simplifies to equation 5.38.

$$-\rho_f \hat{P}^\gamma \hat{\mathbf{n}}_f^\gamma \cdot \det(\hat{\mathbf{F}}) + \hat{\mathbf{T}}_s^\Gamma = 0 \tag{5.38}$$

Assuming $\hat{\mathbf{T}}_s^\Gamma$ as any given function of $\hat{\mathbf{d}}$ (what is obviously the case in the balance of momentum of the structure), the required traction can be computed using equation 5.39.

$$\hat{P}^\gamma \hat{\mathbf{n}}_f^\gamma = \hat{\mathbf{T}}_s^\Gamma \frac{1}{\rho_f \cdot \det(\hat{\mathbf{F}})} \tag{5.39}$$

The kinematic pressure in equation 5.39 has to be set as the manufactured solution at the interface. In order to generate a pressure of the total field, the procedure proposed in section 5.8.10 is used. $\hat{P}^\gamma$ is the manufactured solution of the field $\hat{P}$ at the interface γ. A pressure field definition is suggested in equation 5.40 for a moving wall which is located at the initial time step at $\mathbf{X}^\Gamma = \mathbf{X}(x, y = 0, z, t = 0)$.

$$\hat{P} = \hat{P}^\gamma + \cos\left(x + y + z\right) \cdot \sin\left(\pi t\right) \cdot \left[y - \hat{d}_y\right]^2 \tag{5.40}$$

The expression $\left[y - \hat{d}_y\right]$ cancels the second term for the position of the wall at every timestep. Therefore, the pressure at the wall is equal to $\hat{P}^\gamma$ at every timestep. As the shear stresses are zero at the wall due to $\nu_m = 0$, the velocities can be chosen along benchmark 3 presented in table 5.4, without any further restriction.

Another possibility would be the adaptation of the molecular viscosity $\nu_m = \nu_m(\mathbf{x}, t)$ in terms of the required traction forces of the structure. This procedure has been developed and demonstrated in [26].

5.10 Results

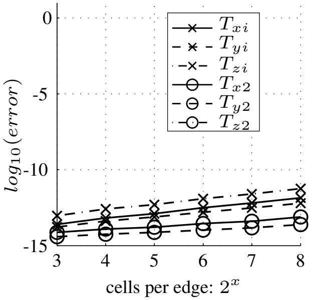


Figure 5.19: Benchmark 1: Convergence plot using matching grids

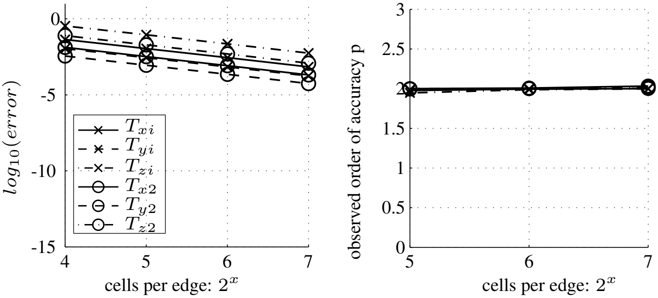


Figure 5.20: Benchmark 1: Convergence plot using one very fine grids

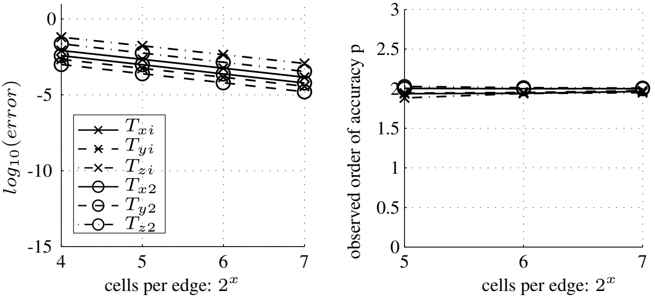


Figure 5.21: Benchmark 1: Convergence plot using a similar grid resolution

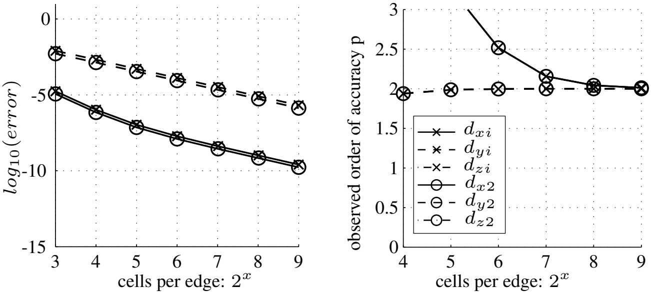


Figure 5.22: Benchmark 2 non-coupled: Convergence plot for structural displacements $\mathbf{d}$

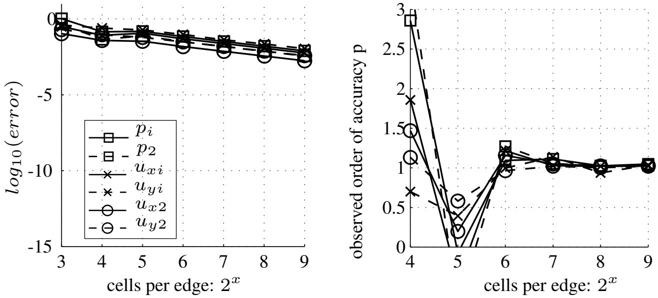


Figure 5.23: Benchmark 2 non-coupled: Convergence plot for fluid fields $\mathbf{u}$ and P

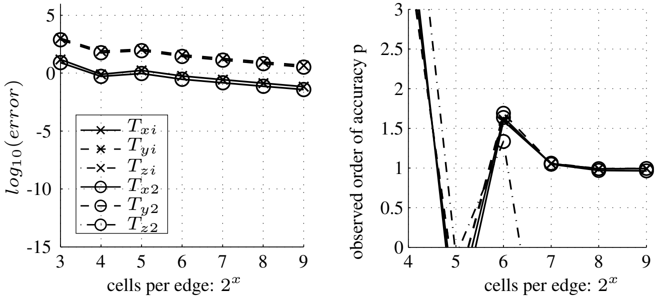


Figure 5.24: Benchmark 2 non-coupled: Convergence plot for fluid surface traction $\mathbf{T}_f^\Gamma$

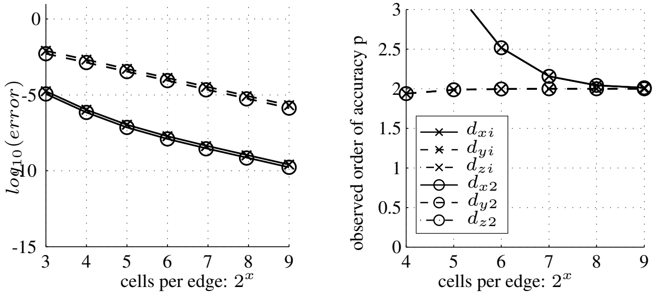


Figure 5.25: Benchmark 2 displacement coupled loose: Convergence plot for structural displacements $\mathbf{d}$

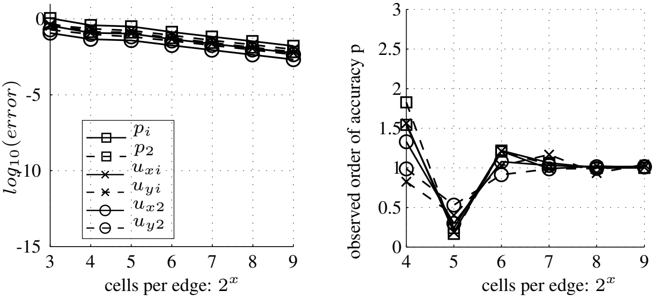


Figure 5.26: Benchmark 2 displacement coupled loose: Convergence plot for fluid fields $\mathbf{u}$ and P

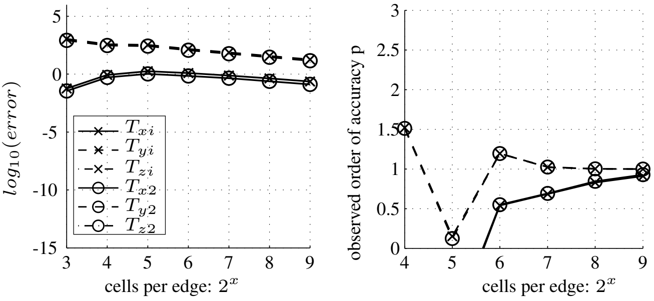


Figure 5.27: Benchmark 2 displacement coupled loose: Convergence plot for fluid surface traction $\mathbf{T}_f^\Gamma$

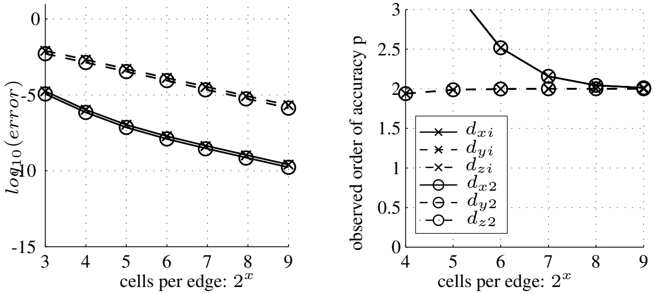


Figure 5.28: Benchmark 2 displacement coupled iterative: Convergence plot for structural displacements $\mathbf{d}$

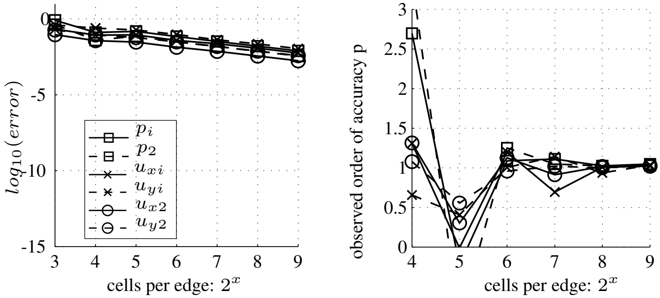


Figure 5.29: Benchmark 2 displacement coupled iterative: Convergence plot for fluid fields $\mathbf{u}$ and P

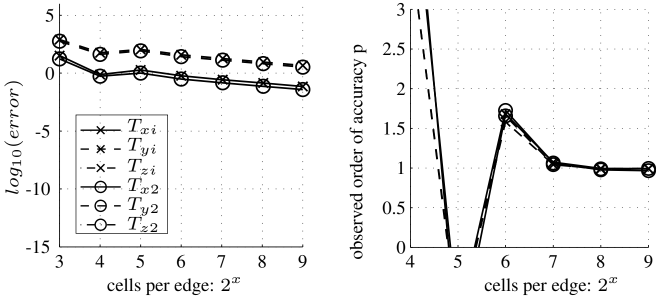


Figure 5.30: Benchmark 2 displacement coupled iterative: Convergence plot for fluid surface traction $\mathbf{T}_f^\Gamma$

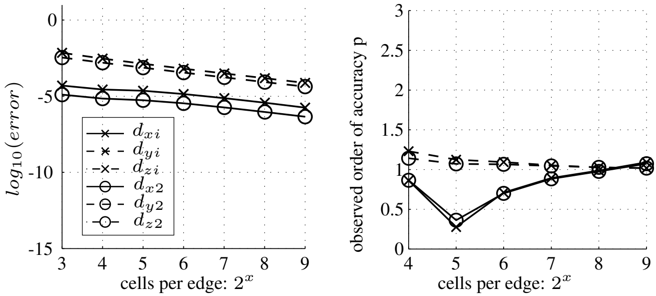


Figure 5.31: Benchmark 2 fully coupled loose: Convergence plot for structural displacements $\mathbf{d}$

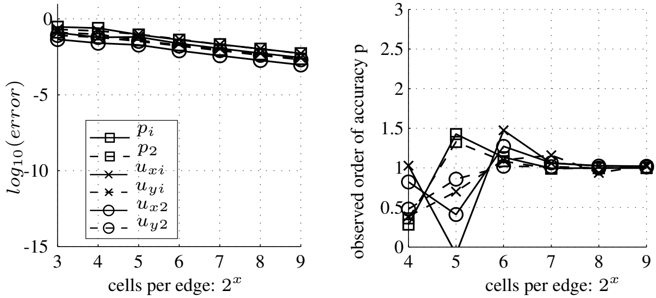


Figure 5.32: Benchmark 2 fully coupled loose: Convergence plot for fluid fields $\mathbf{u}$ and P

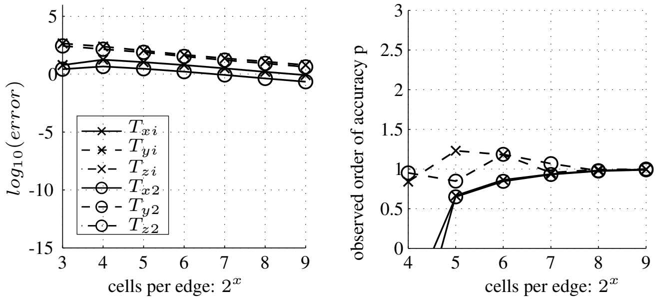


Figure 5.33: Benchmark 2 fully coupled loose: Convergence plot for fluid surface traction $\mathbf{T}_f^\Gamma$

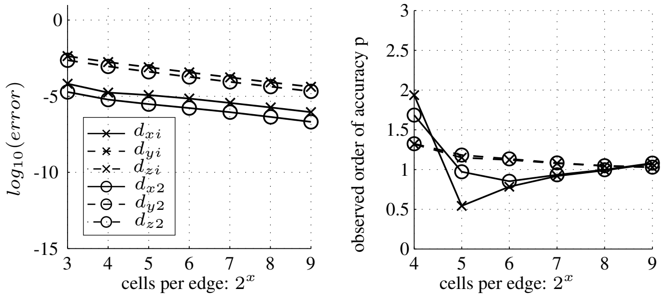


Figure 5.34: Benchmark 2 fully coupled iterative: Convergence plot for structural displacements $\mathbf{d}$

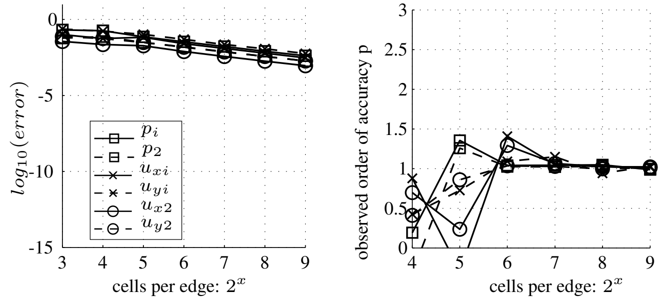


Figure 5.35: Benchmark 2 fully coupled iterative: Convergence plot for fluid fields $\mathbf{u}$ and P

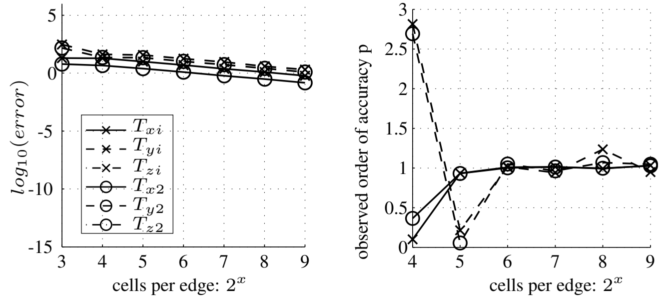


Figure 5.36: Benchmark 2 fully coupled iterative: Convergence plot for fluid surface traction $\mathbf{T}_f^{\Gamma}$

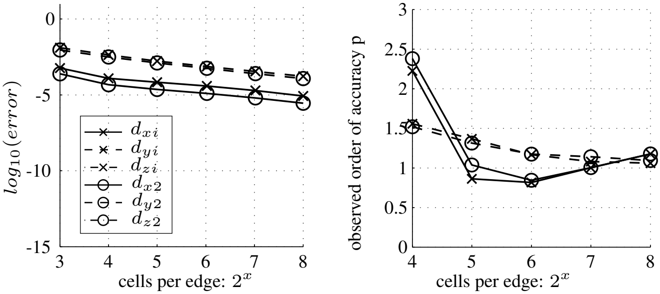


Figure 5.37: Benchmark 2 NMG1 fully coupled iterative: Convergence plot for structural displacements $\mathbf{d}$

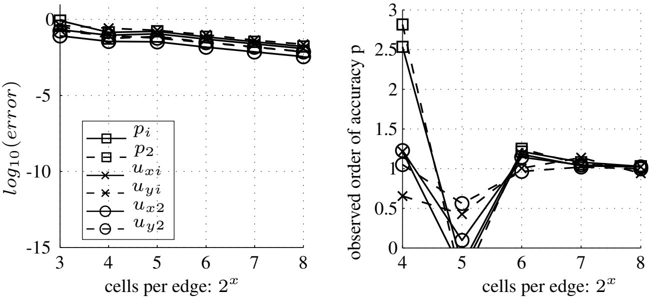


Figure 5.38: Benchmark 2 NMG1 fully coupled iterative: Convergence plot for fluid fields $\mathbf{u}$ and P

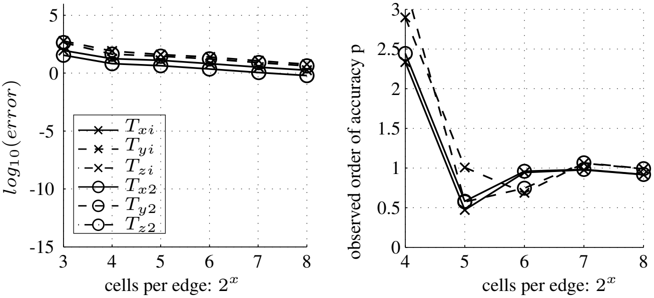


Figure 5.39: Benchmark 2 NMG1 fully coupled iterative: Convergence plot for fluid surface traction $\mathbf{T}_f^\Gamma$

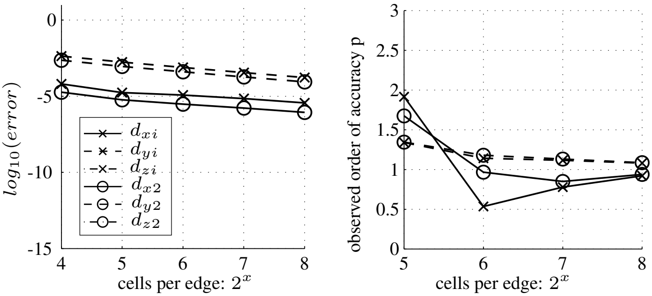


Figure 5.40: Benchmark 2 NMG2 fully coupled iterative: Convergence plot for structural displacements $\mathbf{d}$

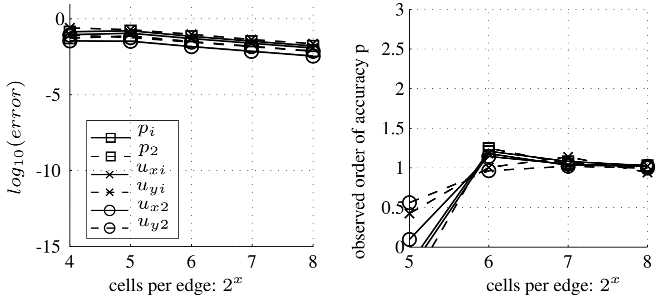


Figure 5.41: Benchmark 2 NMG2 fully coupled iterative: Convergence plot for fluid fields $\mathbf{u}$ and P

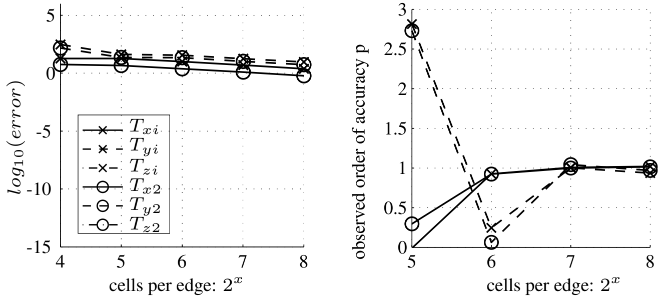


Figure 5.42: Benchmark 2 NMG2 fully coupled iterative: Convergence plot for fluid surface traction $\mathbf{T}_f^{\Gamma}$

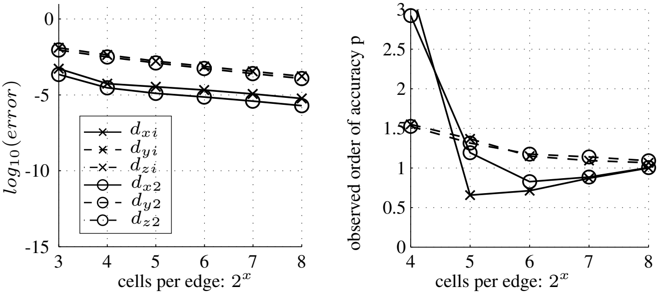


Figure 5.43: Benchmark 2 NMG3 fully coupled iterative: Convergence plot for structural displacements $\mathbf{d}$

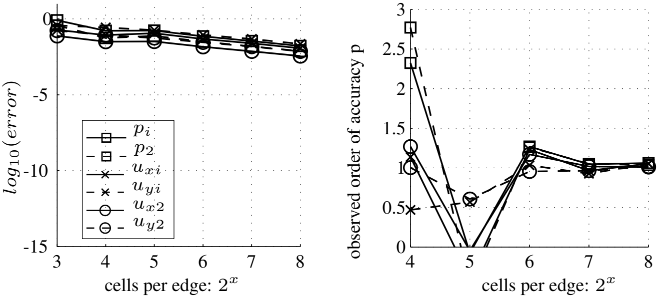


Figure 5.44: Benchmark 2 NMG3 fully coupled iterative: Convergence plot for fluid fields $\mathbf{u}$ and P

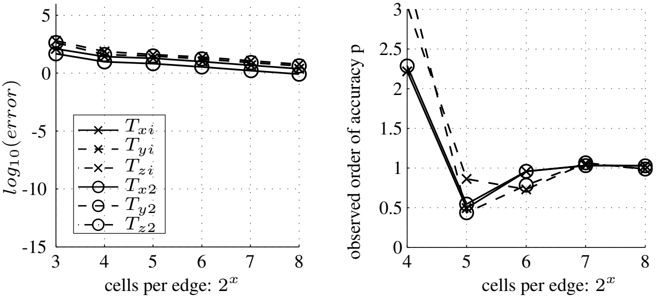


Figure 5.45: Benchmark 2 NMG3 fully coupled iterative: Convergence plot for fluid surface traction $\mathbf{T}_f^\Gamma$

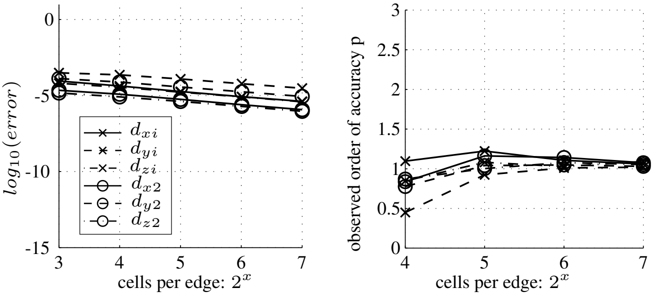


Figure 5.46: Benchmark 3 fully coupled loose 3D: Convergence plot for structural displacements $\mathbf{d}$

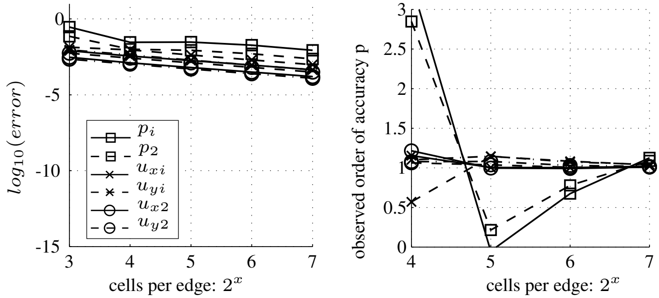


Figure 5.47: Benchmark 3 fully coupled loose 3D: Convergence plot for fluid fields $\mathbf{u}$ and P

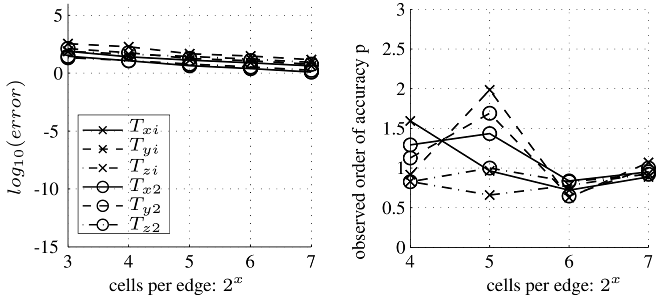


Figure 5.48: Benchmark 3 fully coupled loose 3D: Convergence plot for fluid surface traction $\mathbf{T}_f^\Gamma$

CHAPTER 6

Conclusion and Outlook

6.1 Conclusion

In this thesis, a flexible and modular framework for a stringent Code Verification was developed and applied for a partitioned Fluid-Structure Interaction (FSI) environment. The particular application was the simulation of lightweight and elastic membranes in the wind.
The main methodical focus of this thesis was on the FSI modeling in a partitioned FSI environment, in contrast to a monolithic approach. In this context the topic of non-matching grids is of interest, which has been discussed in chapter 5.
The structural dynamics, the fluid dynamics, and the coupling within the FSI are assessed in a hierarchical manner. This assessment has been performed for the single fields of the fluid and the structure individually, but also altogether for a fully coupled FSI in chapters 3, 4, and 5. The hierarchical order of the proposed benchmarks has successfully been used to precisely detect deficiencies and reveal possible improvements of the software, which influence the order of accuracy. The order of accuracy tests, concentrating on the Method of Manufactured Solutions, confirmed their character of a rigorous and strict tool for the assessment of consistency of software. Especially the independence of the discretization method and the ability to assess every software implementation based on a balance equation turned out as a major virtue of the method.

The developed framework with its benchmark applications very precisely shows the abilities and limits of the investigated software for CSD, CFD, and FSI. The assessed CSD software Carat++ confirmed its formal accuracy of second order in space and time. Therefore it can be stated that the governing equations are implemented in a consistent way. The assessed CFD software OpenFOAM® showed convergence of at least first order in space and time. Due to incomplete implementations, the formal orders of accuracy dropped to a lower level than the schemes could theoretically achieve. The assessed coupling procedures confirmed their formal accuracy of second order. The complete FSI process clearly showed an accuracy of first order in space and time. Although most of the contributing software codes per-

form second order accurate, the CFD software, as the weakest part of the simulation environment, lowered the total order of accuracy to one.
In order to increase the accuracy of the fully coupled FSI simulations with the present environment, some possible improvements have been identified and suggested in chapter 4 and 5.

6.2 Outlook

To conclude it shall once more be mentioned that the presented framework with its hierarchical benchmark suite provides a powerful method for the assessment of CFD, CSD, and FSI environments. The method can equally be used for partitioned and for monolithic simulation environments equally. The modularity of the developed method and the elaborated framework encourage for modifications of the benchmark suite and for extensions to other parts of the simulation environment like, e.g., other balance equations. The elastic structures in mind can be composed of different elements. For example, umbrella structures often consist of steel rods in combination with membranes and edge cables. This combination of elements could be tackled in a series of elementwise individual hierarchical benchmarks, and a benchmark series combining all elements in an unsteady flow regime.
The assessment of software for heat transfer, compressible fluids, contact problems, or the fluid solution using the Lattice-Boltzmann method also represent interesting future investigations regarding Code Verification.
Furthermore, one could also think of the Verification of grid or solution dependent balance equations, such as Detached Eddy Simulations (DES), the modeling of near wall behavior of turbulent flows (wall functions), or sub-grid scale models for Large-Eddy Simulations (LES).

Once the confidence in the simulation environment is set – here the presented framework can contribute its share - the next logical step should be to turn towards physically realistic problems. Simulations in the application of interest have to be performed, the numerical errors have to be evaluated, and comparisons with measurements have to be made in the context of Validation in order to finally obtain confidence in the simulation results of flexible structures in the numerical wind tunnel.

List of Figures

List of Tables

Bibliography

[1] D. Aeschliman and W. Oberkampf. “Experimental methodology for computational fluid dynamics code validation.” In: *AIAA journal* 36.5 (1998), pp. 733–741. ISSN: 0001-1452.

[2] D. Aeschliman, W. Oberkampf, and F. Blottner. “A proposed methodology for computational fluid dynamics code verification, calibration, and validation.” In: *Instrumentation in Aerospace Simulation Facilities, 1995. ICIASF’95 Record.* (2002).

[3] J. Argyris. “Die Matrizentheorie der Statik.” In: *Archive of Applied Mechanics* 25.3 (1957), pp. 174–192.

[4] Y. Başar and W. Krätzig. *Mechanik der Flächentragwerke: Theorie, Berechnungsmethoden, Anwendungsbeispiele*. Vieweg, 1985.

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

[6] T. Belytschko, B. Moran, and W. K. Liu. *Nonlinear finite element analysis for continua and structures*. Vol. 1. Wiley, 1999.

[7] R. B. Bird, W. E. Stewart, and E. N. Lightfoot. *Transport phenomena*. John Wiley & Sons, 2007.

[8] B. Blocken. “50 years of Computational Wind Engineering: Past, present and future.” In: *Journal of Wind Engineering and Industrial Aerodynamics* 129 (2014), pp. 69–102.

[9] F. G. Blottner. “Accurate Navier-Stokes results for the hypersonic flow over a spherical nosetip.” In: *Journal of spacecraft and Rockets* 27.2 (1990), pp. 113–122.

[10] L. Castro and J. Graham. “Numerical wind engineering: the way ahead?” In: *Proceedings - Institution of civil engineers structures and buildings* 134 (1999), pp. 275–278.

[11] I. Celik and O. Karatekin. “Numerical experiments on application of Richardson extrapolation with nonuniform grids.” In: *Journal of Fluids Engineering* 119 (1997), p. 584.

[12] I. Celik. “Procedure for estimation and reporting of discretization error in CFD applications.” In: *ASME Journal of Fluids Engineering* (2004).

[13] I. Celik, U. Ghia, P. Roache, C. Freitas, H. Coleman, and P. Raad. “Procedure for estimation and reporting of uncertainty due to discretization in CFD applications.” In: *ASME J. Fluids Eng* 130.7 (2008).

[14] A. J. Chorin. “Numerical solution of the Navier-Stokes equations.” In: *Mathematics of computation* 22.104 (1968), pp. 745–762.

[15] A. J. Chorin and J. E. Marsden. *A mathematical introduction to fluid mechanics*. Vol. 3. Springer, 1990.

[16] J. Chung and G. Hulbert. “A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method.” In: *Journal of applied mechanics* 60.2 (1993), pp. 371–375.

[17] L. Cochran and R. Derickson. “A physical modeler’s view of Computational Wind Engineering.” In: *Journal of Wind Engineering and Industrial Aerodynamics* 99.4 (2011), pp. 139–153.

[18] W. Dettmer and D. Perić. “A computational framework for fluid–structure interaction: finite element formulation and applications.” In: *Computer Methods in Applied Mechanics and Engineering* 195.41 (2006), pp. 5754–5779.

[19] F. H. Dieringer. “Numerical Methods for the Design and Analysis of Tensile Structures.” PhD thesis. Technische Universiät München, Germany, 2014.

[20] J. Donea, A. Huerta, J. Ponthot, and A. Rodriguez-Ferran. “Arbitrary Lagrangian–Eulerian methods.” In: *Encyclopedia of computational mechanics* (2004).

[21] R. Easterling and J. Berger. “Statistical foundations for the validation of computer models.” In: *V&V State of the Art: Proceedings of Foundations* 2 (2002).

[22] L. Eça and M. Hoekstra. “Evaluation of numerical error estimation based on grid refinement studies with the method of the manufactured solutions.” In: *Computers & Fluids* 38.8 (2009), pp. 1580–1591.

[23] L. Eça, M. Hoekstra, A. Hay, and D. Pelletier. “A manufactured solution for a two-dimensional steady wall-bounded incompressible turbulent flow.” In: *International Journal of Computational Fluid Dynamics* 21.3-4 (2007), pp. 175–188.

[24] L. Eça, M. Hoekstra, A. Hay, and D. Pelletier. “On the construction of manufactured solutions for one and two-equation eddy-viscosity models.” In: *International Journal for Numerical Methods in Fluids* 54.2 (2007), pp. 119–154.

[25] L. Eça, M. Hoekstra, A. Hay, and D. Pelletier. “Verification of RANS solvers with manufactured solutions.” In: *Engineering with computers* 23.4 (2007), pp. 253–270. ISSN: 0177-0667.

[26] S. Étienne, A. Garon, and D. Pelletier. “Some manufactured solutions for verification of fluid-structure interaction codes.” In: *Computers & Structures* 106-107 (2012), pp. 56–67.

[27] C. Farhat. “CFD-Based Nonlinear Computational Aeroelasticity.” In: *Encyclopedia of Computational Mechanics* Volume 3 Fluids, Wiley (2004).

[28] C. Farhat, P. Geuzaine, and C. Grandmont. “The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids.” In: *Journal of Computational Physics* 174.2 (2001), pp. 669–694.

[29] C. Felippa, K. Park, and M. Ross. “A classification of interface treatments for fsi.” In: *Fluid Structure Interaction II*. Springer, 2010, pp. 27–51.

[30] J. H. Ferziger and M. Perić. *Computational methods for fluid dynamics*. Vol. 3. Springer Berlin, 1996.

[31] R. Fisch, J. Franke, R. Wüchner, and K.-U. Bletzinger. “Code Verification Examples of a Fully Geometrical Nonlinear Membrane Element using the Method of Manufactured Solutions.” In: *Proceedings Structural Membranes 2013, Munich Germany* (2013).

[32] R. Fisch, J. Franke, R. Wüchner, and K.-U. Bletzinger. “Code Verification of OpenFOAM® solvers using the Method of Manufactured Solutions.” In: *7th OpenFOAM Workshop Darmstadt, Germany*. 2012.

[33] J. Franke. “A review of verification and validation in relation to CWE.” In: *Proceedings 5th International Symposium on Computational Wind Engineering (CWE2010), Chapel Hill, North Carolina, USA*. 2010.

[34] I. Fried. “Numerical integration in the finite element method.” In: *Computers & Structures* 4.5 (1974), pp. 921–932.

[35] J. Fröhlich. *Large eddy simulation turbulenter Strömungen*. Vol. 1. Springer, 2006.

[36] T. G. Gallinger. “Effiziente Algorithmen zur partitionierten Lösung stark gekoppelter Probleme der Fluid-Struktur-Wechselwirkung.” PhD thesis. Technische Universität München, Germany, 2010.

[37] A. Gasmi, M. A. Sprague, J. M. Jonkman, and W. B. Jones. *Numerical stability and accuracy of temporally coupled multi-physics modules in wind-turbine CAE tools*. National Renewable Energy Laboratory, 2013.

[38] E. W. Gekeler. *Mathematische Methoden zur Mechanik*. Springer, 2006.

[39] M. Glück, M. Breuer, F. Durst, A. Halfmann, and E. Rank. “Numerische Simulation von Fluid-Struktur-Interaktionen mit einem partitionierten Ansatz: Methodik und Anwendungen aus dem Bauwesen.” In: *VDI BERICHTE* 1682 (2002), pp. 31–50.

[40] M. Glück. *Ein Beitrag zur numerischen Simulation von Fluid-Struktur- Interaktionen: Grundlagenuntersuchungen und Anwendung auf Membrantragwerke*. Ed. by G. Universität Erlangen-Nürnberg. Shaker, 2002.

[41] P. Gousseau, B. Blocken, and G. van Heijst. “Quality assessment of Large-Eddy Simulation of wind flow around a high-rise building: validation and solution verification.” In: *Computers & Fluids* (2013).

[42] L. Hatton. “The T experiments: errors in scientific software.” In: *Computing in Science and Engineering* 4.2 (1997), pp. 27–38.

[43] E. Haug and G. H. Powell. *Finite element analysis of nonlinear membrane structures*. University of Calif., College of Eng., Department of Civil Eng., Structural Eng. and Structural Mechanics, 1972.

[44] C. Hirsch. *Numerical computation of internal and external flows: the fundamentals of computational fluid dynamics*. Vol. 1. Butterworth-Heinemann, 2007.

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

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

[47] R. Issa. "Solution of the implicitly discretised fluid flow equations by operator-splitting." In: *Journal of Computational Physics* 62.1 (1986), pp. 40–65. ISSN: 0021-9991.

[48] R. Issa, A. Gosman, and A. Watkins. "The computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme." In: *Journal of Computational Physics* 62.1 (1986), pp. 66–82.

[49] H. Jasak. "Error analysis and estimation for the finite volume method with applications to fluid flows." PhD thesis. Imperial College London, England, 1996.

[50] H. Jasak and Z. Tukovic. "Automatic mesh motion for the unstructured finite volume method." In: *Transactions of FAMENA* 30.2 (2006), pp. 1–20.

[51] X. Jiang and S. Mahadevan. "Bayesian risk-based decision method for model validation under uncertainty." In: *Reliability Engineering & System Safety* 92.6 (2007), pp. 707–718. ISSN: 0951-8320.

[52] R. Johnson, R. Schultz, I. Celik, Y. Hassan, P. Roache, and W. Pointer. "Processes and Procedures for Application of CFD to Nuclear Reactor Safety Analysis." In: *Idaho National Laboratory* (2006).

[53] F. Juretic. "Error analysis in finite volume CFD." PhD thesis. Imperial College London, England, 2005.

[54] V. Kalro and T. E. Tezduyar. "A parallel 3D computational method for fluid–structure interactions in parachute systems." In: *Computer Methods in Applied Mechanics and Engineering* 190.3 (2000), pp. 321–332.

[55] V. Kazemi-Kamyab, A. Zuijlen, and H. Bijl. "Accuracy and stability analysis of a second-order time-accurate loosely coupled partitioned algorithm for transient conjugate heat transfer problems." In: *International Journal for Numerical Methods in Fluids* 74.2 (2014), pp. 113–133.

[56] K. Klöppel and H. Friedmann. "Die Statik im Zeichen der Anpassung an elektronische Rechenautomaten." In: *VDI-Zeitschrift* 107-3 (1965), pp. 1603–1607.

[57] P. Knupp and K. Salari. *Verification of computer codes in computational science and engineering*. CRC Press, 2003. ISBN: 1584882646.

[58] M. Kojić and K.-J. Bathe. "Studies of finite element procedures—Stress solution of a closed elastic strain path with stretching and shearing using the updated Lagrangian Jaumann formulation." In: *Computers & Structures* 26.1 (1987), pp. 175–179.

[59] V. I. Krylov. *Approximate calculation of integrals*. Courier Dover Publications, 1962.

[60] D. Kuhl. “Stabile Zeitintegrationsalgorithmen in der nichtlinearen Elastodynamik dünnwandiger Tragwerke.” PhD thesis. Institut für Baustatik, Universität Stuttgart, Germany, 1996.

[61] K.-E. Kurrer. *Geschichte der Baustatik*. John Wiley and Sons, New York, 2002.

[62] U. Küttler. “Effiziente Lösungsverfahren für Fluid-Struktur-Interaktions-Probleme.” PhD thesis. Technische Universität München, Germany, 2009.

[63] U. Küttler, C. Förster, and W. A. Wall. “A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure Dirichlet fluid domains.” In: *Computational Mechanics* 38.4-5 (2006), pp. 417–429.

[64] U. Küttler and W. A. Wall. “Fixed-point fluid–structure interaction solvers with dynamic relaxation.” In: *Computational Mechanics* 43.1 (2008), pp. 61–72.

[65] B. E. Launder and D. Spalding. “The numerical computation of turbulent flows.” In: *Computer methods in applied mechanics and engineering* 3.2 (1974), pp. 269–289.

[66] J. Lee. “Certainty in Stockpile Computing: Recommending a Verification and Validation Program for Scientific Software.” In: *Sandia National Laboratories, Albuquerque, NM, and Livermore, CA* (1998).

[67] R. J. LeVeque. *Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems*. Vol. 98. SIAM, 2007.

[68] J. Linhard. *Numerisch-mechanische Betrachtung des Entwurfsprozesses von Membrantragwerken*. Ed. by G. Technische Universität München. Shaker, 2009.

[69] F. R. Menter. “Improved two-equation k-omega turbulence models for aerodynamic flows.” In: *NASA STI/Recon Technical Report N* 93 (1992), p. 22809.

[70] C. Michler. “Efficient numerical methods for fluid-structure interaction.” PhD thesis. Technical University Delft, Netherlands, 2005.

[71] P. Middendorf. “Viskoelastisches Verhalten von Polymersystemen.” PhD thesis. Universität der Bundeswehr München, Germany, 2002.

[72] N. M. Newmark. “A method of computation for structural dynamics.” In: *Proc. ASCE*. Vol. 85. 1959, pp. 67–94.

[73] W. L. Oberkampf and C. J. Roy. *Verification and Validation in Scientific Computing*. Cambridge University Press, 2010. ISBN: 0521113601.

[74] W. Oberkampf and M. Barone. “Measures of agreement between computation and experiment: validation metrics.” In: *Journal of Computational Physics* 217.1 (2006), pp. 5–36. ISSN: 0021-9991.

[75] W. Oberkampf and F. Blottner. “Issues in computational fluid dynamics code verification and validation.” In: *AIAA journal* 36.5 (1998), pp. 687–695. ISSN: 0001-1452.

[76] W. Oberkampf, M. Sindir, and A. Conlisk. “Guide for the verification and validation of computational fluid dynamics simulations.” In: *Am. Institute of Aeronautics and Astronautics* (1998).

[77] W. Oberkampf and T. Trucano. “Verification and validation benchmarks.” In: *Nuclear Engineering and Design* 238.3 (2008), pp. 716–743. ISSN: 0029-5493.

[78] W. Oberkampf and T. Trucano. “Verification and validation in computational fluid dynamics.” In: *Progress in Aerospace Sciences* 38.3 (2002), pp. 209–272. ISSN: 0376-0421.

[79] W. Oberkampf, T. Trucano, and C. Hirsch. *Verification, validation, and predictive capability in computational engineering and physics*. 2004, p. 345.

[80] R. Ohayon and C. Felippa. *Advances in computational methods for fluid-structure interaction and coupled problems*. Vol. 124. Elsevier, 2001.

[81] C. H. Page and P. Vigoureux. *The international system of units (SI)*. Vol. 330. US Dept. of Commerce, National Bureau of Standards: for sale by the Supt. of Docs., US Govt. Print. Off., 1974.

[82] S. Patankar. *Numerical heat transfer and fluid flow*. CRC Press, 1980.

[83] S. V. Patankar and D. B. Spalding. “A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows.” In: *International Journal of Heat and Mass Transfer* 15.10 (1972), pp. 1787–1806.

[84] M. Pilch. *Guidelines for Sandia ASCI Verification and Validation Plans-Content and Format: Version 2.0*. Sandia National Laboratories, 2001.

[85] S. Pope. *Turbulent flows*. Cambridge Univ Press, 2000.

[86] L. Prandtl. “Bericht über Untersuchungen zur ausgebildeten Turbulenz.” In: *Z. Angew. Math. Mech* 5.2 (1925), pp. 136–139.

[87] M. A. Puso and T. A. Laursen. “A mortar segment-to-segment contact method for large deformation solid mechanics.” In: *Computer Methods in Applied Mechanics and Engineering* 193.6 (2004), pp. 601–629.

[88] P. Richards and R. Hoxey. “Appropriate boundary conditions for computational wind engineering models using the k-turbulence model.” In: *Journal of wind engineering and Industrial aerodynamics* 46 (1993), pp. 145–153. ISSN: 0167-6105.

[89] S. Richards. “Completed Richardson extrapolation in space and time.” In: *Communications in numerical methods in engineering* 13.7 (1997), pp. 573–582.

[90] A. Rizzi and J. Vos. “Toward establishing credibility in computational fluid dynamics simulations: Special section: Credible computational fluid dynamics simulations.” In: *AIAA journal* 36.5 (1998), pp. 668–675.

[91] P. J. Roache. *Verification and validation in computational science and engineering*. Hermosa Publishers, 1998, pp. 8–9.

[92] P. J. Roache. “Perspective: a method for uniform reporting of grid refinement studies.” In: *Journal of Fluids Engineering* 116.3 (1994), pp. 405–413.

[93] C. J. Roy. “Review of code and solution verification procedures for computational simulation.” In: *Journal of Computational Physics* 205.1 (2004), pp. 131–156.

[94] C. J. Roy and W. L. Oberkampf. “A complete framework for verification, validation, and uncertainty quantification in scientific computing.” In: *48th AIAA Aerospace Sciences Meeting and Exhibit*. Vol. 124. 2010.

[95] C. J. Roy and W. L. Oberkampf. “A comprehensive framework for verification, validation, and uncertainty quantification in scientific computing.” In: *Computer Methods in Applied Mechanics and Engineering* 200.25 (2011), pp. 2131–2144.

[96] C. Roy. “Review of Discretization Error Estimators in Scientific Computing.” In: *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition* 48 (2010).

[97] C. Roy, C. Nelson, T. Smith, and C. Ober. *Verification of Euler/Navier–Stokes codes using the method of manufactured solutions*. Vol. 44. Wiley Online Library, 2004, pp. 599–620.

[98] K. Salari and P. Knupp. “Code verification by the method of manufactured solutions.” In: *Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US)* (2000).

[99] M. Schäfer. “Coupled Fluid-Solid Problems: Survey on Numerical Approaches and Applications.” In: *ASME 2003 Pressure Vessels and Piping Conference*. American Society of Mechanical Engineers. 2003, pp. 3–14.

[100] S. Schlesinger, R. Crosbie, R. Gagné, G. Innis, C. Lalwani, J. Loch, R. Sylvester, R. Wright, N. Kheir, and D. Bartos. “Terminology for model credibility.” In: *Simulation* 32.3 (1979), pp. 103–104.

[101] S. Sicklinger, V. Belsky, B. Engelmann, H. Elmqvist, H. Olsson, R. Wüchner, and K.-U. Bletzinger. “Interface Jacobian-based Co-Simulation.” In: *International Journal for Numerical Methods in Engineering* 98.6 (2014), pp. 418–444. ISSN: 1097-0207.

[102] J. Smagorinsky. “General circulation experiments with the primitive equations: I. The basic experiment*.” In: *Monthly weather review* 91.3 (1963), pp. 99–164.

[103] T. Stathopoulos. “The numerical wind tunnel for industrial aerodynamics: Real or virtual in the new millennium?” In: *Wind and Structures* 5.2 (2002), pp. 193–208.

[104] S. Steinberg and P. J. Roache. “Symbolic manipulation and computational fluid dynamics.” In: *Journal of Computational Physics* 57.2 (1985), pp. 251–284.

[105] F. Stern, R. Wilson, and J. Shao. “Quantitative V&V of CFD simulations and certification of CFD codes.” In: *International Journal for Numerical Methods in Fluids* 50.11 (2006), pp. 1335–1355. ISSN: 1097-0363.

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

[107] A. H. Stroud and D. Secrest. *Gaussian quadrature formulas*. Vol. 374. Prentice-Hall Englewood Cliffs, NJ, 1966.

[108] E. Sudicky and E. Frind. “Contaminant transport in fractured porous media: Analytical solutions for a system of parallel fractures.” In: *Water Resources Research* 18.6 (1982), pp. 1634–1642.

[109] B. Szabó and I. Babuska. *Introduction to finite element analysis*. 1989.

[110] R. Temam. “Une méthode d’approximation de la solution des équations de Navier-Stokes.” In: *Bulletin de la Société Mathématique de France* 96 (1968), pp. 115–152.

[111] P. Thomas and C. Lombard. “Geometric conservation law and its application to flow computations on moving grids.” In: *AIAA journal* 17.10 (1979), pp. 1030–1037.

[112] T. Trucano, R. Easterling, K. Dowding, T. Paez, A. Urbina, V. Romero, B. Rutherford, and R. HILLS. “Description of the Sandia validation metrics project.” In: *Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US)* (2001).

[113] T. Trucano, M. Pilch, and W. Oberkampf. “General concepts for experimental validation of ASCI code applications.” In: *Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US)* (2002).

[114] T. Trucano, L. Swiler, T. Igusa, W. Oberkampf, and M. Pilch. “Calibration, validation, and sensitivity analysis: What’s what.” In: *Reliability Engineering & System Safety* 91.10-11 (2006), pp. 1331–1357. ISSN: 0951-8320.

[115] T. Truncano and J. Moya. “Guidelines for Sandia ASCI Verification and Validation Plans-Content and Format: Version 1.0.” In: *Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US)* (1999).

[116] M. Turner, R. Clough, H. Martin, and L. Topp. “Stiffness and deflection analysis of complex structures.” In: *Journal of the Aeronautical Sciences (Institute of the Aeronautical Sciences)* 23-9.9 (1956), pp. 805–823.

[117] A. Valli, A. Quarteroni, et al. *Domain decomposition methods for partial differential equations*. Oxford University Press, 1999.

[118] A. Van Dam, S. K. Feiner, M. McGuire, and D. F. Sklar. *Computer graphics: principles and practice*. Pearson Education, 2013.

[119] J. Vedovoto, A. Silveira Neto, A. Mura, and L. Figueira da Silva. “Application of the method of manufactured solutions to the verification of a pressure-based finite-volume numerical scheme.” In: *Computers & Fluids* (2011).

[120] H. K. Versteeg and W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.

[121] D. Wilcox. *Turbulence modeling for CFD*. DCW industries La Canada, CA, 2006.

[122] P. Wriggers. *Nichtlineare finite-element-methoden*. Springer Verlag, 2001.

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

[124] R. Wüchner. *Mechanik und Numerik der Formfindung und Fluid-Struktur-Interaktion von Membrantragwerken*. Ed. by G. Technische Universität München. Shaker, 2007.

[125] R. Wüchner, A. Kupzok, and K.-U. Bletzinger. “A framework for stabilized partitioned analysis of thin membrane–wind interaction.” In: *International journal for numerical methods in fluids* 54.6-8 (2007), pp. 945–963.

[126] W. Wunderlich. "Computer und Ingenieurwesen - ein Rückblick." In: *Festkolloquium zur Verleihung der Ehrendoktorwürde von Prof. Wunderlich an der Ruhr-Universität Bochum* (2005).

[127] O. C. Zienkiewicz and R. L. Taylor. *The Finite Element Method*. Vol. 1. McGraw-Hill London, 2000.

Bisherige Titel der Schriftenreihe

Band	Titel
1	Frank Koschnick, *Geometrische Lockingeffekte bei Finiten Elementen und ein allgemeines Konzept zu ihrer Vermeidung*, 2004.
2	Natalia Camprubi, *Design and Analysis in Shape Optimization of Shells*, 2004.
3	Bernhard Thomee, *Physikalisch nichtlineare Berechnung von Stahlfaserbetonkonstruktionen*, 2005.
4	Fernaß Daoud, *Formoptimierung von Freiformschalen - Mathematische Algorithmen und Filtertechniken*, 2005.
5	Manfred Bischoff, *Models and Finite Elements for Thin-walled Structures*, 2005.
6	Alexander Hörmann, *Ermittlung optimierter Stabwerkmodelle auf Basis des Kraftflusses als Anwendung plattformunabhängiger Prozesskopplung*, 2006.
7	Roland Wüchner, *Mechanik und Numerik der Formfindung und Fluid-Struktur-Interaktion von Membrantragwerken*, 2006.
8	Florian Jurecka, *Robust Design Optimization Based on Metamodeling Techniques*, 2007.
9	Johannes Linhard, *Numerisch-mechanische Betrachtung des Entwurfsprozesses von Membrantragwerken*, 2009.
10	Alexander Kupzok, *Modeling the Interaction of Wind and Membrane Structures by Numerical Simulation*, 2009.
11	Bin Yang, *Modified Particle Swarm Optimizers and their Application to Robust Design and Structural Optimization*, 2009.
12	Michael Fleischer, *Absicherung der virtuellen Prozesskette für Folgeoperationen in der Umformtechnik*, 2009.
13	Amphon Jrusjrungkiat, *Nonlinear Analysis of Pneumatic Membranes - From Subgrid to Interface*, 2009.

Band	**Titel**
14	Alexander Michalski, *Simulation leichter Flächentragwerke in einer numerisch generierten atmosphärischen Grenzschicht*, 2010.
15	Matthias Firl, *Optimal Shape Design of Shell Structures*, 2010.
16	Thomas Gallinger, *Effiziente Algorithmen zur partitionierten Lösung stark gekoppelter Probleme der Fluid-Struktur-Wechselwirkung*, 2011.
17	Josef Kiendl, *Isogeometric Analysis and Shape Optimal Design of Shell Structures*, 2011.
18	Joseph Jordan, *Effiziente Simulation großer Mauerwerksstrukturen mit diskreten Rissmodellen*, 2011.
19	Albrecht von Boetticher, *Flexible Hangmurenbarrieren: Eine numerische Modellierung des Tragwerks, der Hangmure und der Fluid-Struktur-Interaktion*, 2012.
20	Robert Schmidt, *Trimming, Mapping, and Optimization in Isogeometric Analysis of Shell Structures*, 2013.
21	Michael Fischer, *Finite Element Based Simulation, Design and Control of Piezoelectric and Lightweight Smart Structures*, 2013.
22	Falko Hartmut Dieringer, *Numerical Methods for the Design and Analysis for Tensile Structures*, 2014.
23	Rupert Johannes Fisch, *Code Verification of Partitioned FSI Environments for Lightweight Structures*, 2014.