

Contents

1	Verification and Validation	3
1.1	Verification of Code	3
1.1.1	Method of manufactured solution	4
1.1.2	Verification of the fluid-structure interaction solver by MMS	5
1.2	Validation	6

Chapter 1

Verification and Validation

Computer simulations are in many engineering applications a cost-efficient way for conducting design and performance optimization of physical problems. However, thrusting blindly numbers generated from a computer code can prove to be naive. It doesn't take a lot of coding experience before one realizes the many things that can brake down and produce unwanted or unexpected results. Therefore, *credability* of computational results are essential, meaning the simulation is worthy of belief or confidence [?]. *Verification and validation* (V&V) is the main approach for assessing and the reliability of computational simulations [?]. A thorough discussion of (V&V) concepts and terminology during the last century can be found in [?]. In this thesis, the definitions provided by the *American Society of Mechanical Engineers guide for Verification and Validation in Computational Solid Mechanics* [?] are followed.

Definition 1.1. Verification: The process of determining that a computational model accurately represents the underlying mathematical model and its solution.

Definition 1.2. Validation: The process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model.

Simplified *verification* considers if one solves the equations right, while *validation* is checking if one solves the right equations for the given problem [?].

To test a computational code for all possible parameters, conditions and applications are simply too time consuming. Verification and validation are therefore ongoing processes, with no clear boundary of completeness unless additional requirements are specified [?]. The goal of this chapter is to verify our implementations using the method of manufactured solution (MMS), addressing validation in a later chapter.

1.1 Verification of Code

Within scientific computing a mathematical model is often the baseline for simulations of a particular problem of interest. For scientists exploring physical phenomena, the mathematical model is often on the form of systems of partial differential equations (PDE's). A computer program therefore must evaluate mathematical identities such a differential operators and functions in order to produce accurate

solutions of the governing PDE's. Through verification of code, the ultimate goal is to ensure a computer program truly represents the mathematical model. To accumulate sufficient evidence that a mathematical model is solved correctly by a computer code, it must excel within predefined criteria. If the acceptance criterion is not satisfied, a coding mistake is suspected. Should the code pass the preset criteria, the code is considered verified. Of the different classes of test found in [?], *Order-of-accuracy* (OAA) is regarded as the most rigorous acceptance criterion for verification [?], [?], [?]. In addition to error estimation and convergence of the numerical solution, the method ensure the discretization error E is reduced in accordance with the *formal order of accuracy* expected from the numerical scheme. The formal order of accuracy is defined to be the theoretical rate at which the truncation error of a numerical scheme is expected to reduce. The *observed order of accuracy* is the actual rate produced by our code. The order of convergence is calculated assuming a PDE of space and time, order-of-accuracy tests are conducted separately of

By monitoring the discretization error E by spatial and temporal refinements, one assumes the asymptotic behavior,

$$E = E_x + E_t = u_e - u_h = C\Delta t^p + D\Delta x^l$$

where C is a constant, h represents the spatial or temporal resolution, and p is the convergence rate of the numerical scheme. For order of convergence tests, the code is assumed to be verified if the discretization error is proportional to h^p .

1.1.1 Method of manufactured solution

The basis of a convergence test is how to find an exact/reference solution, in order to compute the discretization error E . However solutions of PDE's are limited, and often simplifications of the original problem are needed to produce analytically solutions. *The method of manufactured solutions* provides a simple yet robust way of making analytic solutions for PDE's. Let a partial differential equation of interest be on the form

$$\mathbf{L}(\mathbf{u}) = \mathbf{f}$$

Here \mathbf{L} is a differential operator, \mathbf{u} is variable the of interest, and \mathbf{f} is some source term. In the method of manufactured solution one first manufactures a solution \mathbf{u} for the given problem. In general, the choice of \mathbf{u} will not satisfy the governing equations, producing a source term \mathbf{f} after differentiation by \mathbf{L} . The produced source term will cancel any imbalance formed by the manufactured solution \mathbf{u} of the original problem. Therefore, the manufactured solution can be constructed without any physical reasoning, proving code verification as a purely a mathematical exercise were our only interest is to verify the solution [?]. If the MMS is not chosen properly the test will not work, therefore some guidelines for rigorous verification have been proposed in [?, ?, ?].

- The manufactured solution (MS), should be composed of smooth analytic functions such as exponential, trigonometric, or polynomials.
- The MS should have sufficient number of derivatives, exercising all terms and derivatives of the PDE's.

To deeply verify the robustness of the method of manufactured solution, a report regarding code verification by MMS for CFD was published by Salari and Knupp [?]. This thorough work applied the method for both compressible and incompressible time-dependent Navier-Stokes equation. To prove its robustness the authors deliberately implemented code errors in a verified Navier-Stokes solver by MMS presented in the report. In total 21 blind testcases were implemented, where different approaches of verification frameworks were tested. Of these, 10 coding mistakes that reduces the observed order-of-accuracy was implemented. Here the method of manufactured solution captured all of them.

For construction of the sourceterm \mathbf{f} the Unified Form Language (UFL) [?] provided in FEniCS Project will be used. UFL provides a simple yet powerfull method of declaration for finite element forms. An example will be provided in the Fluid Problem section.

1.1.2 Verification of the fluid-structure interaction solver by MMS

In the last section, the choice MMS for verification of code is stressed to not be limited by the overall physics of the given problem of interest, which is true considering the fluid and solid problems separate. The freedom of choosing a MMS for a monolithic fluid-structure interaction problem is however limited to some extent, due coupling coditions at the interface. Recall from chapter 4,

1. Kinematic boundary condition $\hat{\mathbf{u}}_s = \hat{\mathbf{u}}_f$, enforced strongly by a continious velocity field in the fluid and solid domain.
2. Dynamic boundary condition $\sigma_s \cdot \mathbf{n} = \sigma_f \cdot \mathbf{n}$, enforced weakly by omitting the boundary integrals from the weak formulation in problem ?.

The construction of a MMS is therefore not obvious, as it must fulfill condition 1 and 2, in addition to the divergence-free condition in the fluid. The struggle is reflected of the absence of research, regarding MMS for coupled FSI solvers in the litterature. The challenge are often disregarded, such as cite METHOLOGY for COMPARING.., presenting a thorough methodology for comparing fluid-structure interaction solvers based on the verification and validation framework. The verification process is conducted on the fluid and structure solver separatly, taking no consideration to the coupling. Instead, the correctness of the coupling is evaluated by the code validation.

The approach clearly ease the process, assuming verification of each codeblock is "sufficient" to declare the code verified. It must be stressed that solving each problem individually is not true verification, in reference to a monolithic approach

where the problems are solved at the same time.

The construction of a MMS for a monolithic FSI problem is therefore out of the scope of this thesis. Conducting verification on the fluid and structure separately is not, but considered "good enough" to show the mathematical model is discretized accurately.

1.2 Validation

Through *verification*, one can assure that a scientific code implements a mathematical model correctly. However, correctness is unnecessary, if the model fails to serve as an accurate representation of the physical problem of interest. By definition 1.2, *Validation* is the act of demonstrating that a mathematical model is applicable for its intended use with a certain degree of accuracy. This demonstration is not intended to portray the model as an absolute truth, nor the best model available [?]. The acceptance criteria of validation is based on the numerical solution produced, by comparison with existing experiment data. The dependency of thrusting experiments, makes *validation* assess a wide range of issues [?]

- The relevance of mathematical model compared to the experiment.
- Assurance of that the experiments was conducted correctly, in accordance with prescribed parameters, initial and boundary conditions e.t
- Uncertainty of experimental measurements

Comparing numerical results with existing experiments, raise some issues in the validation process. First, reproducing of experimental results...

Bibliography

- [1] Robert T Biedron and Elizabeth M Lee-Rausch. Rotor Airloads Prediction Using Unstructured Meshes and Loose CFD/CSD Coupling.
- [2] P I Crumpton. Implicit time accurate solutions on unstructured dynamic grids. (95):1–23, 1995.
- [3] J Donea, A Huerta, J-Ph Ponthot, and A Rodríguez-Ferran. Arbitrary Lagrangian-Eulerian methods. (1969):1–38, 2004.
- [4] Richard P Dwight. Robust Mesh Deformation using the Linear Elasticity Equations.
- [5] Miguel A Fernández and Jean-Frédéric Gerbeau. Algorithms for fluid-structure interaction problems. 2009.
- [6] Miguel A. Fernández, Jean Frederic Gerbeau, and Ceremade Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.
- [7] Christiane Förster, Wolfgang A. Wall, and Ekkehard Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1278–1293, 2007.
- [8] Bernhard Gatzhammer. Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions. page 261, 2014.
- [9] Brian T. Helenbrook. Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*, 2003.
- [10] Su-Yuen Hsu, Chau-Lyan Chang, and Jamshid Samareh. A Simplified Mesh Deformation Method Using Commercial Structural Analysis Software.
- [11] Hrvoje Jasak and Željko Tuković. Automatic mesh motion for the unstructured Finite Volume Method. *Transactions of Famena*, 30(2):1–20, 2006.
- [12] V V Meleshko. Bending of an Elastic Rectangular Clamped Plate: Exact Versus 'Engineering' Solutions. *Journal of Elasticity*, 48(1):1–50, 1997.

- [13] Selim MM and Koomullil RP. Mesh Deformation Approaches – A Survey. *Journal of Physical Mathematics*, 7(2), 2016.
- [14] J Newman. *Marine Hydrodynamics*. 1977.
- [15] T. Richter and T. Wick. Finite elements for fluid-structure interaction in ALE and fully Eulerian coordinates. *Computer Methods in Applied Mechanics and Engineering*, 199(41-44):2633–2642, 2010.
- [16] Thomas Richter. Fluid Structure Interactions. 2016.
- [17] K Stein, T Tezduyar, and R Benney. Mesh Moving Techniques for Fluid-Structure Interactions With Large Displacements.
- [18] Wolfgang A. Wall, Axel , Gerstenberger, Peter , Gamnitzer, Christiane , Förster, and Ekkehard , Ramm. Large Deformation Fluid-Structure Interaction – Advances in ALE Methods and New Fixed Grid Approaches. In *Fluid-Structure Interaction: Modelling, Simulation, Optimisation*, pages 195—232. Springer Berlin Heidelberg, 2006.
- [19] Thomas Wick. *Adaptive Finite Element Simulation of Fluid-Structure Interaction with Application to Heart-Valve*. PhD thesis, Heidelberg.
- [20] Thomas Wick. Solving Monolithic Fluid-Structure Interaction Problems in Arbitrary Lagrangian Eulerian Coordinates with the deal.II Library.
- [21] Thomas Wick. Fluid-structure interactions using different mesh motion techniques. *Computers and Structures*, 89(13-14):1456–1467, 2011.
- [22] Thomas Wick. Fully Eulerian fluid-structure interaction for time-dependent problems. *Computer Methods in Applied Mechanics and Engineering*, 255:14–26, 2013.
- [23] P. Wriggers. *Computational contact mechanics, second ed., Springer*. 2006.