Partitioned Fluid-Structure Interaction Simulation of Maritime Applications

Vom Promotionsausschuss der
Technischen Universität Hamburg

zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte Dissertation

von
Jorrid Lund, M.Sc.

aus
Kaltenkirchen

2024

Vorsitzender des Prüfungsausschusses
Prof. Dr.-Ing. Robert Seifried

Gutachter
1. Gutachter: Prof. Dr.-Ing. habil. Alexander Düster
2. Gutachter: Prof. Dr.-Ing. Moustafa Abdel-Maksoud

Tag der mündlichen Prüfung: 04. März 2024

Acknowledgements

For the professional support during my PhD, I would like to thank my supervisor Prof. Alexander Düster. His guidance and advice were essential to enable me to complete this work. I would also like to thank Prof. Moustafa Abdel-Maksoud who was the second reviewer of my work for the good cooperation during our collaborative research. For chairing the examination of my work, I extend my thanks to Prof. Robert Seifried.

Further on, I am grateful to the German Research Foundation (Deutsche Forschungsgemeinschaft) for the funding without which my work would not have been possible.

For the many fruitful discussions and support over the last few years, I would like to thank my office colleague Lars. My gratitude also goes to all my other colleagues for their support and the nice work environment in our institute.

I am also thankful to my friends and my family, especially to my parents who were always there for me and supported and encouraged me.

Most of all, I thank my wife Svenja, who believed in me and motivated me at all times.

Contents

List of Acronyms

FSI	Fluid-structure interaction
FEM	Finite element method
BEM	Boundary element method
FVM	Finite volume method
CFRP	Carbon-fiber-reinforced polymer
FE	Finite element
DOF	Degrees of freedom
ALE	Arbitrary Lagrangian Eulerian
QUICK	Quadratic upstream interpolation for convective kinematics
SIMPLE	Semi-implicit method for pressure-linked equations
PISO	Pressure implicit with splitting of operators
AABB	Axis-aligned bounding box
QNLS	Quasi-Newton least squares
MPI	Message passing interface
FWT	Floating wind turbine
HyStOH	Hydrodynamic and structural-mechanical optimization of a semi-submersible for offshore wind turbines
FSIOpt	Fluid-structure interaction and optimization of floating platforms for offshore wind turbines
BMWi	Federal Ministry for Economic Affairs and Energy of Germany (Bundesministerium für Wirtschaft und Energie)
OWT	Offshore wind turbine
JONSWAP	Joint North Sea Wave Project
TUHH	Hamburg University of Technology
IUT	Isfahan University of Technology
WEC	Wave energy converter
GFRP	Glass-fiber reinforced polymer
KCS	KRISO container ship

Abstract

In this work, the benefits of the partitioned approach in the fluid-structure interaction (FSI) simulation of maritime application cases are demonstrated in four different application cases. To begin with, the theoretical basis for the simulation of fluid and structural problems is introduced – in combination with the basics that are important for the later application cases. The applied partitioned coupling approach is explained in detail. The different extrapolation, interpolation, and convergence acceleration techniques used in this work are presented.

The first maritime application case is a floating wind turbine (FWT) that generates electricity in a deep sea offshore environment. The FWT is simulated in different conditions to examine the structural stability and power generation. Based on the presented results, improvements in the design of the FWT are proposed. The second maritime example is a wave energy converter (WEC) that is used to generate electricity from wave energy. A prediction of the motion, the power generation, and the loading on the mooring lines of the WEC based on fully coupled FSI simulations are presented.

An FSI simulation of a multilayered submersible mixer serves as a validation and verification case to prove the accuracy of the presented approach, not only for global variables like thrust and efficiency but also for local strains of the submersible mixer. The high flexibility and the multilayered structure pose special challenges in the simulation. The significant differences between the results for an uncoupled fluid simulation and a fully-coupled simulation highlight the necessity of considering FSI effects for flexible blades.

The last and final example in this work is the ship propeller. In the first part of this chapter, the KCS ship propeller design is analyzed in detail. The impact of varying the flexibility of the blade on thrust, torque, and efficiency is investigated in detail. Also, the deformation of the flexible blades is visualized and examined. Improvements in the coupling of the FSI simulation for faster computation and higher accuracy of the ship propeller simulation are proposed and applied. These improvements concern the correct transfer of the deformations to the fluid solver with a corotational approach and the convergence acceleration.

In the second part of the final chapter, an optimization of a flexible ship propeller based on an evolutionary algorithm is performed. For the optimization, an automatic framework is developed that can simulate a flexible carbon-fiber-reinforced polymer (CFRP) propeller with geometry and material modifications. The bend-twisting effect is utilized to optimize the stiffness and the fiber direction of the CFRP propeller in conjunction with the pitch. The optimization is supposed to ensure high efficiency, a constant thrust in comparison with the original propeller, and a reduction in cavitation area, cavitation variation, and thrust variation. Optimizing the last three parameters should also lead to an indirect reduction of the noise emitted by the propeller. The desired result of the optimization is a propeller with higher efficiency compared to the stiff propeller design – with only a fraction of the cavitation area and thrust and cavitation variation.

All of the examples show the importance of considering FSI effects for flexible maritime structures, proving that the partitioned approach is suitable for this task.

1 Introduction

1.1 Motivation

For many applications, fluid-structure interaction (FSI) is of major importance to capture the mutual dependency of the deformation of a flexible structure under the influence of fluid loads. If the load on a flexible structure due to a fluid flow leads to a large deformation, the deformation can have a significant effect on the fluid flow. In these cases, the mutual dependency of the fluid and the structural behavior should be considered.

The effects of FSI are especially important in the field of marine engineering since the designed structure is usually entirely enclosed by highly dynamic fluids. For floating structures like wave energy converters or floating wind turbines under unsteady loads due to waves and wind, computing the deformation of the structure without considering the dynamic FSI effects is often problematic since the large displacements and deformations of the floating structure have a significant impact on the fluid loads. An accurate estimation of the structural behavior is necessary to design the structure in a resource-efficient manner while avoiding a possible failure of the structure under the rough conditions typical for offshore engineering. However, using prototypes to estimate the structural response of offshore structures is particularly expensive due to the high costs of the construction and installation of a prototype and the huge effort of testing under offshore conditions. Therefore the necessary number of prototypes should be reduced as much as possible to operate economically.

FSI simulations enable this reduction since they allow to accurately estimate the structural response under operating conditions. With this, it is possible to achieve an efficient design that enables and improves the harvesting of the vast renewable energy resources of wind and waves that are available offshore.

The transition to renewable energies is one of the most important challenges in reducing the climate change effects caused primarily by carbon dioxide emissions [1]. Offshore renewable energy plants offer the possibility of using the vast offshore space instead of the limited space onshore and are expected to be one of the most important energy sources in the next decades [2]. Nonetheless, offshore wind energy generation poses several challenges. The simulation of FSI is a powerful technique to analyze problems and to cope with these challenges. Most offshore wind turbines (OWTs) currently operating are using a fixed foundation in the sea bed. OWTs with a foundation in the sea bed become increasingly more costly with increasing water depths. Therefore most of the OWTs currently used operate in regions with low water depths, typically fairly near the shore. Nearshore regions are often more frequented, and public resistance to wind turbines is stronger there. Furthermore, the wind in nearshore regions is often weaker and less steady [3, p. 17].

Floating wind turbines (FWTs) that are attached to the sea bed with a mooring system offer a possibility to level the increasing cost in larger water depths, taking advantage of the better conditions further offshore. However, they are hard to design with classical

approaches since they show a strong motion in waves and wind which has to be considered in the design process. The support by the mooring system is also harder to analyze due to the fact that the simulation approach has to be able to capture the highly dynamic motion of the FWT. For this, FSI simulations offer a great possibility to overcome the limited design capabilities currently available. Nevertheless, FSI simulations are rarely used to analyze the behavior of a full floating wind turbine since they are computationally more demanding and the simultaneous solution of the fluid and the structural field requires special techniques for a stable and fast simulation. However, if the approach is successfully implemented, it offers a deep insight into the dynamic behavior and the structural response of the FWT, as shown in Sec. 5.1. These benefits can also be transferred to other floating structures like floating wave energy converters (WEC) as shown in Chapter 6.

With the increasing usage of more flexible materials such as carbon-fiber-reinforced polymers (CFRPs), it is becoming more and more important to consider FSI in a broader range of applications. Marine propellers are subjected to high dynamic loads, which – in combination with newer and more flexible propeller blades – leads to a significant influence of the fluid-structure interaction. CFRP propellers have already been tested for large commercial vessels, showing a significant improvement in efficiency [4]. However, the higher flexibility of CFRP propellers leads to a deformation of the propeller blades, which should not be neglected in the fluid simulation, as shown in Chapter 8. If the effect is considered in the design, it can even be used to improve the efficiency of the propeller and, in doing so, reduce the carbon dioxide emissions of the ship [5]. Furthermore, it is possible to reduce the cavitation on the propeller blades as shown in Sec. 8.2.5, which results in a longer propeller lifespan and reduces noise emissions, benefiting both the environment and the ship passengers.

Another example from marine engineering is the motion and deformation of ships due to the waves. The dynamic interaction between the ship and the sea waves is not easy to capture with classical modeling approaches. In addition to marine engineering, there are numerous other applications for which fluid-structure interaction is of considerable importance. Some examples in which FSI effects are especially important are the behavior of airbags in the automobile industry, the design of parachutes, and the analysis of cardiovascular diseases.

1.2 State of the art

The techniques to simulate complex fluid and structural problems quickly and accurately have advanced rapidly over the past decades. In combination with the increasing computational power, numerical methods have become more popular since they allow for an accurate analysis of complex engineering problems. With numerical methods like the finite element method (FEM), the boundary element method (BEM), and the finite volume method (FVM), it is possible to accurately solve nonlinear dynamic problems with complex geometries and unsteady boundary conditions. A large number of free and commercial simulation programs has been developed based on these methods for the simulation of fluid and structural problems. Still, most of the programs for this purpose are focused on either solving fluid or structural problems.

The development of FSI solvers which are capable of solving fluid and structural problems simultaneously is challenging for several reasons. The underlying differential equations that

are used for the description of the fluid and structural behavior are inherently different, which makes a fully integrated or monolithic solution of the fluid and structural equations difficult and requires the development of dedicated FSI solvers for this approach [6, p. xv].

An alternative approach for the solution of FSI problems is the partitioned approach as described, e.g., in [7, p. 6] and [8, p. 5]. Based on this approach, a number of software tools have been developed in recent years. Some of the academic and commercial software frameworks implementing the partitioned coupling for multi-physic problems are, for example, MpCCI [9], oomph-lib [10], and preCICE [11]. In 2018, the established commercial simulation framework ANSYS began to include the partitioned approach for the solution of multi-physics problems [12]. The implementation of these techniques in leading commercial suites is an indication of their increasing importance for industrial applications. Furthermore, important techniques for improving the coupling between the fluid and the structure such as convergence acceleration have been investigated and improved in the past years, as shown in [13, 14], for example.

Partitioned and monolithic FSI simulations have already been used for different maritime applications. For floating and classical offshore wind turbines, different studies have been performed to show the working principle, accuracy, and efficiency of FSI simulations for these application cases [15, 16, 17, 18]. The focus here is usually on the numerical methods and the validation of the approach rather than on the improvement of the structural design.

The importance of considering FSI effects in the numerical simulation of flexible fiber-reinforced propellers has been shown, e.g., in [19, 20]. For ship propellers, improvements in the design are possible when considering FSI effects – as shown by Young [21] and Liu et al. [5]. They showed that, with the bend-twisting coupling (see e.g. [22]), a load-dependent twisting of flexible anisotropic composite propeller blades can be used to improve the efficiency of the propeller. Mulcahy [23] also performed optimization of the fiber direction to improve the efficiency of fiber-reinforced polymer propellers without considering cavitation or dynamic effects.

At the Hamburg University of Technology, software codes have been developed for the fast and accurate simulation of structural, fluid, and coupled problems.

The BEM *pan*MARE [24] was developed at the "Institute for Fluid Dynamics and Ship Theory" as a method to solve fluid problems. Initially focusing on the simulation of ship propellers, this method has also been extended to other fluid problems such as floating wind turbines [25]. *pan*MARE serves as the solver for most of the fluid simulations conducted in this work.

The coupling manager *comana* [26] was developed in the research group "Numerical Structural Analysis with Application in Ship Technology" as a flexible multi-physics coupling library. This coupling library is the basis for the FSI simulations conducted in this work and is introduced in more detail in Sec. 4.6.

1.3 Purpose and scope of this thesis

For an engineer focused on calculating complex structures, it is often a major challenge to use the existing mathematical techniques in an optimal way to enable the solution of problems that could up to that point not be simulated in the same degree of accuracy and detail.

The purpose of this thesis is to enable this based on FSI simulations using partitioned coupling techniques to analyze the marine problems mentioned in Sec. 1.1. Numerical methods such as extrapolation and convergence acceleration techniques are combined efficiently to reduce computational time. The benefits of the FSI simulations are shown on realistic application cases. Here, the validation and verification of the FSI simulations are of major importance since the approach must be proven to be reliable. The approach is validated based on experimental results for the example of a submersible mixer, while the verification is performed with mesh convergence studies. After validating and verifying the approach, the FSI simulations are used to show the potential to improve the structure of complexly designed structures. This proves the benefit of the approach in the design of maritime structures.

This thesis is structured as follows: At the beginning, after the introduction, the theory for the simulation of structural and fluid problems is introduced in Chapters 2 and 3. Then, Chapter 4 serves to lay out the theoretical basis for the numerical methods that are necessary for partitioned coupling.

The maritime applications presented in Chapter 5 are the core of this thesis. Among these applications are two examples from the field of offshore renewable energies – which is one of the fastest-growing areas in the field of marine engineering. The examples are a self-aligning floating offshore wind turbine and a point wave energy converter. The third example is a multilayered submersible mixer. This example application is used as a validation case to prove the accuracy and reliability of the whole FSI setup. The last application case considered for this thesis is an anisotropic ship propeller simulation setup that is used to optimize the shape and material of the ship propeller. At the end of the thesis, a conclusion and an outlook are given in Chapter 6.

2 Structural mechanics

This chapter introduces the basics of structural mechanics and the nonlinear finite element method (FEM), which is the most common technique for solving complex structural problems. The theory of structural mechanics and nonlinear finite element method is part of many textbooks such as [27, 28, 29, 30], where a more extensive introduction to these topics can be found. The notation and the formulation of the FEM in this work follow the work of Wriggers in [27].

2.1 Balance equations

The basis for the derivation of the structural mechanics equations used in the FEM are the balance equations. The first important balance equation is the balance of linear momentum (also known as Newton's second law of motion). The balance of linear momentum states that the change of linear momentum is equal to the sum of all external forces acting on a body [27].

By applying the balance of linear momentum to an infinitesimal element and using the divergence operator div(), the vector form for the balance of linear momentum can be derived as

$$\varrho \ddot{\boldsymbol{d}} = \operatorname{div}(\boldsymbol{\sigma}) + \varrho \boldsymbol{b}. \tag{2.1}$$

Here, ϱ is the density and $\ddot{\boldsymbol{d}}$ is the acceleration vector (which is the second derivative with respect to time of the displacement vector). The Cauchy stress tensor $\boldsymbol{\sigma}$ contains the stress components σ_{ij}. The body force vector $\boldsymbol{b}$ captures volumetric loads. All vector and tensor-based quantities are written in **bold** to distinguish them from scalar-valued quantities.

Based on the balance of angular momentum, it is possible to show the symmetry of the Cauchy stress tensor $\boldsymbol{\sigma}$ [31, p. 104].

If the deformation of a body is large, it is important to distinguish between the undeformed reference configuration Ω and the deformed current configuration $\phi(\Omega)$. This distinction is visualized in Fig. 2.1. If an arbitrary point A with the coordinates $\boldsymbol{X}$ in the reference configuration is considered, the coordinates of the same point a in the current configuration are $\boldsymbol{x}$. The displacements $\boldsymbol{d}$ introduced above can then be computed as $\boldsymbol{x} - \boldsymbol{X} = \boldsymbol{d}$. Eq. (2.1) refers to the current configuration. Therefore, the divergence operator div() in Eq. (2.1) is written in small letters.

The deformation gradient tensor $\boldsymbol{F}$ can be introduced with these variables and the identity tensor $\mathbf{I}$ as

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \mathbf{I} + \frac{\partial \boldsymbol{d}}{\partial \boldsymbol{X}}. \tag{2.2}$$

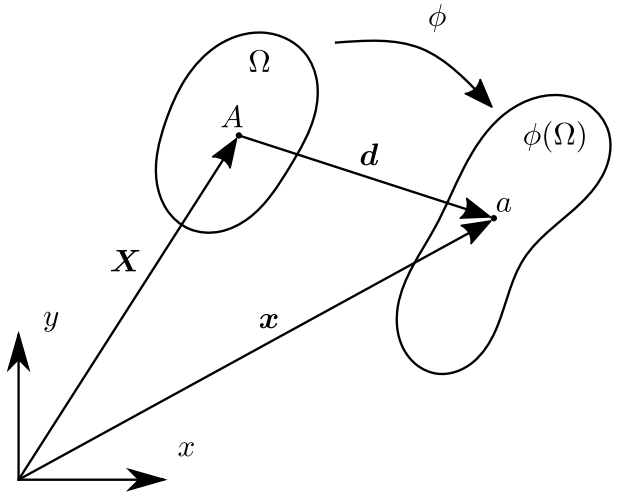


Figure 2.1: Transformation ϕ from reference to current configuration.

Similarly, the Green-Lagrange strain tensor $\boldsymbol{E}$ can be introduced as a symmetric nonlinear strain measure in the reference configuration as

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{F}^{\mathrm{T}}\boldsymbol{F} - \mathbf{I}). \tag{2.3}$$

Using Nanson's formula, the stress tensor $\boldsymbol{\sigma}$ in the current configuration $\phi(\Omega)$ can be transformed to the reference configuration Ω as [27, p. 37]

$$\boldsymbol{P} = \det(\boldsymbol{F})\,\boldsymbol{\sigma}\,\boldsymbol{F}^{-\mathrm{T}} = J\,\boldsymbol{\sigma}\,\boldsymbol{F}^{-\mathrm{T}}. \tag{2.4}$$

In Eq. (2.4), $\det(\boldsymbol{F})$ is the determinant of the deformation gradient and $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor. The abbreviation $J = \det(\boldsymbol{F})$ is used for the determinant of the deformation gradient.

The balance of linear momentum in the reference configuration can be expressed with the first Piola-Kirchhoff stress tensor $\boldsymbol{P}$ as

$$\varrho_0 \ddot{\boldsymbol{d}} = \mathrm{DIV}(\boldsymbol{P}) + \varrho_0 \boldsymbol{b}. \tag{2.5}$$

In Eq. (2.5), ϱ_0 is the density in the reference configuration, $\ddot{\boldsymbol{d}}$ is the acceleration, and $\boldsymbol{b}$ are possible volume loads such as gravitational load. Since it refers to the reference configuration, the divergence operator is represented here in capital letters as DIV().

Eq. (2.5) follows the Lagrangian formulation (also referred to as material formulation) in which the viewpoint follows the motion of the body [29, p. 498].

In order to work with symmetric stress tensors in the reference configuration, the equation $\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S}$ [27, p. 37] can be used to relate the first Piola-Kirchhoff stress tensor $\boldsymbol{P}$ to the symmetric second Piola-Kirchhoff stress tensor $\boldsymbol{S}$ using the deformation gradient $\boldsymbol{F}$.

2.2 Constitutive equations

To obtain a relationship between the Green-Lagrange strain tensor $\boldsymbol{E}$ and the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$, a constitutive equation must be introduced. The constitutive equation describes the stress response of a material to an applied strain.

2.2.1 Isotropic material modelling

The simplest material model for this is the St. Venant-Kirchhoff material model. This material model assumes a linear relationship between the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$ and the Green-Lagrange strain tensor $\boldsymbol{E}$ as [27, p. 45]

$$\boldsymbol{S} = \lambda \operatorname{tr}(\boldsymbol{E})\mathbf{I} + 2\,\mu\,\boldsymbol{E}. \tag{2.6}$$

In Eq. (2.6), $\mathbf{I}$ is the identity tensor and tr() is the trace operator. λ and μ are the first and second Lamé constants and have to be determined experimentally for a chosen material. Even though the material behavior in Eq. (2.6) is linear elastic, this material model is still capable of capturing nonlinear geometric behavior since it is embedded in the nonlinear geometric framework described in the previous section. In Voigt notation (indicated by the superscript V), Eq. (2.6) can also be written as

$$\boldsymbol{S}^V = \boldsymbol{D}\,\boldsymbol{E}^V. \tag{2.7}$$

With $\boldsymbol{S}^V = [\mathrm{S}_{11}, \mathrm{S}_{22}, \mathrm{S}_{33}, \mathrm{S}_{12}, \mathrm{S}_{23}, \mathrm{S}_{13}]^{\mathrm{T}}$, $\boldsymbol{E}^V = [\mathrm{E}_{11}, \mathrm{E}_{22}, \mathrm{E}_{33}, 2\mathrm{E}_{12}, 2\mathrm{E}_{23}, 2\mathrm{E}_{13}]^{\mathrm{T}}$ and the symmetric elasticity matrix $\boldsymbol{D}$ written as

$$\boldsymbol{D} = \begin{bmatrix} \lambda+2\,\mu & \lambda & \lambda & 0 & 0 & 0 \\ & \lambda+2\,\mu & \lambda & 0 & 0 & 0 \\ & & \lambda+2\,\mu & 0 & 0 & 0 \\ & & & \mu & 0 & 0 \\ & sym. & & & \mu & 0 \\ & & & & & \mu \end{bmatrix}. \tag{2.8}$$

It is more common to describe the material behavior of a linear elastic material with the Young's modulus E and the Poisson's ratio ν. The Lamé constants can be related to the Young's modulus and the Poisson's ratio by using the following relationship [29, p. 74]

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \qquad \mu = \frac{E}{2(1+\nu)}. \tag{2.9}$$

The St. Venant-Kirchhoff material model is suitable for small strains and large rotations. Since it is not reliable for large strains, a different material description is necessary for such cases. One common material model for an elastic material behavior under large strain is the Neo-Hooke material description for which the relationship [31, p. 124]

$$\boldsymbol{S} = \mu(\mathbf{I} - \boldsymbol{C}^{-1}) + \lambda(\ln(J))\boldsymbol{C}^{-1} \tag{2.10}$$

holds. $\boldsymbol{C}$ is the right Cauchy-Green tensor that relates to the Green-Lagrange strain tensor as $\boldsymbol{C} = 2\,\boldsymbol{E} + \boldsymbol{I}$. The determinant of the deformation gradient J relates the volume in the current configuration v to the volume in the reference configuration V as $dv = J\,dV$. Since the first Piola-Kirchhoff stress $\boldsymbol{S}$ goes to $-\infty$ if J goes to 0 and since $\boldsymbol{S}$ goes to ∞ if J goes to ∞, the Neo-Hooke material model also fulfills the growth condition.

The material models introduced so far all assume an isotropic material behavior, which means that the structural response of the material is independent of the direction in which the stresses are applied.

2.2.2 Anisotropic material modelling

In order to model anisotropic materials like fiber-reinforced polymers, an anisotropic material model is necessary.

Anisotropic material models can be categorized according to their symmetry conditions. To this end, a symmetry plane is defined as a plane on which a material element can be mirrored without changing the behavior of the material. [32, p. 84]

In the most general case of a triclinic material model, there are no symmetry planes. Since the elasticity matrix $\boldsymbol{D}$ in Eq. (2.7) is symmetric, this general anisotropic (triclinic) material has 21 independent entries in the elasticity matrix. For an orthotropic material, three orthogonal symmetry planes exist, each with normals defined in correspondence to the three base vectors $\mathbf{e}_x, \mathbf{e}_y$ and $\mathbf{e}_z$. For a transverse isotropic material, there exists an infinite number of symmetry planes – and the material can be defined in such a way that every combination of the two base vectors $n_1\mathbf{e}_y + n_2\mathbf{e}_z$ as well as the vector $\mathbf{e}_x$ is the normal to a symmetry plane. The material properties are therefore invariant under every rotation around the x-axis [33, p. 36]. With this definition, the transverse isotropic material is a special case of orthotropic materials.

Unidirectional fiber-reinforced polymer layers can be modeled with a transverse isotropic material model [34, p. 181]. With the inverse matrix $\boldsymbol{D}^{-1}$ to the elasticity matrix, Eq. (2.7) can be transformed to

$$\boldsymbol{E}^V = \boldsymbol{D}^{-1}\,\boldsymbol{S}^V. \tag{2.11}$$

For orthotropic materials, the number of independent entries in the elasticity matrix is reduced to nine [32, p. 88]. Hence, the inverse elasticity matrix can also be expressed in terms of nine different parameters. These parameters are the Young's moduli in E in x-, y-, and z-direction, the Poisson's ratios ν_{xy}, ν_{yz}, and ν_{xz} and the shear modulus G_{xy}, G_{yz}, and G_{xz}. The shear moduli G_{ij} is equal to the second Lamé constant μ_{ij}. With this, $\boldsymbol{D}^{-1}$ can be written as [35, p. 88]

$$\boldsymbol{D}^{-1} = \begin{bmatrix} \frac{1}{E_x} & -\frac{\nu_{yx}}{E_y} & -\frac{\nu_{zx}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{xy}}{E_x} & \frac{1}{E_y} & -\frac{\nu_{zy}}{E_z} & 0 & 0 & 0 \\ -\frac{\nu_{xz}}{E_x} & -\frac{\nu_{yz}}{E_y} & \frac{1}{E_z} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{xy}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{yz}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{xz}} \end{bmatrix}. \tag{2.12}$$

For a symmetric elasticity matrix $\boldsymbol{D}$, the relation $\nu_{xy}/\nu_{yx} = E_y/E_x$ must hold. With this relationship, the Poisson's ratios ν_{xy} and ν_{yx} can be related [34, p. 183].

For the transverse isotropic materials introduced earlier, the properties in the y- and z-direction are the same. Therefore, the following parameters become equal: $E_y = E_z$, $\nu_{xy} = \nu_{xz}$ and $G_{xy} = G_{xz}$. The shear modulus G_{yz} can now be expressed as

$$G_{yz} = \mu_{yz} = \frac{E_y}{2(1+\nu_{yz})}. \tag{2.13}$$

In total, the number of independent parameters is reduced to 5. To keep the elasticity matrix $\boldsymbol{D}$ positive definite and, thus, the simulation stable, additional constraints are required are required – see Li [35], for example.

Fiber-reinforced polymer materials are usually designed and modeled to be either orthotropic or transverse isotropic. In order to obtain the material parameters for an exact sample of fiber-reinforced polymers, extensive experimental tests are necessary to obtain the independent parameters. Furthermore, the exact composition of the material and the manufacturing process have a major impact on the material parameters. Therefore, material parameters for a fiber-reinforced polymer can only be transferred to another if the composition and the manufacturing process are very similar.

2.3 Weak form

To obtain the weak form, Eq. (2.5) is multiplied with an arbitrary test function $\boldsymbol{\eta}$ using the scalar product and the resulting equation is integrated over the whole domain Ω. This yields

$$\int_\Omega \varrho_0 \ddot{\boldsymbol{d}} \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega - \int_\Omega \mathrm{DIV}(\boldsymbol{P}) \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega - \int_\Omega \varrho_0 \boldsymbol{b} \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega = 0. \tag{2.14}$$

According to the product rule, an expression that includes the second term of Eq. (2.14) can be stated as [27, p. 520]

$$\int_\Omega \mathrm{DIV}(\boldsymbol{P}^\mathrm{T}\boldsymbol{\eta}) \, \mathrm{d}\Omega = \int_\Omega \mathrm{DIV}(\boldsymbol{P}) \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{P} \cdot \mathrm{GRAD}(\boldsymbol{\eta}) \, \mathrm{d}\Omega. \tag{2.15}$$

Since the gradient operator refers to the reference configuration, it is written in capital letters. With the divergence theorem [36, p. 116], [37, p. 140], the term on the left-hand side of Eq. (2.15) can be transformed into an integral over the boundary of the domain

$$\int_\Omega \mathrm{DIV}(\boldsymbol{P}^\mathrm{T}\boldsymbol{\eta}) \, \mathrm{d}\Omega = \int_{\Gamma_N} \boldsymbol{P}\boldsymbol{N} \cdot \boldsymbol{\eta} \, \mathrm{d}\Gamma = \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{\eta} \, \mathrm{d}\Gamma. \tag{2.16}$$

The product of the first Piola-Kirchhoff stress $\boldsymbol{P}$ with normal vector $\boldsymbol{N}$ yields the traction vector $\bar{\boldsymbol{t}}$.

The second term on the right-hand side of Eq. (2.15) can then be substituted by using the variation of the Green-Lagrange strain tensor $\delta\boldsymbol{E}$ as [27, p. 84]

$$\int_\Omega \boldsymbol{P} \cdot \mathrm{GRAD}(\boldsymbol{\eta}) \, \mathrm{d}\Omega = \int_\Omega \boldsymbol{F}\boldsymbol{S} \cdot \mathrm{GRAD}(\boldsymbol{\eta}) \, \mathrm{d}\Omega = \int_\Omega \boldsymbol{S} \cdot \delta\boldsymbol{E} \, \mathrm{d}\Omega \tag{2.17}$$

Rearranging Eq. (2.15) and substituting Eq. (2.16) and (2.17) leads to

$$\int_\Omega \mathrm{DIV}(\boldsymbol{P}) \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega = \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{\eta} \, \mathrm{d}\Gamma - \int_\Omega \boldsymbol{S} \cdot \delta\boldsymbol{E} \, \mathrm{d}\Omega. \tag{2.18}$$

After inserting Eq. (2.18) into Eq. (2.14), the weak form is obtained as

$$\int_\Omega \varrho_0 \ddot{\boldsymbol{d}} \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega - \int_{\Gamma_N} \bar{\boldsymbol{t}} \cdot \boldsymbol{\eta} \, \mathrm{d}\Gamma - \int_\Omega \varrho_0 \boldsymbol{b} \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega + \int_\Omega \boldsymbol{S} \cdot \delta\boldsymbol{E} \, \mathrm{d}\Omega = 0. \tag{2.19}$$

2.4 Spatial discretization

To solve the weak form (2.19) in a given domain Ω, the equation must be discretized in a spatial and – for a dynamic simulation – also in a temporal way. In the spatial discretization process, the domain Ω is subdivided into a set of n finite elements as

$$\Omega \approx \bigcup_{e=1}^{n} \Omega_e. \tag{2.20}$$

To solve the weak form numerically, the displacement field $\boldsymbol{d}$ for an element can be approximated with the spatially discretized displacements $\boldsymbol{d}_{\mathrm{d},i}$ and the shape function $N_i(\boldsymbol{\xi})$ as

$$\boldsymbol{d}_h = \sum_{i=1}^{m} N_i(\boldsymbol{\xi}) \boldsymbol{d}_{\mathrm{d},i} \tag{2.21}$$

with m being the number of shape functions per element. The index d of the discretized displacements $\boldsymbol{d}_{\mathrm{d},i}$ is used to indicate that these displacements are discrete. The index i is the number of the shape function. When using the *isoparametric* concept, the geometry is described using the same set of shape functions N_i that is used for the approximation of the displacement field. The displacements $\boldsymbol{d}_{\mathrm{d},i}$ are the unknowns for which the whole system is solved at the end. Since the viewpoint follows the motion of the body in the Lagrangian formulation, the elements follow the deformation of the body.

In the so-called h-version of the FEM, the discretization error is controlled by changing the number of finite elements n. In the so-called p-version of the FEM, the discretization error is controlled by increasing the polynomial degree of the shape functions. A combination of both methods is called the hp-version of the FEM.

2.4.1 Shape functions

Here, the concept of shape functions is introduced using a two-dimensional example applying a quadrilateral element. The concept can easily be extended to three dimensions and different element types.

The shape functions are defined on a standard element in the reference coordinate system with the local coordinates ξ and η in the space from -1 to 1 in both directions, as depicted in Fig. 2.2 on the left-hand side. The nodes are numbered as $[P_1 \ ... \ P_4]$ and have the corresponding coordinates $[X_{1,d} \ ... \ X_{4,d}]$ and $[Y_{1,d} \ ... \ Y_{4,d}]$ in the global coordinate system on the right-hand side of Fig. 2.2.

A mapping function $\boldsymbol{Q}^e(\xi, \eta)$ is introduced to map the reference element coordinates ξ and η to the global coordinates x and y. The mapping functions for a two-dimensional element can be written as

$$\begin{aligned} x_e &= Q_x^e(\xi, \eta) = \sum_{i=1}^{m} N_i(\xi, \eta) X_{i,d} \\ y_e &= Q_y^e(\xi, \eta) = \sum_{i=1}^{m} N_i(\xi, \eta) Y_{i,d}\,. \end{aligned} \tag{2.22}$$

Shape functions are commonly defined by using Lagrange polynomials that fulfill the partition of unity ($\sum_i^m N_i = 1$) and take a value of 1 on their own nodes and zero on all

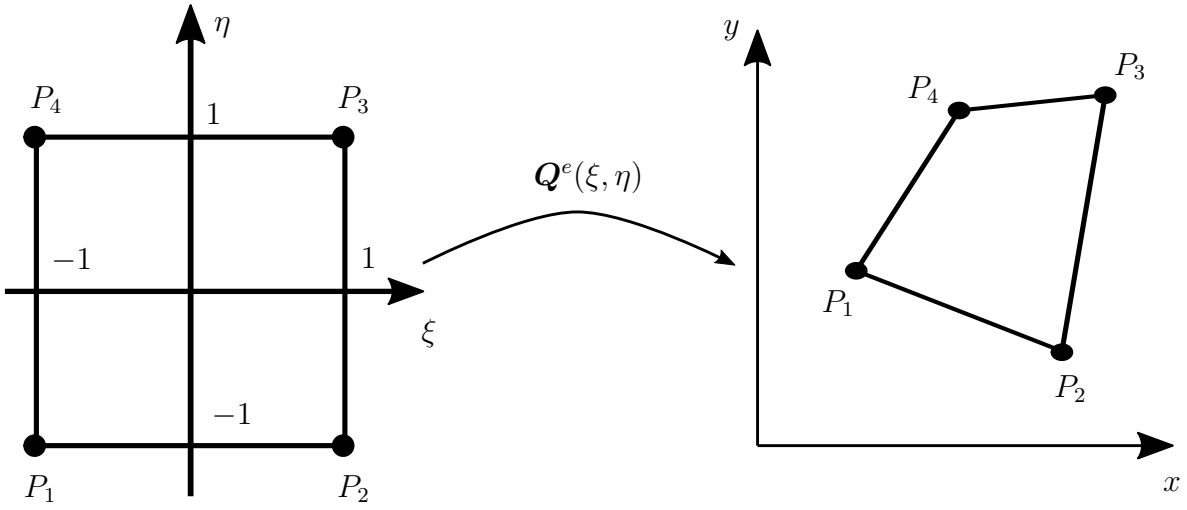


Figure 2.2: Mapping of a two-dimensional quadrilateral element

other nodes. This property has the advantage that the primary solution vector comprises the displacement at nodes.

For a two-dimensional quadrilateral element with $m = 4$, the bilinear shape functions are the most simple to choose. They can be constructed by combining the one-dimensional linear shape functions $N_{\xi 1} = \frac{1}{2}(\xi - 1)$ and $N_{\xi 2} = \frac{1}{2}(\xi + 1)$ with the respective functions for the second coordinate direction $N_{\eta 1}$ and $N_{\eta 2}$ to

$$\begin{aligned} N_1(\xi,\eta) &= N_{\xi 1} N_{\eta 1} \qquad & N_2(\xi,\eta) &= N_{\xi 2} N_{\eta 1} \\ N_3(\xi,\eta) &= N_{\xi 2} N_{\eta 2} \qquad & N_4(\xi,\eta) &= N_{\xi 1} N_{\eta 2}. \end{aligned} \tag{2.23}$$

In principle, all types of functions could be used as shape functions. By using, for example, higher-order polynomials and an increasing number of shape functions per element, a superior convergence behavior can be achieved for an increasing number of degrees of freedom – compared to an approach that increases the number of elements. With the extension to the *hp*-version of the FEM, this is also possible for problems that contain singularities [28, p. 293].

2.4.2 Discretization of the weak form

To solve the weak form Eq. (2.19) in a physical domain, it must be discretized. The variation of the Green-Lagrange strain tensor in (2.17) can be substituted with [27, p. 84]

$$\int_\Omega \boldsymbol{S} \cdot \delta \boldsymbol{E} \, \mathrm{d}\Omega = \int_\Omega \boldsymbol{S} \cdot \boldsymbol{F}^{\mathrm{T}} \, \mathrm{GRAD}(\boldsymbol{\eta}) \, \mathrm{d}\Omega = \boldsymbol{S} \, \frac{1}{2} \cdot (\boldsymbol{F}^{\mathrm{T}} \, \mathrm{GRAD}(\boldsymbol{\eta}) + \, \mathrm{GRAD}(\boldsymbol{\eta})^{\mathrm{T}} \, \boldsymbol{F}). \tag{2.24}$$

On the element level, the shape functions can be summarized in a matrix as

$$\boldsymbol{N} = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix}. \tag{2.25}$$

With the matrix $\boldsymbol{N}$, the displacement function $\boldsymbol{d}$ can be substituted by the discrete FEM approximation

$$\boldsymbol{d} = \boldsymbol{N}\boldsymbol{d}_{\mathrm{d}}. \tag{2.26}$$

The vector $\boldsymbol{d}_{\mathrm{d}}$ contains the spatially discretized displacements with which the shape functions are defined and which have to be computed at the end. For the shape functions defined with Eq. (2.22) and Eq. (2.23), $\boldsymbol{d}_{\mathrm{d}}$ can be written as

$$\boldsymbol{d}_{\mathrm{d}} = [D_{x,1},\ D_{y,1},\ D_{x,2},\ D_{y,2},\ D_{x,3},\ D_{y,3},\ D_{x,4},\ D_{y,4}]^{\mathrm{T}} \tag{2.27}$$

and contains the displacements at the nodes. Here, the spatially discretized displacement vector $\boldsymbol{d}_{\mathrm{d}}$ is written with the index d to distinguish the quantity from the continuously defined displacements $\boldsymbol{d}$.

Since the Bubnov-Galerkin approach is used, the test function $\boldsymbol{\eta}$ is discretized with the same shape functions as the displacements $\boldsymbol{d}_{\mathrm{d}}$ as

$$\boldsymbol{\eta} = \boldsymbol{N}\boldsymbol{h} \tag{2.28}$$

where the virtual displacements $\boldsymbol{h}$ can be written as

$$\boldsymbol{h} = [H_{x,1},\ H_{y,1},\ H_{x,2},\ H_{y,2},\ H_{x,3},\ H_{y,3},\ H_{x,4},\ H_{y,4}]^{\mathrm{T}}. \tag{2.29}$$

With the virtual displacements $\boldsymbol{h}$, the discretized version of the variation of the Green-Lagrange strain tensor for one finite element can be written in Voigt notation as [27, p. 124]

$$\delta\boldsymbol{E}_e = \boldsymbol{B}_l\boldsymbol{h} \tag{2.30}$$

For the example of a two-dimensional element, matrix $\boldsymbol{B}_l$ is defined as [27, p. 131]

$$\boldsymbol{B}_l = \begin{bmatrix} \boldsymbol{B}_{l1} & \boldsymbol{B}_{l2} & \boldsymbol{B}_{l3} & \boldsymbol{B}_{l4} \end{bmatrix} \tag{2.31}$$

with

$$\boldsymbol{B}_{li} = \begin{bmatrix} F_{11}\,N_{i,X} & F_{21}\,N_{i,X} \\ F_{12}\,N_{i,Y} & F_{22}\,N_{i,Y} \\ F_{11}\,N_{i,Y} + F_{12}\,N_{i,X} & F_{21}\,N_{i,Y} + F_{22}\,N_{i,X} \end{bmatrix}. \tag{2.32}$$

The terms $N_{i,X}$ and $N_{i,Y}$ are the derivatives of the shape function N_i with respect to X and Y direction. Since the shape functions are defined in the reference coordinate system in terms of ξ and η, a transformation is necessary to compute the derivatives in the global coordinate system. This transformation between the global and reference coordinate system can be written as

$$\begin{bmatrix} N_{i,X} \\ N_{i,Y} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_i}{\partial X} \\ \frac{\partial N_i}{\partial Y} \end{bmatrix} = \boldsymbol{J}_e^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix}. \tag{2.33}$$

$\boldsymbol{J}_e$ is the Jacobi matrix here, which is defined as

$$\boldsymbol{J}_e = \begin{bmatrix} \frac{\partial X}{\partial \xi} & \frac{\partial Y}{\partial \xi} \\ \frac{\partial X}{\partial \eta} & \frac{\partial Y}{\partial \eta} \end{bmatrix}. \tag{2.34}$$

On the element level, the weak form (Eq. (2.19)) can be written as

$$\int_{\Omega^e} \varrho_0 \ddot{\boldsymbol{d}} \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega^e + \int_{\Omega^e} \boldsymbol{S} \cdot \delta \boldsymbol{E} \, \mathrm{d}\Omega^e - \int_{\Omega^e} \varrho_0 \boldsymbol{b} \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega^e - \int_{\Gamma_N^e} \bar{\boldsymbol{t}} \cdot \boldsymbol{\eta} \, \mathrm{d}\Gamma = 0. \tag{2.35}$$

The first term in Eq. (2.35) contains the scalar product of two vector-valued functions $\ddot{\boldsymbol{d}} \cdot \boldsymbol{\eta}$ that can be reformed to a different notation as

$$\ddot{\boldsymbol{d}} \cdot \boldsymbol{\eta} = \boldsymbol{\eta} \cdot \ddot{\boldsymbol{d}} = \boldsymbol{\eta}^{\mathrm{T}} \ddot{\boldsymbol{d}}. \tag{2.36}$$

Substituting Eq. (2.26) and Eq. (2.28) gives

$$\boldsymbol{\eta}^{\mathrm{T}} \ddot{\boldsymbol{d}} = (\boldsymbol{N}\boldsymbol{h})^{\mathrm{T}} \boldsymbol{N} \ddot{\boldsymbol{d}}_{\mathrm{d}} = \boldsymbol{h}^{\mathrm{T}} \boldsymbol{N}^{\mathrm{T}} \boldsymbol{N} \ddot{\boldsymbol{d}}_{\mathrm{d}}. \tag{2.37}$$

With Eq. (2.36) and Eq. (2.37), the first term of the weak form can be reformed to

$$\int_{\Omega^e} \varrho_0 \ddot{\boldsymbol{d}} \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega^e = \int_{\Omega^e} \varrho_0 \boldsymbol{h}^{\mathrm{T}} \boldsymbol{N}^{\mathrm{T}} \boldsymbol{N} \ddot{\boldsymbol{d}}_{\mathrm{d}} \, \mathrm{d}\Omega^e = \boldsymbol{h}^{\mathrm{T}} \int_{\Omega^e} \varrho_0 \boldsymbol{N}^{\mathrm{T}} \boldsymbol{N} \, \mathrm{d}\Omega^e \, \ddot{\boldsymbol{d}}_{\mathrm{d}} = \boldsymbol{h}^{\mathrm{T}} \boldsymbol{M} \, \ddot{\boldsymbol{d}}_{\mathrm{d}}. \tag{2.38}$$

With this, the mass matrix $\boldsymbol{M}$ is introduced as

$$\boldsymbol{M} = \int_{\Omega^e} \varrho_0 \boldsymbol{N}^{\mathrm{T}} \boldsymbol{N} \, \mathrm{d}\Omega^e. \tag{2.39}$$

The discretized variation of the Green-Lagrange strain tensor is now restated in a similar way by using the internal load vector $\boldsymbol{s}(\boldsymbol{d}_{\mathrm{d}})$ as

$$\int_{\Omega^e} \boldsymbol{S}_e \cdot \delta \boldsymbol{E}_e \, \mathrm{d}\Omega^e = \int_{\Omega^e} \boldsymbol{S}_e \cdot \boldsymbol{B}_l \boldsymbol{h} \, \mathrm{d}\Omega^e = \int_{\Omega^e} (\boldsymbol{B}_l \boldsymbol{h})^{\mathrm{T}} \boldsymbol{S}_e \, \mathrm{d}\Omega^e = \boldsymbol{h}^{\mathrm{T}} \int_{\Omega^e} \boldsymbol{B}_l^{\mathrm{T}} \boldsymbol{S}_e \, \mathrm{d}\Omega^e = \boldsymbol{h}^{\mathrm{T}} \boldsymbol{s}(\boldsymbol{d}_{\mathrm{d}}) \tag{2.40}$$

With this, the internal load vector $\boldsymbol{s}(\boldsymbol{d}_{\mathrm{d}})$ is introduced as

$$\boldsymbol{s}(\boldsymbol{d}_{\mathrm{d}}) = \int_{\Omega^e} \boldsymbol{B}_l^{\mathrm{T}} \boldsymbol{S}_e \, \mathrm{d}\Omega^e. \tag{2.41}$$

The two last terms of the weak form that correspond to body loads and Neumann boundary conditions are discretized similarly as

$$\int_{\Omega^e} \varrho_0 \boldsymbol{b} \cdot \boldsymbol{\eta} \, \mathrm{d}\Omega^e + \int_{\Gamma_N^e} \bar{\boldsymbol{t}} \cdot \boldsymbol{\eta} \, \mathrm{d}\Gamma = \boldsymbol{h}^T \int_{\Omega^e} \varrho_0 \boldsymbol{N}^{\mathrm{T}} \boldsymbol{b} \, \mathrm{d}\Omega^e + \boldsymbol{h}^T \int_{\Gamma_N^e} \boldsymbol{N}^{\mathrm{T}} \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma = \boldsymbol{h}^{\mathrm{T}} \boldsymbol{f} \tag{2.42}$$

With this, the internal load vector $\boldsymbol{s}(\boldsymbol{d}_{\mathrm{d}})$ is introduced as

$$\boldsymbol{f} = \int_{\Omega^e} \varrho_0 \boldsymbol{N}^{\mathrm{T}} \boldsymbol{b} \, \mathrm{d}\Omega^e + \int_{\Gamma_N^e} \boldsymbol{N}^{\mathrm{T}} \bar{\boldsymbol{t}} \, \mathrm{d}\Gamma. \tag{2.43}$$

If Eq. (2.38) to Eq. (2.42) are inserted into Eq. (2.19), the spatially discretized weak form can be written as

$$\boldsymbol{h}^{\mathrm{T}} \boldsymbol{M} \ddot{\boldsymbol{d}}_{\mathrm{d}} + \boldsymbol{h}^{\mathrm{T}} \boldsymbol{s}(\boldsymbol{d}_{\mathrm{d}}) = \boldsymbol{h}^{\mathrm{T}} \boldsymbol{f}. \tag{2.44}$$

Eq. (2.44) has so far been written only for one finite element. After the matrices for all the elements are computed, an assembly step is necessary to combine the element matrices into global matrices. More about this assembly step can be found, e.g., in [29, p. 983 ff].

Due to the complex shape of the finite elements, the integration is usually carried out in the reference element shown in Fig. 2.2 – and the integral is transformed using Eq. (2.33).

The integration is usually performed with Gaussian quadrature since an analytic integration is in the general case not possible due to the inverse of the Jacobian in Eq. (2.33), which is introduced by the transformation and contains polynomial fractions.

Gaussian quadrature is one of the most common numerical integration techniques, not only in the FEM. The integration with Gaussian quadrature is done by evaluating the integrand at predefined Gaussian points and multiplying the evaluated function with Gaussian weights, defined in such a way that a polynomial up to a certain order can be integrated exactly. Details about numerical integration using Gaussian quadrature can be found, e.g., in [38, p. 171].

2.5 Temporal discretization

Since the virtual displacements $\boldsymbol{h}$ can be chosen arbitrarily, Eq. (2.44) can be transformed to [27, p. 128]

$$\boldsymbol{M}\ddot{\boldsymbol{d}}_{\mathrm{d}} + \boldsymbol{s}(\boldsymbol{d}_{\mathrm{d}}) = \boldsymbol{f}. \tag{2.45}$$

Eq. (2.45) is the nonlinear semi-discrete equation of motion and, unlike Eq. (2.44), is a vector-valued instead of a scalar-valued one.

In Eq. (2.45), $\boldsymbol{M}$ is the mass matrix and $\boldsymbol{s}(\boldsymbol{d}_{\mathrm{d}})$ is the vector of internal loads that depends non-linearly on the displacements $\boldsymbol{d}_{\mathrm{d}}$. The external loads $\boldsymbol{f}$ stem from the integration of the tractions and body loads.

To introduce damping effects in the dynamic equation, mass or stiffness-proportional Rayleigh damping can be included in the form of a matrix

$$\boldsymbol{C} = \alpha\boldsymbol{M} + \beta\boldsymbol{K}_T. \tag{2.46}$$

Here, $\boldsymbol{K}_T$ corresponds to the tangent stiffness matrix, which will be explained in more detail later on. If the desired damping ratios of the modes are known, α and β can be estimated based on modal analysis [29, p. 797].

With Rayleigh damping, Eq. (2.45) can be rewritten as

$$\boldsymbol{M}\ddot{\boldsymbol{d}}_{\mathrm{d}} + \boldsymbol{C}\dot{\boldsymbol{d}}_{\mathrm{d}} + \boldsymbol{s}(\boldsymbol{d}_{\mathrm{d}}) = \boldsymbol{f}. \tag{2.47}$$

In the next step, Eq. (2.47) must also be discretized with respect to time. A number of different methods exist to solve the dynamic equation of motion. The main difference between those methods is the point in time at which Eq. (2.47) is evaluated in the fully discretized equations.

2.5.1 Central difference method

In the central difference method, the forces $\boldsymbol{f}_t$, the displacements $\boldsymbol{d}_t$, and the derivatives ($\dot{\boldsymbol{d}}_t$ and $\ddot{\boldsymbol{d}}_t$) of the displacements in Eq. (2.47) are evaluated at the current time step t. Eq. (2.47) can with this assumption be rewritten as

$$\boldsymbol{M}\ddot{\boldsymbol{d}}_t + \boldsymbol{C}\dot{\boldsymbol{d}}_t + \boldsymbol{s}(\boldsymbol{d}_t) = \boldsymbol{f}_t. \tag{2.48}$$

By assuming a constant velocity between $t - \Delta t$ and $t + \Delta t$ and a fixed time-step size Δt, the velocities $\dot{\boldsymbol{d}}_t$ in the current time step can be approximated as

$$\dot{\boldsymbol{d}}_t = \frac{\boldsymbol{d}_{t+\Delta t} - \boldsymbol{d}_{t-\Delta t}}{2\Delta t}. \tag{2.49}$$

The accelerations $\ddot{\boldsymbol{d}}_t$ in the current time step can be approximated by combining the two central difference quotients

$$\dot{\boldsymbol{d}}_t = \frac{\boldsymbol{d}_{t+\Delta t/2} - \boldsymbol{d}_{t-\Delta t/2}}{\Delta t} \tag{2.50}$$

and

$$\ddot{\boldsymbol{d}}_t = \frac{\dot{\boldsymbol{d}}_{t+\Delta t/2} - \dot{\boldsymbol{d}}_{t-\Delta t/2}}{\Delta t} \tag{2.51}$$

which can be derived similarly to Eq. (2.49) as [39, p. 42]

$$\ddot{\boldsymbol{d}}_t = \frac{\boldsymbol{d}_{t+\Delta t} - 2\,\boldsymbol{d}_t + \boldsymbol{d}_{t-\Delta t}}{\Delta t^2}. \tag{2.52}$$

If the approximation of the velocity (Eq. (2.49)) and the approximation of the acceleration (Eq. (2.52)) is inserted into Eq. (2.48), the following equation is obtained

$$\boldsymbol{M}\frac{\boldsymbol{d}_{t+\Delta t} - 2\,\boldsymbol{d}_t + \boldsymbol{d}_{t-\Delta t}}{\Delta t^2} + \boldsymbol{C}\frac{\boldsymbol{d}_{t+\Delta t} - \boldsymbol{d}_{t-\Delta t}}{2\Delta t} + \boldsymbol{s}(\boldsymbol{d}_t) = \boldsymbol{f}_t. \tag{2.53}$$

Rearranging leads to

$$\left(\boldsymbol{M}\frac{1}{\Delta t^2} + \boldsymbol{C}\frac{1}{2\Delta t}\right)\boldsymbol{d}_{t+\Delta t} = \boldsymbol{f}_t + \boldsymbol{M}\frac{2\,\boldsymbol{d}_t - \boldsymbol{d}_{t-\Delta t}}{\Delta t^2} + \boldsymbol{C}\frac{\boldsymbol{d}_{t-\Delta t}}{2\Delta t} - \boldsymbol{s}(\boldsymbol{d}_t). \tag{2.54}$$

With Eq. (2.54), the solution for the displacements $\boldsymbol{d}_{t+\Delta t}$ at the next time step can be directly computed based on the current and previous solution for the displacements and their derivatives by solving a linear equation system. Therefore, the central difference method belongs to the class of explicit time integration methods.

If the mass matrix $\boldsymbol{M}$ is diagonalized (e.g. with mass lumping techniques) and the damping matrix $\boldsymbol{C}$ is neglected, Eq. (2.54) can be even solved without a computationally expensive solution of a linear system. With this approach, it is possible to efficiently evaluate highly dynamic problems that require a small time-step size since the computational effort per time step is comparatively low. The disadvantage of the central difference method is the instability of the method for a time-step size larger than the critical time-step size. Definitions for the critical time-step size can be found, e.g., in Wriggers [27, p. 211].

2.5.2 Newmark method

In the Newmark method [40], the displacements and its derivatives in Eq. (2.47) are evaluated at the next time step $t + \Delta t$, which leads to the following expression

$$\boldsymbol{M}\ddot{\boldsymbol{d}}_{t+\Delta t} + \boldsymbol{C}\dot{\boldsymbol{d}}_{t+\Delta t} + \boldsymbol{s}(\boldsymbol{d}_{t+\Delta t}) = \boldsymbol{f}_{t+\Delta t}. \tag{2.55}$$

To approximate the velocities in the next time step $\dot{\boldsymbol{d}}_{t+\Delta t}$ a central difference expression is used in which the accelerations $\ddot{\boldsymbol{d}}$ are linearly interpolated between the accelerations of the current time step and the next time step. With a weighting factor γ determining the point between the current and the next time step where the interpolated accelerations are evaluated, this leads to [40, p. 71]

$$\dot{\boldsymbol{d}}_{t+\Delta t} = \dot{\boldsymbol{d}}_t + (1-\gamma)\ddot{\boldsymbol{d}}_t\Delta t + \gamma\,\ddot{\boldsymbol{d}}_{t+\Delta t}\Delta t. \tag{2.56}$$

To find an expression for the displacements in the next time step $\boldsymbol{d}_{t+\Delta t}$, a second-order Taylor expansion is required – derived, e.g., by integrating the acceleration twice [39, p. 124].

$$\boldsymbol{d}_{t+\Delta t} = \boldsymbol{d}_t + \dot{\boldsymbol{d}}_t \Delta t + [(1-2\beta)\ddot{\boldsymbol{d}}_t + 2\beta\, \ddot{\boldsymbol{d}}_{t+\Delta t}]\frac{\Delta t^2}{2}. \tag{2.57}$$

The accelerations here are interpolated similarly to Eq. (2.56) with $\gamma = 2\beta$.

To solve Eq. (2.55), the equation must be rewritten as a nonlinear system of equations with only one unknown vector. If the system is solved directly for the displacements $\boldsymbol{d}_{t+\Delta t}$ in the next time step, Eq. (2.57) can be rearranged to provide an expression for the accelerations in the next time step $\ddot{\boldsymbol{d}}_{t+\Delta t}$ as

$$\ddot{\boldsymbol{d}}_{t+\Delta t} = \boldsymbol{d}_{t+\Delta t}\frac{1}{\beta \Delta t^2} - \boldsymbol{d}_t \frac{1}{\beta \Delta t^2} - \dot{\boldsymbol{d}}_t \frac{1}{\beta \Delta t} - \ddot{\boldsymbol{d}}_t (\frac{1}{2\beta} - 1). \tag{2.58}$$

If Eq. (2.58) is inserted into Eq. (2.56), a corresponding expression for the velocities $\dot{\boldsymbol{d}}_{t+\Delta t}$ in the next time step can be found as

$$\begin{aligned} \dot{\boldsymbol{d}}_{t+\Delta t} =& \dot{\boldsymbol{d}}_t + (1-\gamma)\ddot{\boldsymbol{d}}_t \Delta t + \boldsymbol{d}_{t+\Delta t}\frac{\gamma}{\beta \Delta t} - \boldsymbol{d}_t \frac{\gamma}{\beta \Delta t} - \dot{\boldsymbol{d}}_t \frac{\gamma}{\beta} - \ddot{\boldsymbol{d}}_t \gamma\, \Delta t (\frac{1}{2\beta} - 1) \\ =& \boldsymbol{d}_{t+\Delta t}\frac{\gamma}{\beta \Delta t} - \boldsymbol{d}_t \frac{\gamma}{\beta \Delta t} + \dot{\boldsymbol{d}}_t (1 - \frac{\gamma}{\beta}) - \ddot{\boldsymbol{d}}_t \Delta t (\frac{\gamma}{2\beta} - 1). \end{aligned} \tag{2.59}$$

After inserting Eq. (2.59) and Eq. (2.58) into the initial equation for the Newmark method (Eq. (2.55)), a nonlinear system of equations is obtained that can be solved for the displacements $\boldsymbol{d}_{t+\Delta t}$ in the next time step:

$$\begin{aligned} \boldsymbol{s}(\boldsymbol{d}_{t+\Delta t}) = \boldsymbol{f}_{t+\Delta t} &- \boldsymbol{M}(\boldsymbol{d}_{t+\Delta t}\frac{1}{\beta \Delta t^2} - \boldsymbol{d}_t \frac{1}{\beta \Delta t^2} - \dot{\boldsymbol{d}}_t \frac{1}{\beta \Delta t} - \ddot{\boldsymbol{d}}_t (\frac{1}{2\beta} - 1)) \\ &- \boldsymbol{C}(\boldsymbol{d}_{t+\Delta t}\frac{\gamma}{\beta \Delta t} - \boldsymbol{d}_t \frac{\gamma}{\beta \Delta t} + \dot{\boldsymbol{d}}_t (1 - \frac{\gamma}{\beta}) - \ddot{\boldsymbol{d}}_t \Delta t (\frac{\gamma}{2\beta} - 1)). \end{aligned} \tag{2.60}$$

Instead of writing the equation of motions with the displacements $\boldsymbol{d}_{t+\Delta t}$ in the next time step as unknown, Eq. (2.56) and Eq. (2.57) could also be inserted into Eq. (2.55) in order to obtain a formulation with the accelerations $\ddot{\boldsymbol{d}}_{t+\Delta t}$ in the next time step as an unknown vector. The resulting nonlinear system then can be written as [27, p. 213]

$$\boldsymbol{s}(\ddot{\boldsymbol{d}}_{t+\Delta t}, \boldsymbol{d}_t, \dot{\boldsymbol{d}}_t, \ddot{\boldsymbol{d}}_t) = \boldsymbol{f}_{t+\Delta t} - \boldsymbol{M}\ddot{\boldsymbol{d}}_{t+\Delta t} - \boldsymbol{C}(\dot{\boldsymbol{d}}_t + \ddot{\boldsymbol{d}}_t(1-\gamma)\Delta t - \ddot{\boldsymbol{d}}_{t+\Delta t}\gamma \Delta t). \tag{2.61}$$

By choosing different parameters γ and β, different time integration methods can be constructed. The most common choice for the parameters is $\gamma = 0.5$ and $\beta = 0.25$. This selection of parameters is called the average acceleration scheme since the accelerations in Eq. (2.56) and Eq. (2.57) are approximated as the average between the current and the next time step [39, p. 125].

The average acceleration scheme has the advantage that it is stable with arbitrarily large time steps for linear problems, as shown by Newmark [40, p. 75]. If the parameters in the Newmark method are set to $\gamma = 0.5$ and $\beta = 0.0$, the Newmark method leads to the central difference method. This means that it is a generalization of the central difference method or, in other words, that the central difference method is a specialization of the Newmark method.

2.5.3 Generalized alpha method

In the generalized alpha method [41], the idea of evaluating the equation of motion (2.47) at different points of time is extended to the range between the current and the next time step. Furthermore, not all expressions are evaluated at the same point of time; the accelerations are evaluated at a different point of time than the other expressions in Eq. (2.47). To achieve this, the coefficients α_m and α_f are introduced [41] as

$$\boldsymbol{M}\ddot{\boldsymbol{d}}_{t+(1-\alpha_m)\Delta t} + \boldsymbol{C}\dot{\boldsymbol{d}}_{t+(1-\alpha_f)\Delta t} + \boldsymbol{s}(\boldsymbol{d}_{t+(1-\alpha_f)\Delta t}) = \boldsymbol{f}_{t+(1-\alpha_f)\Delta t}. \tag{2.62}$$

The accelerations $\ddot{\boldsymbol{d}}_{t+(1-\alpha_m)\Delta t}$, velocities $\dot{\boldsymbol{d}}_{t+(1-\alpha_f)\Delta t}$ and displacements $\boldsymbol{d}_{t+(1-\alpha_f)\Delta t}$ can be interpolated between the current and the next time step in a similar way as, e.g., the accelerations in the Newmark method in Eq. (2.56). Here, $\alpha_m = 1$ and $\alpha_f = 1$ correspond to an evaluation at the current time step and would therefore transform (if both were set to 1) the generalized alpha method into the central difference method. $\alpha_m = 0$ and $\alpha_f = 0$ correspond to an evaluation at the next time step and would therefore transform (if both were set to 0) the generalized alpha method into the Newmark method. Therefore, the generalized alpha method is a generalization of the two other methods. With this extension, the derivation of the method is similar to the Newmark method and therefore not stated in detail here. For the nonlinear formulation of the generalized alpha method, more details can be found, e.g., in [42] or [43].

In the general case, the Newmark method and the generalized alpha method are (in contrast to the central difference method) implicit time integration methods since the result depends not only on the quantities of the current time step but also on the quantities of the next time step. This leads to a system of equations (visible in Eq. (2.61) or Eq. (2.60)) that cannot be solved by direct evaluation. In the nonlinear case in which the vector of internal loads $\boldsymbol{s(d)}$ depends nonlinearly on the displacements, a nonlinear solution algorithm must be used.

2.6 Solution of nonlinear equation systems

In order to solve nonlinear equations, special techniques are necessary. Based on the nonlinear equation system (e. g. Eq. (2.61)) obtained in Sec. 2.5, it is possible to obtain an equation that can be recast to

$$\begin{aligned}\boldsymbol{G}(\boldsymbol{d}_{t+\Delta t}) = \boldsymbol{s}(\boldsymbol{d}_{t+\Delta t}) + \boldsymbol{M}\left(\boldsymbol{d}_{t+\Delta t}\frac{1}{\beta\Delta t^2} - \boldsymbol{d}_t\frac{1}{\beta\Delta t^2} - \dot{\boldsymbol{d}}_t\frac{1}{\beta\Delta t} - \ddot{\boldsymbol{d}}_t(\frac{1}{2\beta} - 1)\right)\\ +\boldsymbol{C}\left(\boldsymbol{d}_{t+\Delta t}\frac{\gamma}{\beta\Delta t} - \boldsymbol{d}_t\frac{\gamma}{\beta\Delta t} + \dot{\boldsymbol{d}}_t(1-\frac{\gamma}{\beta}) - \ddot{\boldsymbol{d}}_t\Delta t(\frac{\gamma}{2\beta} - 1)\right) - \boldsymbol{f}_{t+\Delta t} = \boldsymbol{0}.\end{aligned} \tag{2.63}$$

If a vector $\boldsymbol{d}_{t+\Delta t}$ is found that sets the left-hand side of Eq. (2.63) to zero, the nonlinear equation system is solved. Usually, iterative methods are used to solve nonlinear equation systems. The idea of an iterative method is to start with an initial guess $\boldsymbol{d}^0_{t+\Delta t}$ and get a better approximation with every iteration k of the algorithm. The most famous solution algorithm for nonlinear equation systems is the Newton-Raphson method.

The Newton-Raphson method was originally proposed for scalar equations by Raphson and extended to systems of equations by Simpsons, as detailed in [44].

For the solution of Eq. (2.63), the tangential stiffness matrix $\boldsymbol{K}_T$ must be computed at the location of an intermediate solution vector at the current iteration $\boldsymbol{d}_{t+\Delta t}^k$ as

$$\frac{\partial \boldsymbol{G}(\boldsymbol{d}_{t+\Delta t}^k)}{\partial \boldsymbol{d}_{t+\Delta t}^k} = \boldsymbol{K}_T^k. \tag{2.64}$$

The tangential stiffness matrix $\boldsymbol{K}_T$ is also called the Jacobi matrix, and it contains the derivatives of $\boldsymbol{d}_{t+\Delta t}^k$ as visible in Eq. (2.64). More details about this can be found, e.g., in [27, p. 96 ff.]. In the vicinity of the current solution $\boldsymbol{d}_{t+\Delta t}^k$, Eq. (2.63) can be approximated with the linear expression

$$\boldsymbol{G}(\boldsymbol{d}_{t+\Delta t}) \approx \boldsymbol{G}(\boldsymbol{d}_{t+\Delta t}^k) + \boldsymbol{K}_T^k \Delta \boldsymbol{d}_{t+\Delta t}^k = \boldsymbol{0}. \tag{2.65}$$

After rearranging Eq. (2.65), a *linear* system of equations (2.66) is obtained that delivers a displacement increment $\Delta \boldsymbol{d}_{t+\Delta t}^k$ with which the current solution vector $\boldsymbol{d}_{t+\Delta t}^k$ can be updated:

$$\begin{aligned} \boldsymbol{K}_T^k \Delta \boldsymbol{d}_{t+\Delta t}^k &= -\boldsymbol{G}(\boldsymbol{d}_{t+\Delta t}^k) \\ \boldsymbol{d}_{t+\Delta t}^{k+1} &= \boldsymbol{d}_{t+\Delta t}^k + \Delta \boldsymbol{d}_{t+\Delta t}^k. \end{aligned} \tag{2.66}$$

The benefit of the Newton method is a quadratic rate of convergence in the vicinity of the solution [45, p. 115]. The disadvantage of the Newton method lies in the potentially expensive computation of the tangential stiffness matrix $\boldsymbol{K}_T$ [27, p. 153].

2.7 Post-processing

After completing the FEM simulation and computing the displacements as the primary variable, several additional variables can be derived.

Strains and stresses have so far only been expressed with second-order tensors. However, when accessing the stability of a structure under load, a scalar description of the stress state is beneficial.

The deviatoric part of the stress tensor can be computed as [27, p. 54]

$$\boldsymbol{\sigma}_{dv} = \mathrm{dev}(\boldsymbol{\sigma}) = \boldsymbol{\sigma} - \frac{1}{3}\mathrm{tr}(\boldsymbol{\sigma})\mathbf{I}. \tag{2.67}$$

The point of loading at which a ductile material starts to yield can be computed with the von Mises yield criterion. The von Mises equivalent stress σ_{vM} is computed with the scalar product of two second-order tensors as follows [27, p. 58]

$$\sigma_{vM} = \sqrt{\frac{3}{2}\boldsymbol{\sigma}_{dv} \cdot \boldsymbol{\sigma}_{dv}}. \tag{2.68}$$

For the definition of the scalar product of two second-order tensors see [27, p. 512]. An alternative calculation of the von Mises equivalent stress σ_{vM} is possible with the principle stresses σ_1, σ_2, and σ_3. The principle stresses are the eigenvalues of the stress tensor $\boldsymbol{\sigma}$. The von Mises stress can be computed based on the principle stresses as [28, p. 203]

$$\sigma_{vM} = \sqrt{\frac{1}{2}\left[(\sigma_1 - \sigma_2)^2 + (\sigma_2 - \sigma_3)^2 + (\sigma_3 - \sigma_1)^2\right]}. \tag{2.69}$$

A simple and widely used approach for the safe design of a structure under loading is to maintain the stress level below the critical von Mises stress $\sigma_{vM} < \sigma_{vM,max}$. The critical von Mises stress $\sigma_{vM,max}$ for a certain material depends strongly on the production process and composition. For example, the yield stress of steel typically varies between 200 MPa and 1000 MPa [46, p. 127 ff.].

Similarly to the von Mises equivalent stress, it is possible to define a von Mises equivalent strain ε_{vM} based on the linear engineering strain tensor. The engineering strain tensor can be computed as [27, p. 24]

$$\boldsymbol{\varepsilon} = \frac{1}{2}\left((\boldsymbol{F} - \mathbf{I})^{\mathrm{T}} + (\boldsymbol{F} - \mathbf{I})\right). \tag{2.70}$$

If significant geometrical nonlinearities are present, a different strain measure (for example the Green-Lagrange strain tensor) must be used. Based on the principle strains ε_1, ε_2, and ε_3, which are the eigenvalues of the engineering strain tensor $\boldsymbol{\varepsilon}$, the von Mises equivalent strain is computed as [47, p. 831]

$$\varepsilon_{vM} = \frac{1}{1+\nu}\sqrt{\frac{1}{2}\left[(\varepsilon_1 - \varepsilon_2)^2 + (\varepsilon_2 - \varepsilon_3)^2 + (\varepsilon_3 - \varepsilon_1)^2\right]}. \tag{2.71}$$

2.8 Static condensation

A technique that is beneficial to reduce the number of degrees of freedom (DOF) in a FEM simulation for large linear systems is static condensation.

For the static condensation, the dynamic equation of motion (Eq. (2.47)) is written in its linear form with the constant stiffness matrix $\boldsymbol{K}$ as

$$\boldsymbol{M}\ddot{\boldsymbol{d}}_{\mathrm{d}} + \boldsymbol{C}\dot{\boldsymbol{d}}_{\mathrm{d}} + \boldsymbol{K}\boldsymbol{d}_{\mathrm{d}} = \boldsymbol{f}. \tag{2.72}$$

The displacements which represent the DOF in Eq. (2.72) are now divided into slave nodes and master nodes as

$$\begin{bmatrix} \boldsymbol{M}_{mm} & \boldsymbol{M}_{ms} \\ \boldsymbol{M}_{sm} & \boldsymbol{M}_{ss} \end{bmatrix} \begin{bmatrix} \ddot{\boldsymbol{d}}_m \\ \ddot{\boldsymbol{d}}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{C}_{mm} & \boldsymbol{C}_{ms} \\ \boldsymbol{C}_{sm} & \boldsymbol{C}_{ss} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{d}}_m \\ \dot{\boldsymbol{d}}_s \end{bmatrix} + \begin{bmatrix} \boldsymbol{K}_{mm} & \boldsymbol{K}_{ms} \\ \boldsymbol{K}_{sm} & \boldsymbol{K}_{ss} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}_m \\ \boldsymbol{d}_s \end{bmatrix} = \begin{bmatrix} \boldsymbol{f}_m \\ \boldsymbol{f}_s \end{bmatrix} \tag{2.73}$$

where the index m corresponds to the master nodes and the index s corresponds to the slave nodes. The displacement vector of the master nodes $\boldsymbol{d}_m$ has a length of M, while the displacement vector of the slave nodes $\boldsymbol{d}_s$ has a length of S.

The aim of the static condensation is to reduce the displacement vector $\boldsymbol{d}$, which has a length of $M + S$ to $\boldsymbol{d}_m = \bar{\boldsymbol{d}}$ with a length of M.

To this end, the second row of Eq. (2.73) is solved for the slave displacements $\boldsymbol{d}_s$ while neglecting dynamic effects. The slave displacements can under this assumption be computed as [47, p. 753]

$$\boldsymbol{d}_s = \boldsymbol{K}_{ss}^{-1}\boldsymbol{f}_s - \boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm}\boldsymbol{d}_m. \tag{2.74}$$

If Eq. (2.74) is substituted back into the first row of Eq. (2.73), a reduced system with the number of DOF reduced to a length of M can be computed as

$$\bar{\boldsymbol{M}}\ddot{\bar{\boldsymbol{d}}} + \bar{\boldsymbol{C}}\dot{\bar{\boldsymbol{d}}} + \bar{\boldsymbol{K}}\bar{\boldsymbol{d}} = \bar{\boldsymbol{f}}. \tag{2.75}$$

With the transformation matrix $\boldsymbol{T}$ defined with the identity matrix $\mathbf{I}$ as [48]

$$\boldsymbol{T} = \begin{bmatrix} \mathbf{I} \\ -\boldsymbol{K}_{ss}^{-1}\boldsymbol{K}_{sm} \end{bmatrix}, \tag{2.76}$$

the matrices and the force vector in Eq. (2.75) can be computed as [49]

$$\bar{\boldsymbol{M}} = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{M}\boldsymbol{T}, \quad \bar{\boldsymbol{C}} = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{C}\boldsymbol{T}, \quad \bar{\boldsymbol{K}} = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{K}\boldsymbol{T}, \quad \bar{\boldsymbol{f}} = \boldsymbol{T}^{\mathrm{T}}\boldsymbol{f}. \tag{2.77}$$

In Eq. (2.77), only the approximation of the reduced stiffness matrix $\bar{\boldsymbol{K}}$ is exact in its representation of the full stiffness matrix $\boldsymbol{K}$. The reduced mass matrix $\bar{\boldsymbol{M}}$ and damping matrix $\bar{\boldsymbol{C}}$ are only approximated with respect to Eq. (2.73). Due to this, one can only obtain exact results with static condensation in this formulation for linear static problems. Nonetheless, a sufficiently high number of master nodes will still allow for a reasonable approximation of the reduced mass matrix $\bar{\boldsymbol{M}}$ and damping matrix $\bar{\boldsymbol{C}}$.

One limitation of the reduced system is that boundary conditions can only be applied to the master nodes. This is especially problematic when, e.g., applying distributed loads, which often appear in FSI simulation. Therefore, it is important to have a sufficiently high number of nodes in the regions of highly distributed loads. During post-processing, it is possible to compute the displacements at slave nodes using Eq. (2.74). If all the necessary matrices are stored, this technique allows for an analysis of all quantities such as local stresses for the full system (Eq. (2.73)) in the post-processing.

3 Fluid mechanics

This chapter introduces the basis for the understanding of fluid mechanics and the corresponding methods that are commonly used for the simulation of fluid mechanic problems, namely the finite volume method (FVM) and the boundary element method (BEM). The theory of fluid mechanics and the related methods are explained in many standard textbooks. The main reference used here is [50] for the Navier-Stokes equations and for the FVM. For the potential theory and the BEM, the key reference is [51].

3.1 Basic equations of fluid mechanics

In contrast to the FEM introduced in the previous chapter, in fluid mechanics, it is more common to use the Eulerian description, in which the viewpoint is fixed, while the fluid is moving. The fixed viewpoint in the Eulerian description also implies a fixed domain through which the fluid is moving.

3.1.1 Navier-Stokes equations

The continuity equation and the momentum conservation equation are derived for fluids based on the Eulerian description on the example of an infinitesimal control volume. In combination, the continuity equation and the momentum conservation equation are also referred to as the *Navier-Stokes equations*.

The conservation of mass requires that the change of mass per unit time in an infinitesimal volume element with a fixed reference space (as imposed by the Eulerian description) is equal to the sum of the mass flow over all the boundaries of that volume element. In differential form, this condition can be written as [50, p. 5]

$$\frac{\partial \varrho}{\partial t} + \operatorname{div}(\varrho \boldsymbol{v}) = 0. \tag{3.1}$$

In Eq. (3.1), the mass flow $\varrho \boldsymbol{v}$ is the product of the density ϱ and the velocity vector $\boldsymbol{v}$. If a fluid is incompressible, the density is constant and Eq. (3.1) simplifies to

$$\operatorname{div}(\boldsymbol{v}) = 0. \tag{3.2}$$

The derivation of the momentum conservation equation follows the same idea as the derivation of the linear momentum equation introduced in Sec. 2.1. Accordingly, the basis of the momentum conservation equation for fluids is Newton's second law applied to an infinitesimal volume element with an Eulerian viewpoint.

The momentum equation can be written in differential form as [50, p. 7]

$$\frac{\partial(\varrho \boldsymbol{v})}{\partial t} + \operatorname{grad}(\varrho \boldsymbol{v})\boldsymbol{v} = \operatorname{div}(\boldsymbol{T}) + \varrho \boldsymbol{b}. \tag{3.3}$$

The first term in Eq. (3.3) corresponds to the change of momentum with time, the second term contains the momentum change by convection, and the last term on the right-hand side captures possible body loads $\boldsymbol{b}$ (e.g., gravity loads). The first term on the right-hand side of Eq. (3.3) corresponds to the change of momentum due to pressure and shear stresses. The fluid stress tensor $\boldsymbol{T}$ is stated for Newtonian fluids as [50, p. 6]

$$\boldsymbol{T} = -(p + \frac{2}{3}\mu \operatorname{div}(\boldsymbol{v}))\mathbf{I} + \mu(\operatorname{grad}(\boldsymbol{v}) + \operatorname{grad}(\boldsymbol{v})^{\mathrm{T}}). \tag{3.4}$$

In Eq. (3.4), p is the pressure, μ is the dynamic viscosity, and $\mathbf{I}$ the identity tensor.

To obtain the incompressible momentum equation, Eq. (3.3) is divided by the density ϱ which is assumed to be constant. With this simplification, the momentum equation reads

$$\frac{\partial \boldsymbol{v}}{\partial t} + \operatorname{grad}(\boldsymbol{v})\boldsymbol{v} = \frac{\operatorname{div}(\boldsymbol{T})}{\varrho} + \boldsymbol{b}. \tag{3.5}$$

The dimensionless Mach number Ma can be defined as

$$Ma = \frac{|\boldsymbol{v}|}{c} \tag{3.6}$$

where c is the speed of sound. The assumption of an incompressible flow is not only valid for incompressible fluids, but also valid for flows with a low Mach number $Ma < 0.3$ [50, p. 14].

For FSI simulations, the formulation of the Navier-Stokes equations is slightly modified. Since the interface between the fluid and the structure in an FSI simulation is not fixed but moving, the fluid domain is deforming and a modification to the Eulerian description is necessary. The Arbitrary Lagrangian Eulerian (ALE) description can be used for a moving fluid domain. In the ALE approach, in contrast to the Lagrangian approach, the domain does not follow the motion of the fluid but (in the case of FSI simulations) the motion of the interface between the fluid and the structure [6, p. 31].

Therefore, the fluid mesh is not fixed. Instead, it moves with a mesh velocity $\boldsymbol{v}_m$ which differs from the velocity of the fluid $\boldsymbol{v}$. The convective transport of momentum in the second term on the left-hand side of Eq. (3.5) is changed due to the moving mesh. With this modification, the incompressible momentum equation reads [6, p. 34]

$$\frac{\partial \boldsymbol{v}}{\partial t} + \operatorname{grad}(\boldsymbol{v})(\boldsymbol{v} - \boldsymbol{v}_m) = \frac{\operatorname{div}(\boldsymbol{T})}{\varrho} + \boldsymbol{b}. \tag{3.7}$$

3.1.2 Potential theory

In the potential theory, the fluid is assumed to be inviscid, incompressible, and irrotational [51, p. 44]. With the simplification of an incompressible flow, the density is assumed to be constant, and Eq. (3.2) can be used in the potential theory.

The convective term in Eq. (3.5) can be replaced using the vector identity [51, p. 24]

$$\operatorname{grad}(\boldsymbol{v})\boldsymbol{v} = \frac{\operatorname{grad}(\boldsymbol{v} \cdot \boldsymbol{v})}{2} - \boldsymbol{v} \times \operatorname{curl}(\boldsymbol{v}) \tag{3.8}$$

with the circulation of $\boldsymbol{v}$ defined with the vector cross product ($\times$) as

$$\operatorname{curl}(\boldsymbol{v}) = \nabla \times \boldsymbol{v} = \begin{bmatrix} \frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \\ \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} \\ \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \end{bmatrix}. \tag{3.9}$$

The irrotational property can be expressed mathematically as $\mathrm{curl}(\boldsymbol{v}) = \mathbf{0}$. Therefore, the last term on the right-hand side in Eq. (3.8) is zero for an irrotational flow.

The assumption of an inviscid flow implies that the divergence of the fluid stress tensor $\boldsymbol{T}$ (defined in Eq. (3.4)) in Eq. (3.5) is reduced to $\mathrm{div}(\boldsymbol{T}) = -\mathrm{div}(p\mathbf{I}) = -\mathrm{grad}(p)$. With this simplification, the incompressible momentum equation is also called the Euler equation. By replacing the second term on the left-hand side of Eq. (3.5) with Eq. (3.8), it is possible to state Eq. (3.5) for an inviscid and irrotational flow as

$$\frac{\partial \boldsymbol{v}}{\partial t} + \frac{\mathrm{grad}(\boldsymbol{v}^2)}{2} = -\frac{\mathrm{grad}(p)}{\varrho} + \boldsymbol{b}. \tag{3.10}$$

Since the flow is assumed to be irrotational, a velocity potential $\Phi(x, y, z)$ can be defined [50, p. 16]. The velocity potential is defined by [52, p. 4], [51, p. 27]

$$\boldsymbol{v} = \mathrm{grad}(\Phi(x, y, z)) = \mathrm{grad}(\Phi). \tag{3.11}$$

Eq. (3.11) is now substituted in the first and second term on the left-hand side of Eq. (3.10). If the body load $\boldsymbol{b}$ is the gravity force and z is the direction pointing upwards against the gravitational force, $\boldsymbol{b}$ can be replaced by $-\mathrm{grad}(gz)$ [51, p. 28]

$$\mathrm{grad}\left(\frac{\partial \Phi}{\partial t}\right) + \frac{\mathrm{grad}(\mathrm{grad}(\Phi)^2)}{2} = -\frac{\mathrm{grad}(p)}{\varrho} - \mathrm{grad}(gz). \tag{3.12}$$

Rearranging Eq. (3.12) gives

$$\mathrm{grad}\left(\frac{\partial \Phi}{\partial t} + \frac{\mathrm{grad}(\Phi)^2}{2} + \frac{p}{\varrho} + gz\right) = \mathbf{0}. \tag{3.13}$$

The gradient is zero over the whole domain in Eq. (3.13). This implies that

$$\frac{\partial \Phi}{\partial t} + \frac{\mathrm{grad}(\Phi)^2}{2} + \frac{p}{\varrho} + gz = C(t) \tag{3.14}$$

where C is a constant over the domain Ω and only depends on the time t.

Eq. (3.14) is the Bernoulli equation, indicating the relationship between the velocity potential Φ and the hydrodynamic pressure p. After the velocity field is obtained, the Bernoulli equation can be used to calculate the pressure distribution on a body – which is especially important in the context of FSI simulations.

The continuity equation (Eq. (3.2)) can also be rewritten with the velocity potential (Eq. (3.11)) as

$$\mathrm{div}(\mathrm{grad}(\Phi)) = 0. \tag{3.15}$$

Eq. (3.15) is called the Laplace equation and it is the fundamental equation in the potential theory.

The usual approach in the potential theory is to take advantage of the linearity in the Laplace equation and to superimpose various solutions in order to obtain a realistic flow field for the given problem. The velocity potential can therefore be subdivided into multiple parts that all fulfill the Laplace equation. If the velocity potential is obtained, Eq. (3.14) can be used to compute the pressure field and the corresponding fluid loads.

3.2 Finite volume method for the solution of the Navier-Stokes equations

One of the most common approaches for the numerical solution of the incompressible Navier-Stokes equations (Eqs. (3.2) and (3.5)) is the finite volume method (FVM). The idea of the FVM is to apply the Navier-Stokes equations in the conservative form to finite volume elements. The advantage of this approach is that it is strictly conservative, since the fluxes of mass and momentum that leave an element must be transported to another element (if the mesh is not moving with respect to the initial coordinate system).

Eq. (3.5) can be transformed into the conservative form by taking the volume integral over the control volume. To this end, the second term on the left-hand side of equation (3.5) is rewritten using the following vector calculus rule [27, p. 520]

$$\operatorname{grad}(\boldsymbol{v})\boldsymbol{v} = \operatorname{div}(\boldsymbol{v} \otimes \boldsymbol{v}) - \operatorname{div}(\boldsymbol{v})\boldsymbol{v} = \operatorname{div}(\boldsymbol{v} \otimes \boldsymbol{v}). \tag{3.16}$$

The term $\operatorname{div}(\boldsymbol{v})\boldsymbol{v}$ is zero due to Eq. (3.2). The term on the right-hand side of Eq. (3.16) contains the dyadic product of the velocity vector $\boldsymbol{v}$ with itself. The dyadic product of two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$ can be defined in matrix notation as $\boldsymbol{a} \otimes \boldsymbol{b} = \boldsymbol{a}\,\boldsymbol{b}^T$ [27, p. 27]. With this transformation, the volume integral over Eq. (3.5) can be written as

$$\int_{\Omega,CV} \frac{\partial \boldsymbol{v}}{\partial t}\,\mathrm{d}\Omega + \int_{\Omega,CV} \operatorname{div}(\boldsymbol{v} \otimes \boldsymbol{v})\,\mathrm{d}\Omega = \int_{\Omega,CV} \frac{\operatorname{div}(\boldsymbol{T})}{\varrho}\,\mathrm{d}\Omega + \int_{\Omega,CV} \boldsymbol{b}\,\mathrm{d}\Omega. \tag{3.17}$$

With the divergence theorem, Eq. (3.17) can be transformed to

$$\int_{\Omega,CV} \frac{\partial \boldsymbol{v}}{\partial t}\,\mathrm{d}\Omega + \int_{\Gamma,CV} (\boldsymbol{v} \otimes \boldsymbol{v})\boldsymbol{n}\,\mathrm{d}\Gamma = \int_{\Gamma,CV} \frac{\boldsymbol{T}}{\varrho}\boldsymbol{n}\,\mathrm{d}\Gamma + \int_{\Omega,CV} \boldsymbol{b}\,\mathrm{d}\Omega. \tag{3.18}$$

Similarly, Eq. (3.2) can be rewritten as

$$\int_{\Omega,CV} \operatorname{div}(\boldsymbol{v})\,\mathrm{d}\Omega = \int_{\Gamma,CV} \boldsymbol{v} \cdot \boldsymbol{n}\,\mathrm{d}\Gamma = 0. \tag{3.19}$$

The conservative form of the Navier-Stokes Equations (Eq. (3.18) and (3.19)) can now directly be applied to a volume element in the FVM.

3.2.1 Spatial discretization

The geometry in the FVM is discretized based on volume elements that are connected with faces to each other. The velocity and the pressure are evaluated at the centers of these faces to compute the surface integrals. Since the equations are finally solved for the values at the cell centers of the finite volumes, the values on the face centers have to be interpolated from the cell centers.

The interpolation of the quantities at the face centers has a significant influence on the accuracy and stability of the method. The discretization is illustrated in Fig. 3.1 by means of a one-dimensional example. Here, indices for values at the cell center are capitalized (e.g., v_P is the velocity in the center of the cell P) and indices for values at the faces are written in small letters.

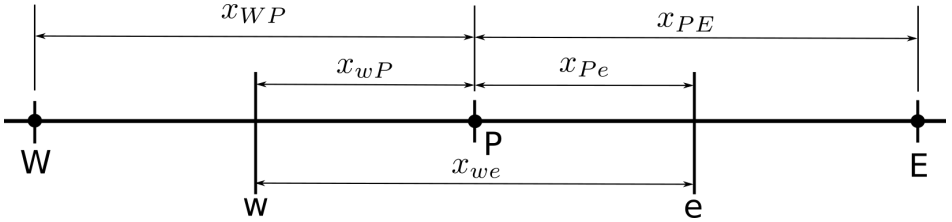


Figure 3.1: One-dimensional grid for the FVM (based on [53, p. 136]).

The surface integral of the convective term in Eq. (3.18) is, for example, discretized for the one-dimensional grid as

$$\int_{\Gamma,CV} (\boldsymbol{v} \otimes \boldsymbol{v})\boldsymbol{n} \, \mathrm{d}\Gamma = A_w v_w^2 - A_e v_e^2. \tag{3.20}$$

To compute the values at the face centers (e.g., v_w in Eq. (3.20)), a common approach is a linear interpolation between the adjacent cell centers. This interpolation can be written for the example introduced in Fig. 3.1 as [50, p. 98]

$$v_w = v_W \frac{x_{wP}}{x_{WP}} + v_P \left(1 - \frac{x_{wP}}{x_{WP}}\right). \tag{3.21}$$

Another common approach is to assume that the velocity at the face center is the same as the velocity in the upstream direction. This approach is also called the upwind method [53, p. 141]. The linear interpolation introduced in Eq. (3.21) is of course more accurate than the upwind method, but the latter produces a significantly more stable simulation [53, p. 141].

Beneficial interpolation methods can also be obtained by combining the idea of the upwind method with polynomial interpolation. The quadratic upstream interpolation for convective kinematics (QUICK) method introduced in [54] uses a second-order polynomial based on the information of one cell center in the downstream direction and two cell centers in the upstream direction to interpolate the value at the face center.

Not only the velocity but also the velocity gradient has to be approximated at the face center. By assuming a constant change of the velocity, the gradient of the velocity (which reduces to a simple derivative in the one-dimensional example introduced in Fig. 3.1) can be computed as [50, p. 99]

$$\mathrm{grad}(v_e) = \left(\frac{\partial v_e}{\partial x}\right) = \frac{v_E - v_P}{x_{PE}}. \tag{3.22}$$

The approximation in Eq. (3.22) is similar to the computation of the time derivative of the displacement in the central difference method in Eq. (2.49).

3.2.2 Temporal discretization

The simplest approximation of the velocity in the next time step $t + \Delta t$ inside the cell center can be found with the *explicit Euler method*. The explicit Euler method is a first-order discretization scheme that can be stated for an incompressible flow as [50, p. 180]

$$\boldsymbol{v}_{t+\Delta t} = \boldsymbol{v}_t + \frac{\partial \boldsymbol{v}_t}{\partial t} \Delta t. \tag{3.23}$$

Similarly to the central difference method, all the quantities that are necessary to compute the velocity at the next time step depend only on the current time step. Therefore, the velocity $\boldsymbol{v}_{t+\Delta t}$ can be computed directly after the change of velocity in the current time step $\frac{\partial \boldsymbol{v}_t}{\partial t}$ has been computed using Eq. (3.18). The simple and fast computation of the velocity in the next time step comes with the disadvantage that the explicit Euler method is unstable for a time-step size larger than the critical time-step size. Definitions for the critical time-step size can be found, e.g., in Ferziger [50, p. 192].

An implicit time discretization method can be constructed by using the derivative of the velocity not in the current time step but in the next time step. In this way, the *implicit Euler method* is deduced – which can be stated as

$$\boldsymbol{v}_{t+\Delta t} = \boldsymbol{v}_t + \frac{\partial \boldsymbol{v}_{t+\Delta t}}{\partial t} \Delta t. \tag{3.24}$$

Compared to the explicit Euler method, the implicit Euler method is stable and, therefore, allows for larger time steps. This comes with higher computational effort due to the implicit formulation that requires an iterative solution procedure, as detailed in the following.

3.2.3 Velocity-pressure coupling

One of the biggest challenges in the numerical solution of the Navier-Stokes equations is the velocity-pressure coupling. This applies especially to implicit time integration schemes and the incompressible Navier-Stokes equations since the pressure does not appear in the incompressible continuity equation that is necessary for a closed equation system.

If the explicit Euler method (Eq. (3.23)) is used and combined with Eq. (3.5), the velocity in the next time step $\boldsymbol{v}_{t+\Delta t}$ can be computed as

$$\boldsymbol{v}_{t+\Delta t} = \boldsymbol{v}_t + \left(\frac{\operatorname{div}(\boldsymbol{T}_t)}{\varrho} + \boldsymbol{b}_t - \operatorname{grad}(\boldsymbol{v}_t)\boldsymbol{v}_t \right) \Delta t. \tag{3.25}$$

Even though the velocity in the next time step $\boldsymbol{v}_{t+\Delta t}$ can be computed with Eq. (3.25), the pressure at a time t step must also be known in order to compute the fluid stress tensor $\boldsymbol{T}$. The equation that can be used to compute the pressure is the continuity Eq. (3.2). The problem is that the incompressible continuity equation does not even depend on the pressure.

In order to solve this issue, the incompressible momentum equation and the continuity equation must be combined. To this end, the divergence of Eq. (3.25) is computed as [50, p. 229]

$$\operatorname{div}(\boldsymbol{v}_{t+\Delta t}) = \operatorname{div}(\boldsymbol{v}_t) + \operatorname{div}\left(\frac{\operatorname{div}(\boldsymbol{T}_t)}{\varrho} + \boldsymbol{b}_t - \operatorname{grad}(\boldsymbol{v}_t)\boldsymbol{v}_t \right) \Delta t. \tag{3.26}$$

If the continuity equation (Eq. (3.2)) is fulfilled at time t and time $t + \Delta t$, the term on the left-hand side and the first term on the right-hand side of Eq. (3.26) must be equal to zero. With this, Eq. (3.26) becomes

$$\operatorname{div}\left(\frac{\operatorname{div}(\boldsymbol{T}_t)}{\varrho} + \boldsymbol{b}_t - \operatorname{grad}(\boldsymbol{v}_t)\boldsymbol{v}_t \right) = 0. \tag{3.27}$$

Now, the pressure term $p_t\mathbf{I}$ must be extracted from the fluid stress tensor $\boldsymbol{T}_t$. For an incompressible fluid, the fluid stress tensor defined in Eq. (3.4) can be simplified to

$$\boldsymbol{T}_t = -p_t\mathbf{I} + \mu(\mathrm{grad}(\boldsymbol{v}_t) + \mathrm{grad}(\boldsymbol{v}_t)^{\mathrm{T}}). \tag{3.28}$$

The fluid stress tensor $\boldsymbol{S}_t$ without the pressure can be defined with this as

$$\boldsymbol{S}_t = \boldsymbol{T}_t + p_t\boldsymbol{I} = \mu\left(\mathrm{grad}(\boldsymbol{v}_t) + \mathrm{grad}(\boldsymbol{v}_t)^{\mathrm{T}}\right). \tag{3.29}$$

With $\mathrm{div}(p_t\mathbf{I}) = \mathrm{grad}(p_t)$ and $\boldsymbol{T}_t = \boldsymbol{S}_t - p_t\mathbf{I}$,

$$\mathrm{div}(\boldsymbol{T}_t) = \mathrm{div}(\boldsymbol{S}_t) - \mathrm{div}(p_t\mathbf{I}) = \mathrm{div}(\boldsymbol{S}_t) - \mathrm{grad}(p_t) \tag{3.30}$$

is obtained. Inserting Eq. (3.30) into Eq. (3.27) and rearranging gives

$$\mathrm{div}\left(\frac{\mathrm{div}(\boldsymbol{S}_t)}{\varrho} + \boldsymbol{b}_t - \mathrm{grad}(\boldsymbol{v}_t)\boldsymbol{v}_t\right) = \frac{\mathrm{div}(\mathrm{grad}(p_t))}{\varrho}. \tag{3.31}$$

This indicates a relationship that can be used to calculate the pressure at an arbitrary time step if the velocity at this time step is known. Combined with Eq. (3.25), the velocity and pressure fields can be computed such that both the momentum and the continuity equation are fulfilled for an incompressible flow with an explicit time integration method.

If an implicit time integration method is used, a coupled system of equations has to be solved simultaneously. For the implicit Euler method defined in Eq. (3.24), the formula for the computation of the velocity in the next time step can, for example, be written as [50, p. 231]

$$\boldsymbol{v}_{t+\Delta t} = \boldsymbol{v}_t + \left(\frac{\mathrm{div}(\boldsymbol{T}_{t+\Delta t})}{\varrho} + \boldsymbol{b}_{t+\Delta t} - \mathrm{grad}(\boldsymbol{v}_{t+\Delta t})\boldsymbol{v}_{t+\Delta t}\right)\Delta t. \tag{3.32}$$

The same procedure as in Eqs. (3.25) to (3.31) can be used to obtain a formula for the computation of the pressure as

$$\mathrm{div}\left(\frac{\mathrm{div}(\boldsymbol{S}_{t+\Delta t})}{\varrho} + \boldsymbol{b}_{t+\Delta t} - \mathrm{grad}(\boldsymbol{v}_{t+\Delta t})\boldsymbol{v}_{t+\Delta t}\right) = \frac{\mathrm{div}(\mathrm{grad}(p_{t+\Delta t}))}{\varrho}. \tag{3.33}$$

Eqs. (3.32) and (3.33) give a set of nonlinear implicit equations that must be solved simultaneously with an iterative solution method. The Newton-Raphson method defined in Eq. (2.66) can be used to this end.

More advanced methods like, e.g., the semi-implicit method for pressure-linked equations (SIMPLE) and variations of this method like the pressure implicit with splitting of operators (PISO) or the PIMPLE algorithm are commonly used for the velocity-pressure coupling in modern FVM solvers. More details about the SIMPLE and PISO algorithms can be found, e.g., in [50, p. 239 ff.] and [53, p. 148 ff.].

3.3 Boundary element method for the computation of potential flows

The introduction to the boundary element method (BEM) presented here follows the book by Katz and Plotkin [51]. The solver *pan*MARE [24], which is used as one of the fluid solvers in this thesis, was developed on the basis of the boundary element method theory.

In order to solve the Laplace equation (Eq. (3.15)) with the BEM, the potential must be restated as an integral over all the boundaries of the flow domain.

To this end, the boundary Γ in a potential flow is divided into three regions as

$$\Gamma = \Gamma_B \cup \Gamma_W \cup \Gamma_\infty. \tag{3.34}$$

In Eq. (3.34), Γ_B refers to the surface of a body inside the domain (usually the FSI surface in the case of maritime FSI simulations), Γ_W refers to a wake model and Γ_∞ to the outer boundary. In the next step, the potential is split into a part caused by a background flow Φ_∞ and a part induced by the presence of a body Φ_{ind} as

$$\Phi = \Phi_\infty + \Phi_{ind}. \tag{3.35}$$

In addition to the Laplace equation, boundary conditions must be satisfied for the potential flow. The first boundary condition states that on the surface of a body Γ_B, the fluid cannot flow through the boundary. This can be expressed as

$$\boldsymbol{v} \cdot \boldsymbol{n} = \operatorname{grad}(\Phi) \cdot \boldsymbol{n} = \boldsymbol{v}_B \cdot \boldsymbol{n} \qquad \text{on} \qquad \Gamma_B. \tag{3.36}$$

In Eq. (3.36), $\boldsymbol{v}_B$ is the velocity of the boundary, and $\boldsymbol{n}$ is the normal direction to the boundary surface. In addition to this, the influence of the potential induced by the presence of a body should decay far away from the body. This second boundary condition can be stated as

$$\lim_{r \to \infty} \operatorname{grad}(\Phi_{\text{ind}}) = 0. \tag{3.37}$$

3.3.1 General formulation

To evaluate the potential at an arbitrary point P inside the fluid domain Ω_F, an integral can be evaluated over the whole boundary and over a sphere Γ_{sp} with a radius of ϵ around the point P. Based on Green's second identity, it can be shown that the integral over the whole boundary and the boundary of the sphere becomes zero. This can be expressed as [51, p. 45]

$$\int_{\Gamma+\Gamma_{sp}} \left(\frac{1}{r} \operatorname{grad}(\Phi) - \Phi \operatorname{grad}\left(\frac{1}{r}\right) \right) \cdot \boldsymbol{n} \, d\Gamma = 0. \tag{3.38}$$

In Eq. (3.38), r refers to the distance to the point P, and Φ is the potential for which the equation should be solved. After splitting the integral into the parts Γ and Γ_{sp} and rearranging, [51, p. 45]

$$\int_{\Gamma_{sp}} \left(\frac{1}{r} \operatorname{grad}(\Phi) - \Phi \operatorname{grad}\left(\frac{1}{r}\right) \right) \cdot \boldsymbol{n} \, d\Gamma = \int_{\Gamma} \left(\frac{1}{r} \operatorname{grad}(\Phi) - \Phi \operatorname{grad}\left(\frac{1}{r}\right) \right) \cdot \boldsymbol{n} \, d\Gamma \tag{3.39}$$

is obtained. As detailed in [51, p. 46], the term on the left-hand side of Eq. (3.39) can be replaced with the limit value consideration $\epsilon \to 0$ by $4\pi\Phi(P)$. With this, the potential at the point P can be found to be

$$\Phi(P) = \frac{1}{4\pi} \int_{\Gamma} \left(\frac{1}{r} \operatorname{grad}(\Phi) - \Phi \operatorname{grad}\left(\frac{1}{r}\right) \right) \cdot \boldsymbol{n} \, d\Gamma. \tag{3.40}$$

In Eq. (3.40), a spherical coordinate system at the point P is used. If Eq. (3.40) is extended by an internal potential Φ_I, which accounts for a flow that is inside the body Γ_B, then

$$\Phi(P) = \frac{1}{4\pi}\int_{\Gamma_B}\left(\frac{1}{r}\operatorname{grad}(\Phi-\Phi_I) - (\Phi-\Phi_I)\operatorname{grad}\left(\frac{1}{r}\right)\right)\cdot\boldsymbol{n}\,d\Gamma + \frac{1}{4\pi}\int_{\Gamma\setminus\Gamma_B}\left(\frac{1}{r}\operatorname{grad}(\Phi) - \Phi\operatorname{grad}\left(\frac{1}{r}\right)\right)\cdot\boldsymbol{n}\,d\Gamma. \tag{3.41}$$

is obtained. From Eq. (3.41), the first term on the right-hand side can be defined as potential induced by a body as [51, p. 47]

$$\Phi_{ind,B}(P) = \int_{\Gamma_B}\left(\frac{1}{r}\operatorname{grad}(\Phi-\Phi_I) - (\Phi-\Phi_I)\operatorname{grad}\left(\frac{1}{r}\right)\right)\cdot\boldsymbol{n}\,d\Gamma. \tag{3.42}$$

The difference of potential between the internal side and the external side of the boundary Γ_B can now be defined with the doublet strength γ [51, p. 47]

$$-\gamma = \Phi - \Phi_I. \tag{3.43}$$

The difference between the derivative of the potential in the normal direction to the boundary Γ_B can be defined with the source strength σ: [51, p. 47]

$$-\sigma = \frac{\partial\Phi}{\partial n} - \frac{\partial\Phi_I}{\partial n}. \tag{3.44}$$

After substituting Eq. (3.43) and Eq. (3.44) into Eq. (3.42), the following relationship can be obtained

$$\Phi_{ind,B}(P) = \int_{\Gamma_B}\left(\frac{\sigma}{4\pi r} + \frac{\gamma}{4\pi}\boldsymbol{n}\cdot\operatorname{grad}\left(\frac{1}{r}\right)\right)\,d\Gamma. \tag{3.45}$$

3.3.2 Elementary solutions

From Eq. (3.45), the potential of a point source can be deduced as [51, p. 49]

$$\Phi_{src} = \frac{\sigma}{4\pi r}. \tag{3.46}$$

Fig. (3.2) shows streamlines and the equipotential lines of a point source. The streamlines are the lines that are parallel to the local velocity vector [51, p. 3]. In the case of a steady flow, the streamlines coincide with the pathlines that follow the motion of a particle through space.

The equipotential lines are the lines along which the potential Φ is constant. Streamlines and equipotential lines are perpendicular to each other at any point of the flow [51, p. 43]. This can also be seen in Fig. 3.2 and Fig. 3.3.

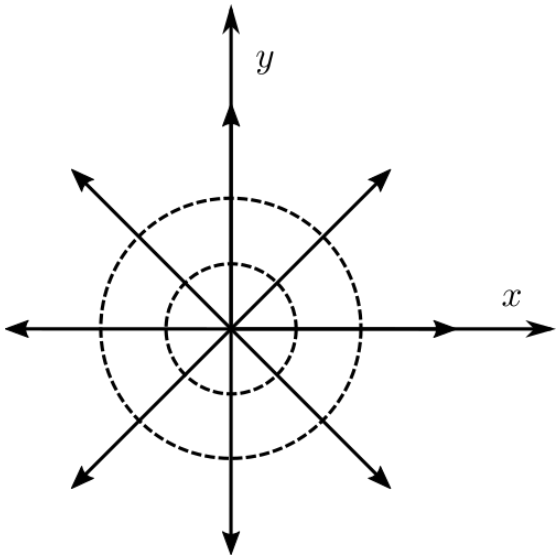


Figure 3.2: Streamlines (continuous) and equipotential (dashed) lines of a source (based on [51, p. 50]).

The potential of a point doublet Φ_{dbl} can be defined similarly to the potential of a point source as [51, p. 51]

$$\Phi_{dbl} = \frac{\gamma}{4\pi} \boldsymbol{n} \cdot \operatorname{grad}\left(\frac{1}{r}\right). \tag{3.47}$$

The corresponding equipotential lines and streamlines are visualized in Fig. 3.3.

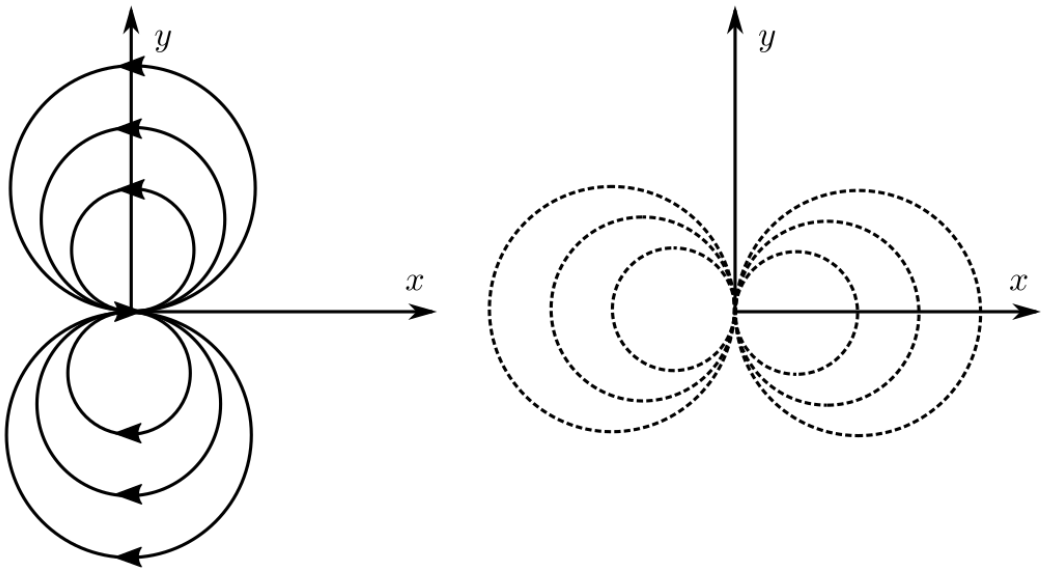


Figure 3.3: Streamlines (continuous) and equipotential (dashed) lines of a doublet (based on [51, p. 58]).

More elementary solutions such as polynomials and (irrotational) vortices are possible. A linear polynomial can be used, for example, to model a uniform background flow as

$$\Phi_{\infty} = \boldsymbol{v}_{\infty} \cdot \boldsymbol{x}. \tag{3.48}$$

The superposition principle can now be used to combine elementary solutions in order to obtain realistic flow fields. The combination of a uniform flow $\Phi_\infty = v_\infty x$ with a doublet as defined by Eq. (3.47) can be used, for example, to obtain a flow around a cylinder that fulfills the boundary condition defined in Eq. (3.36).

Vortex elements are important to capture lifting effects since a body without the presence of vortices, which cause circulation, is not able to generate lift in a potential flow. To this end, the Kutta condition is introduced, which states that "A flow leaves a sharp trailing edge of a body smoothly and with a finite velocity" [51, p. 88]. For the enforcement of the Kutta condition, zero-thickness wake elements are inserted at the sharp trailing edge of a wing. These wake elements are also considered as a boundary to the fluid domain according to Eq. (3.34), and they are modeled using doublet strength distributions. With this, Eq. (3.45) can be split into an integral over a body and an integral over the wake. With the background flow Φ_∞ defined in Eq. (3.48), the full potential according to Eq. (3.35) can be written as

$$\Phi = \int_{\Gamma_B} \left(\frac{\sigma}{4\pi r} + \frac{\gamma}{4\pi} \boldsymbol{n} \cdot \operatorname{grad}\left(\frac{1}{r}\right) \right) d\Gamma + \int_{\Gamma_W} \left(\frac{\gamma}{4\pi} \boldsymbol{n} \cdot \operatorname{grad}\left(\frac{1}{r}\right) \right) d\Gamma + \Phi_\infty. \tag{3.49}$$

3.3.3 Spatial discretization

In order to apply the potential flow theory to a complex geometry, a discretization of the surface geometry must be performed. Since all the integrals in the potential theory are conservative and can be transformed into boundary integrals, the computational effort for the solution of the potential flow can be reduced by one dimension compared to the volumetric integrals that have to be evaluated in the FVM. This makes the potential flow approach in combination with the boundary element approach especially appealing for larger problems – as, for these cases, the conditions for a potential flow as introduced in Sec. 3.1.2 are fulfilled quite well, while the solution with the FVM is prohibitively expensive due to the large number of finite volumes that are necessary to discretize the whole domain.

By combining the Neumann boundary condition defined in Eq. (3.36) with Eq. (3.49), an equation can be found for the computation of the source and doublet strength as [51, p. 208]

$$0 = \operatorname{grad}\left[\int_{\Gamma_B} \left(\frac{\sigma}{4\pi r} + \frac{\gamma}{4\pi} \boldsymbol{n} \cdot \operatorname{grad}\left(\frac{1}{r}\right) \right) d\Gamma + \int_{\Gamma_W} \left(\frac{\gamma}{4\pi} \boldsymbol{n} \cdot \operatorname{grad}\left(\frac{1}{r}\right) \right) d\Gamma + \Phi_\infty \right] \cdot \boldsymbol{n}. \tag{3.50}$$

Here, the coordinate system is assumed to be body-fixed such that $\boldsymbol{v}_b = 0$. The boundary surfaces are now discretized using panels with a constant doublet and source strength over the panel surface. The strength of the source elements in Eq. (3.50) can be directly computed based on the free-stream velocity $\boldsymbol{v}_\infty$ using the Neumann boundary condition as [51, p. 209]

$$\sigma = \boldsymbol{n} \cdot \boldsymbol{v}_\infty. \tag{3.51}$$

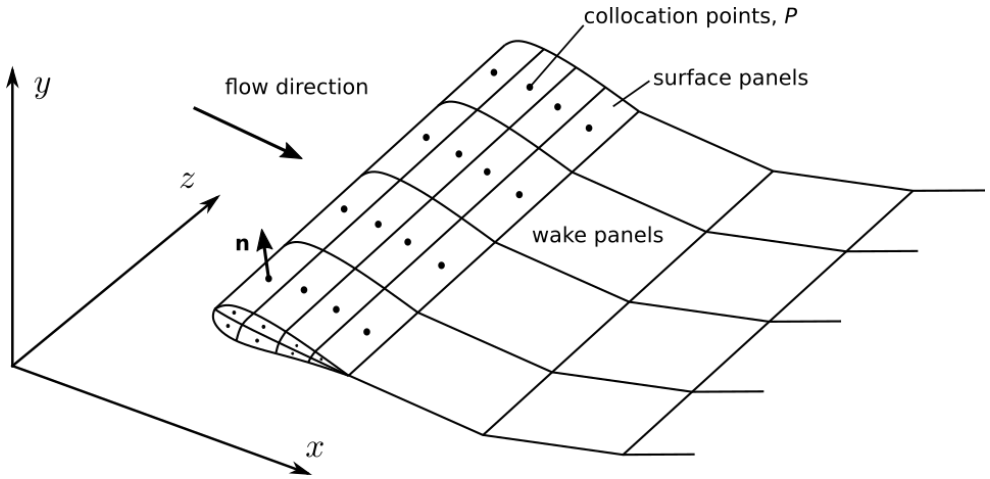


Figure 3.4: Discretization of a wing with panels and collocation points (based on [51, p. 214]).

Fig. 3.4 shows the discretization of the surface of a wing with panels. In the general case, the panels can have the shape of polygons. For each panel k, a collocation point P is defined at which Eq. (3.50) must be fulfilled. The collocation points are usually not located directly at the panel surface to avoid singularity effects. This precaution is necessary because the potential induced by a doublet directly at the location of the doublet approaches infinity, as $\frac{1}{r} \to \infty$ for $r \to 0$ in Eq. (3.47).

In the next step, the influence of all panels on this collocation point is computed. By repeating this procedure for all n collocation points, a set of equations is obtained, as detailed in [51, p. 213 ff.], based on Eq. (3.50). The system of equations can be stated as [51, p. 216]

$$\begin{bmatrix} b_1 \\ \dots \\ b_n \end{bmatrix} = \begin{bmatrix} c_{11} & \dots & c_{1n} \\ \dots & \dots & \dots \\ c_{n1} & \dots & c_{nn} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \dots \\ \gamma_n \end{bmatrix}. \tag{3.52}$$

With the Neumann boundary condition defined in Eq. (3.51), the strength of source elements is known and the remaining unknown variables are the strength of the doublets γ_N. By solving the linear equation system defined in Eq. (3.50), the strength of the doublets γ_N can be computed. In contrast to the FEM, the coefficient matrix in Eq. (3.50) is usually not sparse, so that other linear equation solvers should be used instead of the sparse linear equation solvers which are commonly used in the FEM. After the flow field is computed, the Bernoulli equation is used to compute the pressure at the FSI interface.

The friction force in the tangential direction to the FSI surface, which cannot be computed based on the potential flow theory, is corrected by a friction model. This friction model estimates the friction forces using friction coefficients c_f based on the *Schoenherr line*, see [55]. The combination of the pressure computed by the Bernoulli equation and the friction divided by the area is referred to as the traction $\boldsymbol{t}$. The traction is one of the main coupling quantities in the partitioned solution approach for FSI problems, which will be introduced in the next chapter.

3.4 Propeller parameters

In this section, non-dimensionalized numbers are introduced (based on the dimensional analysis) that are essential for the dimensionless evaluation and comparison of propeller performance.

Dimensionless numbers can be obtained by combining variables that are influential to the flow (e.g. velocity of the fluid, characteristic length scale, or density of the fluid) with multiplication and division operations such that the resulting variable is dimensionless. According to the Pi theorem, two flows are physically similar if these dimensionless parameters have the same values for both flows. All other factors not accounted for by the dimensionless parameters must remain unchanged between the two flows or be deemed not significantly influential. The number of dimensionless parameters n_Π that are necessary for the description of a flow can be computed as

$$n_\Pi = n_{iv} - n_{bd}\,, \tag{3.53}$$

where n_{iv} is the number of influential variables taken into account by the dimensionless numbers and n_{bd} is the number of different basic dimensions (e.g., meter or seconds) used by the influential variables [56, p. 19].

The first dimensionless parameter introduced here is the propeller advance ratio J, defined based on the inflow speed v_∞, the number of revolutions per second f, and the propeller diameter D as

$$J = \frac{v_\infty}{f\,D}. \tag{3.54}$$

The second important parameter is the thrust coefficient k_T. The thrust coefficient k_T can be defined using the propeller thrust T, which is the force that pushes the propeller (and the ship) in the forward direction inside a fluid with density ϱ. With these variables, the thrust, which has the unit $\mathrm{N} = \mathrm{kg}\,\frac{\mathrm{m}}{\mathrm{s}^2}$, can be non-dimensionalized as

$$k_T = \frac{T}{\varrho\, f^2\, D^4}. \tag{3.55}$$

The torque coefficient k_Q is defined in a similar way – with the difference that the thrust is replaced with the torque Q. Since the torque has the unit $\mathrm{Nm} = \mathrm{kg}\,\frac{\mathrm{m}^2}{\mathrm{s}^2}$, a possible non-dimensionalized definition of the torque coefficient k_Q is

$$k_Q = \frac{Q}{\varrho\, f^2\, D^5}. \tag{3.56}$$

According to Eq. (3.53), the six influential variables ($v_\infty, f, D, T, Q, \varrho \rightarrow n_{iv} = 6$) with the three basic dimensions ($\mathrm{m}, \mathrm{s}, \mathrm{kg} \rightarrow n_{bd} = 3$) can be captured with the three dimensionless variables above ($J, k_t, k_Q \rightarrow n_\Pi = 3$).

With the definition of the thrust and torque coefficient, the efficiency of the propeller η_0 can be computed as

$$\eta_0 = \frac{J}{2\pi}\frac{k_T}{k_Q} = \frac{v_\infty}{2\pi\, f}\frac{T}{Q}\,. \tag{3.57}$$

As a measure for the area A_{cav} on the propeller on which cavitation occurs, the dimensionless cavitation area can be computed using the propeller surface area A_{prop} as

$$k_{cav} = \frac{A_{cav}}{A_{prop}}. \tag{3.58}$$

Non-dimensionalized numbers, such as the propeller parameters mentioned above, are especially important when comparing different propellers in terms of thrust, efficiency, and other performance measures such as the cavitation area. If the aim is to define a propeller shape for a large ship based on experimental tests, the dimensionless propeller parameters can be used to set up a model test that can be conducted with a scaled-down version of the propeller in a small tank – enabling affordable experimental tests for larger ship propellers as well.

4 Fluid-structure interaction

The field of fluid-structure interaction (FSI) concerns the mutual dependency of a fluid and a structure interacting with each other. For FSI problems, different approaches for the solution can be chosen.

A one-way coupling approach is commonly used when the effect of one of the two fields on the other remains small enough to be neglected. For instance, this might be the case for a structure with a strong foundation that experiences very little movement under the influence of fluid loads. In this case, only the effect of the fluid on the structure has to be considered in the structural design, and a one-way coupling approach is advisable. If the loads introduced by the fluid field deform or displace the structure so much that the fluid field is strongly influenced by the deformation of the fluid domain, a two-way coupling approach should be used.

There are two different solution strategies for two-way coupling approaches to FSI problems: In the monolithic coupling approach, all fields are solved simultaneously with a general solver that can solve both the fluid and the structural field. In the partitioned or staggered approach, the fluid and the structural fields are solved sequentially, and only a small set of quantities is exchanged at the boundary of the two fields [6].

Even though the monolithic coupling approach is generally more stable than the partitioned approach, the partitioned coupling approach can be stabilized with extrapolation and convergence acceleration techniques, as will be detailed later. The advantage of the partitioned approach over the monolithic approach is that existing advanced, fast, and highly specialized solvers can be used for both fields, and modifications to the existing solvers can be kept to a minimum. Furthermore, the resulting equation systems in both solvers are smaller compared to the monolithic solution of one large system.

4.1 Partitioned solution approach

In the partitioned solution, the fluid and the structural fields are solved completely separately, which is why we can introduce an operation for the fluid solver as

$$\boldsymbol{t}_i^t = \mathcal{F}(\tilde{\boldsymbol{d}}_i^t). \tag{4.1}$$

In Eq. (4.1), only the discretized displacement $\tilde{\boldsymbol{d}}_i^t$ on the FSI interface is considered as the input parameter. Here, the index i refers to the implicit iteration and the variable t to the time step. The fluid domain (and with it the mesh) must deform due to the displacements that act on the interface. Due to the deformation of the fluid domain, the velocity and pressure fields are changed inside the fluid domain. With the velocity and pressure field of the fluid, a traction $\boldsymbol{t}_i^t$ can be computed at the FSI interface.

The term traction here refers to the force per area that acts on the fluid boundaries. It combines pressure (which is normal to the surface) with the tangential stresses due to the

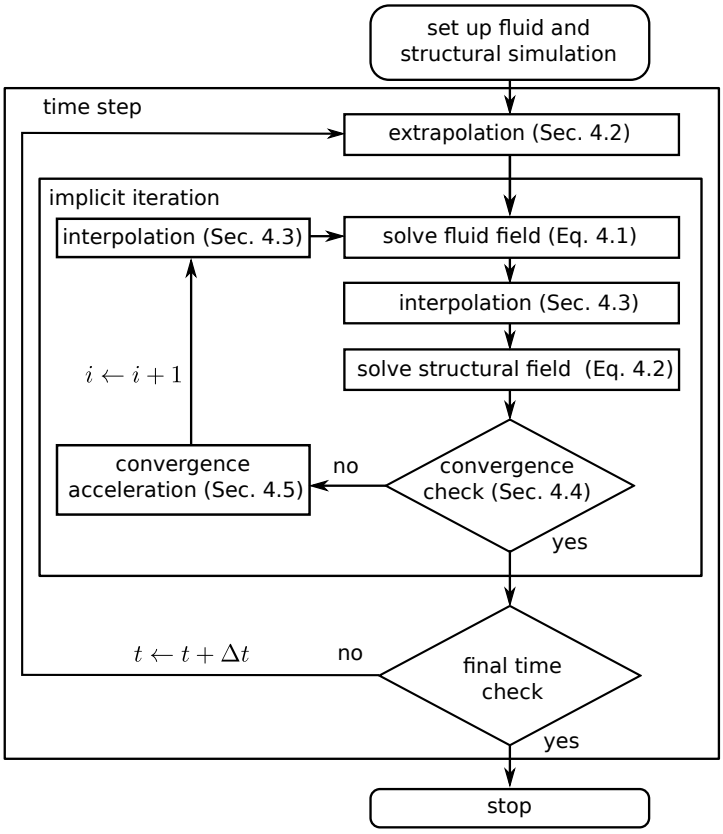


Figure 4.1: Partitioned coupling solution approach for FSI problems

velocity gradient and the friction effects at the boundary. The computation of the tractions in the fluid solver $\mathcal{F}$ was already described in Chapter 3.

The tractions on the FSI interface are now considered as the input parameter to the structural solver $\mathcal{S}$ as

$$\boldsymbol{d}_i^t = \mathcal{S}(\boldsymbol{t}_i^t). \tag{4.2}$$

The computation of the displacements in the structural solver $\mathcal{S}$ is described in Chapter 2. The output of the structural solver are new displacements $\boldsymbol{d}_i^t$ that differ from the displacement $\tilde{\boldsymbol{d}}_i^t$ which are the input to the fluid solver in Eq. (4.1).

If Eq. (4.1) and (4.2) are combined, the fixed point iteration

$$\boldsymbol{d}_i^t = \mathcal{S}(\mathcal{F}(\tilde{\boldsymbol{d}}_i^t)) \tag{4.3}$$

is obtained. The displacements $\boldsymbol{d}_i^t$, which are computed by the structural solver, could directly serve as the displacements $\tilde{\boldsymbol{d}}_{i+1}^t$, which are the input to the fluid solver in the next iteration. This would be the unmodified fixed-point iteration, which could be repeated until convergence is obtained. Since the unmodified fixed-point iteration is often unstable, it is stabilized and accelerated in the partitioned solution approach with extrapolation and convergence acceleration techniques, as shown in Fig. 4.1.

4.2 Extrapolation methods

At the beginning of every time step, an extrapolation of the displacements in the current time step $\tilde{\boldsymbol{d}}_1^t$ is conducted based on the converged displacements $\boldsymbol{d}^{t-\Delta t}, ..., \boldsymbol{d}^{t-m\Delta t}$ of the m previous time steps. Written with the extrapolation operation $\mathcal{E}$, this can be stated as

$$\tilde{\boldsymbol{d}}_1^t = \mathcal{E}(\boldsymbol{d}^{t-\Delta t}, ..., \boldsymbol{d}^{t-m\Delta t}). \tag{4.4}$$

For the extrapolation in the first time step, no previous data are known and the displacements in the first time step must be initialized as $\tilde{\boldsymbol{d}}_1^1 = \mathbf{0}$, where $\mathbf{0}$ is a vector filled with zeros. If a sufficient number of converged displacements from the previous time steps are available, a polynomial extrapolation can be constructed as

$$d^{t+j\Delta t} = \sum_{i=0}^{p} \alpha_i \, t_j^i \qquad \text{for } j = -1, ..., -(p+1). \tag{4.5}$$

Here, all three spatial directions are treated independently and, thus, the equation is scalar-valued. If Eq. (4.5) is solved for a polynomial of order p, coefficients a_i can be computed by solving the resulting system of equations. If the constructed polynomial is then evaluated at the current time step t, a formula is obtained with which the previous converged displacements $\boldsymbol{d}^{t-\Delta t}, ..., \boldsymbol{d}^{t-m\Delta t}$ can be extrapolated to the current time step. Tab. 4.1 lists the coefficients for this procedure for a polynomial of order 0, 1, and 2 and for the assumption of a constant time-step size. Higher order polynomials usually do not lead to a better approximation, which is due to Runge's phenomenon [57].

Table 4.1: Equations for polynomial extrapolation.

order p	**equation**
0	$\tilde{\boldsymbol{d}}_1^t = \boldsymbol{d}^{t-\Delta t}$
1	$\tilde{\boldsymbol{d}}_1^t = 2\boldsymbol{d}^{t-\Delta t} - \boldsymbol{d}^{t-2\Delta t}$
2	$\tilde{\boldsymbol{d}}_1^t = 3\boldsymbol{d}^{t-\Delta t} - 3\boldsymbol{d}^{t-2\Delta t} + \boldsymbol{d}^{t-3\Delta t}$

Besides polynomial extrapolation, there are also other extrapolation methods, such as extrapolation based on a Taylor-series. For the Taylor-series expansion, not only the previous displacements are taken into account, but also the derivatives of the displacements. Therefore, additional information is required from the structural solver.

4.3 Interpolation methods

The discretization of the fluid and the structural field on the FSI surface are generally nonmatching, which means that the mesh nodes of the fluid and structure are at different locations.

For nonmatching meshes, interpolation methods must be used to map the displacements from the structural mesh to the fluid mesh. For the displacements, the structural mesh is the source mesh, and the fluid mesh is the target mesh. A typical combination of a

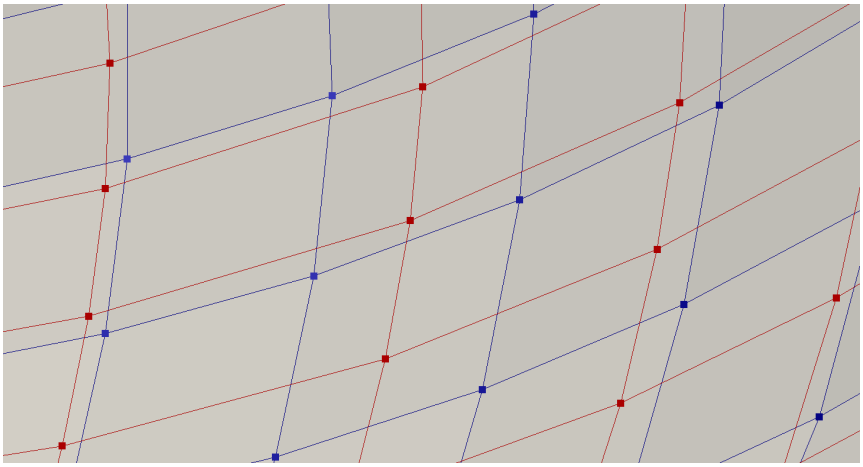


Figure 4.2: Mapping from source mesh (in red) to target mesh (in blue).

source and a target mesh is shown in Fig. 4.2. The interpolation can be expressed with the operator $\mathcal{I}$ as

$$\boldsymbol{d}_f = \mathcal{I}(\boldsymbol{d}_s, \boldsymbol{x}_s, \boldsymbol{x}_f). \tag{4.6}$$

The vector $\boldsymbol{x}_s$ contains the source points for the interpolation. In the case of Eq. (4.6), the location of the structural nodes is stored in $\boldsymbol{x}_s$. The target points for the interpolation are in this case the fluid nodes $\boldsymbol{x}_f$.

Similarly, the tractions must be mapped from the fluid mesh to the structural mesh as

$$\boldsymbol{t}_s = \mathcal{I}(\boldsymbol{t}_f, \boldsymbol{x}_f, \boldsymbol{x}_s). \tag{4.7}$$

The tractions are usually computed at the face centers of the fluid mesh $\boldsymbol{x}_f$ and must be mapped to Gaussian integration points on the surface of the structural mesh $\boldsymbol{x}_s$. In Eqs. (4.6) and (4.7), the interpolation is only considered as an interpolation between point clouds. The connectivity at the mesh is not considered in these two equations, so that these equations only capture the mesh-independent interpolation. Mesh-based interpolation will be introduced later.

If the mesh topology remains unchanged during the simulation, a constant interpolation matrix $\boldsymbol{I}_m$ with the size $N \times M$ can be introduced for the mapping from a set of source points $\boldsymbol{x}_s$ of length M to a set of target points $\boldsymbol{x}_t$ of length N. With this matrix, an arbitrary quantity $\boldsymbol{q}$ can be mapped from the source points to the target points as

$$\boldsymbol{q}_t = \boldsymbol{I}_m \boldsymbol{q}_s. \tag{4.8}$$

Interpolation methods can be divided into conservative and consistent interpolation methods.

A conservative interpolation method keeps the sum over all the entries of the quantity constant as $\sum_i^m \boldsymbol{q}_{s,i} = \sum_{j=1}^n \boldsymbol{q}_{t,j}$. For a conservative mapping, the sum over all the columns of the matrix $\boldsymbol{I}_m$ in Eq. 4.8 is equal to one. With this property, conservative interpolation methods are appropriate for extensive quantities such as force and energy [58, p. 72].

A consistent interpolation method keeps a quantity at a point unchanged if the corresponding source and target point have the same coordinates, and it serves to exactly interpolate a quantity that is constant over the entire mesh [59]. For a consistent mapping,

the sum over all the rows of the matrix $\boldsymbol{I}_m$ in Eq. (4.8) is equal to one. Therefore, consistent interpolation methods are appropriate if intensive variables are considered. Since the tractions and the displacements are both intensive variables, a consistent interpolation method is appropriate for the mapping introduced in Eqs. (4.6) and (4.7).

4.3.1 Nearest neighbor interpolation

The simplest approach for the mapping between two meshes is the nearest neighbor interpolation. The consistent version of the nearest neighbor interpolation assigns to each point on the target mesh the point of the source mesh for which the distance is minimal. The Euclidean norm of a vector with the size n can be defined as

$$\|\boldsymbol{a}\|_2 = \sqrt{\sum_{i=1}^{n} a_i^2}. \tag{4.9}$$

The distance between two points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be computed using the Euclidean norm as

$$d_{ij} = ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2. \tag{4.10}$$

For the computation of the interpolation matrix $\boldsymbol{I}_m$, all the $n \times m$ distances have to be computed in order to identify the minimum distances between the source points and the target points.

The nearest neighbor interpolation can be designed either as a consistent mapping or as a conservative mapping [58, p. 76]. If to each point of the target mesh one point of the source mesh is assigned, the mapping is consistent. In that case, only one entry in every row of the interpolation matrix $\boldsymbol{I}_m$ is one and all other entries are zero. If, on the other hand, to each point of the source mesh one point of the target mesh is assigned, the mapping is conservative. In this case, only one entry in every column of the interpolation matrix $\boldsymbol{I}_m$ is one, and all other entries in the corresponding column are zero.

4.3.2 Barycentric interpolation

A more sophisticated mapping approach is the barycentric interpolation method. If the barycentric interpolation is applied to the mapping of surfaces, the three nearest neighbors from the source mesh to a point on the target mesh are determined for mapping in a two-dimensional space.

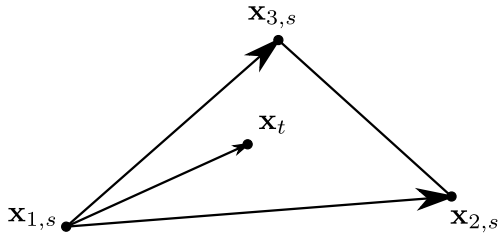


Figure 4.3: Barycentric interpolation for surface mapping in a two-dimensional space.

These three points span a triangle as shown in Fig. 4.3. The point $\boldsymbol{x}_t$ can be expressed using the barycentric coordinates as

$$\boldsymbol{x}_t = \boldsymbol{x}_{1,s} + \alpha_1(\boldsymbol{x}_{2,s} - \boldsymbol{x}_{1,s}) + \alpha_2(\boldsymbol{x}_{3,s} - \boldsymbol{x}_{1,s}). \tag{4.11}$$

The vectors $(\boldsymbol{x}_{2,s} - \boldsymbol{x}_{1,s})$ and $(\boldsymbol{x}_{3,s} - \boldsymbol{x}_{1,s})$ point from the first point $\boldsymbol{x}_{1,s}$ of the triangle to $\boldsymbol{x}_{2,s}$ and $\boldsymbol{x}_{3,s}$, as shown in Fig. 4.3. The parameters α_1 and α_2 should be determined in such a way that a combination of these two vectors is obtained that points $\boldsymbol{x}_{1,s}$ to $\boldsymbol{x}_t$. The correct combination can be obtained by rearranging and solving Eq. (4.11). The resulting linear equation system can be used to compute the barycentric coordinates α_1 and α_2 in a two-dimensional space. Cramer's rule can, for example, be used to solve the obtained equation system [60, p. 47].

For the three-dimensional space, a fourth point is necessary to determine the third independent coordinate α_3 with a three-dimensional linear equation system as

$$\boldsymbol{x}_t = \boldsymbol{x}_{1,s} + \alpha_1(\boldsymbol{x}_{2,s} - \boldsymbol{x}_{1,s}) + \alpha_2(\boldsymbol{x}_{3,s} - \boldsymbol{x}_{1,s}) + \alpha_3(\boldsymbol{x}_{4,s} - \boldsymbol{x}_{1,s}) \tag{4.12}$$

The four points in Eq. (4.12) span a tetrahedron.

After computing the barycentric coordinates, the mapping can now be performed with the barycentric coordinates in the two-dimensional space as

$$\boldsymbol{q}_t = \boldsymbol{q}_{1,s} + \alpha_1(\boldsymbol{q}_{2,s} - \boldsymbol{q}_{1,s}) + \alpha_2(\boldsymbol{q}_{3,s} - \boldsymbol{q}_{1,s}) \tag{4.13}$$

and in the three-dimensional space as

$$\boldsymbol{q}_t = \boldsymbol{q}_{1,s} + \alpha_1(\boldsymbol{q}_{2,s} - \boldsymbol{q}_{1,s}) + \alpha_2(\boldsymbol{q}_{3,s} - \boldsymbol{q}_{1,s}) + \alpha_3(\boldsymbol{q}_{4,s} - \boldsymbol{q}_{1,s}). \tag{4.14}$$

If the three points are located on a line for Eq. (4.11) or the four points are located on a plane for Eq. (4.12), the next nearest neighbor can be used to obtain a set of linearly independent vectors $(\boldsymbol{x}_{i,s} - \boldsymbol{x}_{1,s})$. The algorithm that implements this approach efficiently can be found in [60, p. 47 f].

4.3.3 Inverse distance weighting

Inverse distance weighting, introduced by Shepard in [61, p. 518], is another sophisticated interpolation method. The formula for the interpolation based on inverse distance weighting is

$$\boldsymbol{q}_t = \frac{\sum_{i=1}^{N} a_i \, \boldsymbol{q}_{i,s}}{\sum_{i=1}^{N} a_i}. \tag{4.15}$$

In Eq. (4.15), the weights α_i are computed according to

$$\alpha_i = \frac{1}{||\boldsymbol{x}_t - \boldsymbol{x}_{i,s}||_2}. \tag{4.16}$$

If the distance between the target point and one of the source points $\boldsymbol{x}_{i,s}$ is zero, the value $\boldsymbol{q}_t$ is set to the value $\boldsymbol{q}_{i,s}$. To reduce the computational effort, the number of source points is usually limited to the k nearest neighbors. Alternatively, it is possible to use only the points that are within a certain radius around the target point.

4.3.4 Mesh-based interpolation

The idea of mesh-based interpolation is not only to use the information of the values at the point locations but also to take into account the mesh topology. The mesh-independent interpolation can lead to problems – for example, if the tractions are interpolated at sharp edges. The traction vector is usually changed abruptly at edges due to the changed normal vector of the surface at these locations. Nonetheless, points that are located on the other side of the sharp edges can be taken into account and, thus, have a strong impact on the interpolation.

If the FEM is used on the structural side, the shape functions defined in Eq. (2.23) inherently define the solution on the whole structural domain and can therefore be used to obtain the displacements at an arbitrary location in the domain. If the location to which the displacements have to be mapped lies outside of the structural domain, a projection of this location to the structural mesh must be performed.

To interpolate the tractions to the structural mesh, the tractions must first be computed at locations on the boundary of the fluid domain. Subsequently, they have to be projected to the boundary of the structural domain. For the projection, a so-called axis-aligned bounding box (AABB)-tree is used as a basis for the search algorithm to find the elements closest to the given target point $\boldsymbol{x}_t$, as described in [62, p. 69 f][63]. In the next step, a Newton-Raphson procedure is applied for all candidates to find the minimum distance of the target point to the corresponding element. The element with the lowest distance is evaluated at the location where the distance is minimized. More details about the projection can be found in [64, p. 139].

After the target point has been projected onto the source mesh, the interpolation depends on the type of the source mesh. If the FEM is used, the displacement is evaluated inside or on the boundary of a finite element. In this case, the shape functions inherently define the solution inside the element. After conducting the inverse mapping, the interpolated displacements can be computed directly by inserting the obtained local coordinates into the discrete FEM approximation defined in Eq. (2.26).

If the FVM mesh or BEM mesh is used as the source mesh for the interpolation of the tractions to the structural integration points, a different procedure must be applied. Similar to the previously described approach, the faces or panels closest to the target point are identified. After identifying the closest panel (in the case of a BEM mesh), an inverse distance weighting is performed based on the traction values of the panel center and the panel centers of all neighboring panels which have a normal close to the normal of the closest panel.

4.4 Convergence criteria

If the fluid field is solved before the structural field (as visualized in the procedure shown in Fig. 4.1), the residual $\boldsymbol{r}_i$ in the iteration i is usually computed based on the displacement that entered the fluid solver $\tilde{\boldsymbol{d}}_i^t$ and the displacements that are computed by the structural solver $\boldsymbol{d}_i^t$ as

$$\boldsymbol{r}_i = \boldsymbol{d}_i^t - \tilde{\boldsymbol{d}}_i^t. \tag{4.17}$$

Two different convergence criteria are checked based on the Euclidean norm of the residual $\|\boldsymbol{r}_i\|_2$. The absolute convergence criterion is fulfilled if a predefined absolute tolerance ϵ_a

is reached by the residual as

$$\|\boldsymbol{r}_i\|_2 < \epsilon_a. \tag{4.18}$$

The relative convergence criterion is fulfilled if the quotient of the current residual r_i to the residual in the first iteration in the current time step r_1 is below a predefined relative tolerance ϵ_r. This can be expressed as

$$\frac{\|\boldsymbol{r}_i\|_2}{\|\boldsymbol{r}_1\|_2} < \epsilon_r. \tag{4.19}$$

The current time step is usually assumed to be converged if one of the two convergence criteria is fulfilled.

If the time step is converged, the next time step is initialized with the extrapolation of the displacements, as explained in Sec. 4.2. If the time step is not converged, a convergence acceleration method is applied before the next implicit iteration $i + 1$ is entered.

4.5 Convergence acceleration methods

If the convergence criteria are not fulfilled in an iteration, a convergence acceleration method $\mathcal{A}$ is usually applied. The idea of a convergence acceleration method is to stabilize and accelerate the convergence of the displacements at the interface based on the information from all the previous iterations in a time step. This can be expressed with the convergence acceleration operator $\mathcal{A}$ as

$$\tilde{\boldsymbol{d}}_{i+1}^t = \mathcal{A}(\boldsymbol{d}_i^t, \tilde{\boldsymbol{d}}_i^t, \boldsymbol{d}_{i-1}^t, ..., \boldsymbol{d}_{i-n}^t). \tag{4.20}$$

For some methods, such as the multi-time-step quasi-Newton least squares method, it is also possible to use information from previous time steps.

4.5.1 Static relaxation

The simplest convergence acceleration technique available is static relaxation. For the static relaxation method, the residual $\boldsymbol{r}_i$ according to Eq. (4.17) is multiplied with a constant relaxation factor ω before it is added to the current displacements $\tilde{\boldsymbol{d}}_i^t$ as

$$\tilde{\boldsymbol{d}}_{i+1}^t = \mathcal{A}(\boldsymbol{d}_i^t, \tilde{\boldsymbol{d}}_i^t) = \tilde{\boldsymbol{d}}_i^t + \omega(\boldsymbol{d}_i^t - \tilde{\boldsymbol{d}}_i^t) = \tilde{\boldsymbol{d}}_i^t + \omega \boldsymbol{r}_i. \tag{4.21}$$

At the beginning of the simulation, the relaxation factor ω is set to a constant value – usually between 0 and 1. The optimal choice for an appropriate relaxation factor is nontrivial and strongly dependent on the problem. If the relaxation factor is set closer to 0, the stabilizing effect can be increased at the expense of a possibly slower convergence.

4.5.2 Aitken method

In contrast to the previous method, the Aitken method computes the relaxation factor ω dynamically depending on the displacements of the three last iterations $\boldsymbol{d}_i^t, \boldsymbol{d}_{i-1}^t$ and $\boldsymbol{d}_{i-2}^t$. For the Aitken method, many different extensions of the scalar-valued version to multidimensional problems exist, as detailed, e.g., in [65]. Three different approaches should be discussed here.

Aitken 1

Since the displacement vector $\boldsymbol{d}$ has a constant size n, the vector sequence can be treated as n independent scalar sequences for each entry $\boldsymbol{d}_j$ of the displacement vector.

The formula to compute the initial displacement for the next iteration with this version of the Aitken method then reads [66, p. 57] [65, p. 385]

$$\tilde{d}^t_{j,i+1,k} = d^t_{j,i+1,k} - \frac{(d^t_{j,i+1,k} - d^t_{j,i,k})^2}{d^t_{j,i+1,k} - 2\,d^t_{j,i,k} - d^t_{j,i-1,k}} \qquad \text{for } j = 1, ..., n \text{ and } k = x, y, z. \quad (4.22)$$

Since the displacement vector is a three-dimensional vector for a three-dimensional problem, the index k (referring to the x-, y-, or z-direction) is introduced in Eq. (4.22).

Aitken 2

A more sophisticated approach can be reached by computing a dynamic relaxation factor ω_i as the solution of the minimization problem $\min(\|\tilde{\boldsymbol{d}}^t_i - \tilde{\boldsymbol{d}}^t_{i+1}\|_2)$ for the system

$$\tilde{\boldsymbol{d}}^t_i = \boldsymbol{d}^t_{i-1} + \omega_i(\boldsymbol{d}^t_{i-2} - \boldsymbol{d}^t_{i-1})$$
$$\tilde{\boldsymbol{d}}^t_{i+1} = \boldsymbol{d}^t_i + \omega_i(\boldsymbol{d}^t_{i-1} - \boldsymbol{d}^t_i).$$

Solving this minimization gives the formula to compute the dynamic relaxation factor ω_i as [65, p. 386]

$$\omega_i = -\frac{\hat{\boldsymbol{r}}_{i-1} \cdot (\hat{\boldsymbol{r}}_i - \hat{\boldsymbol{r}}_{i-1})}{\|\hat{\boldsymbol{r}}_{i-1} - \hat{\boldsymbol{r}}_i\|_2^2}. \quad (4.23)$$

In Eq. (4.23), the residual $\hat{\boldsymbol{r}}_i$ is defined as $\hat{\boldsymbol{r}}_i = \boldsymbol{d}^t_i - \boldsymbol{d}^t_{i-1}$. Since the displacements $\boldsymbol{d}^t_i, \boldsymbol{d}^t_{i-1}$, and $\boldsymbol{d}^t_{i-2}$ are necessary to compute a new relaxation factor, the dynamic relaxation factor ω_i can only be computed in every third iteration.

The accelerated displacement can then be computed similarly to Eq. (4.20) as

$$\tilde{\boldsymbol{d}}^t_{i+1} = \tilde{\boldsymbol{d}}^t_i + \omega_i \boldsymbol{r}_i. \quad (4.24)$$

For the intermediate iterations, the relaxation factor is set to 1.

Aitken 3

A modification of the method introduced in Eq. (4.23) was proposed by Irons and Tuck in [67]. This method modifies Eq. (4.23) to compute a new dynamic relaxation factor ω_i in every iteration. The updated formula for the relaxation factor ω_i is [26, p. 1771]

$$\omega_i = -\omega_{i-1}\frac{\hat{\boldsymbol{r}}_{i-1} \cdot (\hat{\boldsymbol{r}}_i - \hat{\boldsymbol{r}}_{i-1})}{\|\hat{\boldsymbol{r}}_{i-1} - \hat{\boldsymbol{r}}_i\|_2^2}.$$

In the first iteration, the initial relaxation factor ω_0 is set to 0.5.

4.5.3 Quasi-Newton methods

Another possible approach to achieve convergence acceleration in the iterative solution process is the Newton-Raphson method, as introduced in Sec. 2.6. To apply the Newton-Raphson method, Eq. (4.3) is reformulated in such a way that the right-hand side becomes zero. The function $\boldsymbol{G}$, for which the roots have to be found, can be written as

$$\boldsymbol{G}(\tilde{\boldsymbol{d}}_i^t) = \boldsymbol{r}_i^t = \mathcal{S}(\mathcal{F}(\tilde{\boldsymbol{d}}_i^t)) - \tilde{\boldsymbol{d}}_i^t = \boldsymbol{0} \tag{4.25}$$

With $\boldsymbol{G}$, the Jacobian matrix can now be expressed as

$$\boldsymbol{J}_i^t = \frac{\partial \boldsymbol{G}(\tilde{\boldsymbol{d}}_i^t)}{\partial \tilde{\boldsymbol{d}}_i^t}. \tag{4.26}$$

According to the Newton-Raphson method, the iterative formula to compute the improved displacements $\tilde{\boldsymbol{d}}_{i+1}^t$ is

$$\begin{aligned} \boldsymbol{J}_i^t \Delta\tilde{\boldsymbol{d}}_{i,i+1}^t &= -\boldsymbol{G}(\tilde{\boldsymbol{d}}_i^t) \\ \tilde{\boldsymbol{d}}_{i+1}^t &= \tilde{\boldsymbol{d}}_i^t + \Delta\tilde{\boldsymbol{d}}_{i,i+1}^t. \end{aligned} \tag{4.27}$$

The idea of quasi-Newton methods compared to the Newton-Raphson method is to approximate the Jacobian matrix (or its inverse) instead of computing the full Jacobian matrix – which can be very expensive, especially when done with difference quotients [68].

Broyden method

The first method of the class of the quasi-Newton methods is the Broyden method [69]. The Broyden method belongs to the class of secant methods [68, p. 169]. In this method, the Jacobian matrix is thus replaced by an approximation $\boldsymbol{B}_{i+1}^t$ that fulfills the secant equation [70, p. 210]. The secant equation can be stated for the case introduced above as

$$\Delta\boldsymbol{r}_i^t = \boldsymbol{G}(\tilde{\boldsymbol{d}}_i^t) - \boldsymbol{G}(\tilde{\boldsymbol{d}}_{i-1}^t) = \boldsymbol{B}_{i+1}^t \Delta\tilde{\boldsymbol{d}}_i^t. \tag{4.28}$$

For problems in which the displacement vector has a size of $n = 1$, the approximate Jacobian $\boldsymbol{B}_{i+1}^t$ becomes a scalar variable. In this case, the above equation is sufficient to determine the approximate Jacobian $\boldsymbol{B}_{i+1}^t$. For other systems with $n > 1$, the system is underdetermined and an additional condition is necessary for the computation of $\boldsymbol{B}_{i+1}^t$. In the Broyden method, the additional condition is that the updated matrix $\boldsymbol{B}_{i+1}^t$ is a minimum modification from the previous approximation of the Jacobian $\boldsymbol{B}_i^t$ (see, e.g., [68, p. 169]).

With these conditions, a formula to update the approximate Jacobian in the Broyden method can be constructed as [68, p. 170]

$$\boldsymbol{B}_{i+1}^t = \boldsymbol{B}_i^t + \frac{(\Delta\boldsymbol{r}_i^t - \boldsymbol{B}_i^t \Delta\tilde{\boldsymbol{d}}_{i,i+1}^t) \otimes \Delta\tilde{\boldsymbol{d}}_{i,i+1}^t}{\Delta\tilde{\boldsymbol{d}}_{i,i+1}^t \cdot \Delta\tilde{\boldsymbol{d}}_{i,i+1}^t}. \tag{4.29}$$

Eq. (4.29) does not define an initial approximation $\boldsymbol{B}_0$. If no other information is available, the initial approximation $\boldsymbol{B}_0$ is set to the unit matrix. The approximation improves with

an increasing number of iterations – which implies that the Broyden method needs a certain number of iterations for good performance.

A more computationally efficient version of the Broyden method already proposed by Broyden in [69, p. 580] is obtained by approximating the inverse of the approximate Jacobian matrix $(\boldsymbol{B}_{i+1}^t)^{-1}$, since this overcomes the disadvantage of solving a linear system of equations to compute the displacement increment in the first line of Eq. (4.27).

Quasi-Newton least squares method

The next method from the class of quasi-Newton methods explained here is the quasi-Newton least squares (QNLS) method. The quasi-Newton least squares method, one of the most recent methods for convergence acceleration, was initially published by Degroote et al. [13] and Haelterman et al. [71] in 2009.

The basic idea of the quasi-Newton least squares method is to minimize the unknown residual $\boldsymbol{r}_{i+1}$ of the next implicit iteration, which can also be expressed with the residual difference

$$\boldsymbol{r}_i + \Delta\boldsymbol{r}_{i,i+1} = \boldsymbol{r}_{i+1}. \tag{4.30}$$

The residual differences $\Delta\boldsymbol{r}_{i,j}$ of the previous implicit iterations can be expressed using the residuals in the iterations i and j as $\Delta\boldsymbol{r}_{i,j} = \boldsymbol{r}_j - \boldsymbol{r}_i$. For the residuals, the displacements according to Eq. (4.17) can further be substituted as

$$\Delta\boldsymbol{r}_{i,j} = \boldsymbol{r}_j - \boldsymbol{r}_i = \boldsymbol{d}_j^t - \tilde{\boldsymbol{d}}_j^t - \boldsymbol{d}_i^t + \tilde{\boldsymbol{d}}_i^t = \Delta\boldsymbol{d}_{i,j}^t - \Delta\tilde{\boldsymbol{d}}_{i,j}^t. \tag{4.31}$$

In Eq. (4.31), the difference of the displacements is used as $\Delta\boldsymbol{d}_{i,j} = \boldsymbol{d}_j^t - \boldsymbol{d}_i^t$ and, accordingly, $\Delta\tilde{\boldsymbol{d}}_{i,j}^t = \tilde{\boldsymbol{d}}_j^t - \tilde{\boldsymbol{d}}_i^t$. With the known residual differences $\Delta\boldsymbol{r}_{i,j}$, a linear combination can be formulated as

$$\Delta\boldsymbol{r}_{i,i+1} \approx \sum_{j=0}^{i-1} \alpha_j \Delta\boldsymbol{r}_{i,j}. \tag{4.32}$$

By combining Eq. (4.30) and (4.32), it is possible to construct a minimization problem in which $\boldsymbol{r}_{i+1}$ is minimized. This minimization can be formulated as

$$\min \left\| \boldsymbol{r}_i + \sum_{j=0}^{i-1} \alpha_j \Delta\boldsymbol{r}_{i,j} \right\|_2 = \min \left|| \boldsymbol{r}_i + \boldsymbol{V}_i \boldsymbol{\alpha}_i \right||_2 . \tag{4.33}$$

In Eq. (4.33), the only unknowns are the coefficients α_j, for which the minimization problem is solved in the least squares sense, as indicated by the operator $\min ||*||_2$. To this end, the matrix $\boldsymbol{V}_i$ containing the residual differences is QR-decomposed.

By inserting Eq. (4.31) into the minimized argument of Eq. (4.33), the following equation is obtained:

$$\boldsymbol{r}_i + \sum_{j=0}^{i-1} \alpha_j \Delta\boldsymbol{r}_{i,j} = \boldsymbol{r}_i + \sum_{j=0}^{i-1} \alpha_j \Delta\boldsymbol{d}_{i,j}^t - \sum_{j=0}^{i-1} \alpha_j \Delta\tilde{\boldsymbol{d}}_{i,j}^t \approx \boldsymbol{0}. \tag{4.34}$$

Since the expression on the left-hand side is assumed to be close to zero, Eq. (4.34) can be rearranged to

$$\boldsymbol{r}_i + \sum_{j=0}^{i-1} \alpha_j \Delta \boldsymbol{d}_{i,j}^t \approx \sum_{j=0}^{i-1} \alpha_j \Delta \tilde{\boldsymbol{d}}_{i,j}^t. \tag{4.35}$$

If an analogy is assumed between the residual differences $\Delta \boldsymbol{r}_{i,j}$ in Eq. (4.32) and the displacement differences $\Delta \tilde{\boldsymbol{d}}_{i,j}^t$, the unknown displacement difference can be obtained as

$$\Delta \tilde{\boldsymbol{d}}_{i,i+1}^t \approx \sum_{j=0}^{i-1} \alpha_j \Delta \tilde{\boldsymbol{d}}_{i,j}^t. \tag{4.36}$$

With Eq. (4.36), the solution for the initial displacements in the next implicit iteration can then be computed (like in the Newton-Raphson method) as $\tilde{\boldsymbol{d}}_{i+1}^t = \tilde{\boldsymbol{d}}_i^t + \Delta \tilde{\boldsymbol{d}}_{i,i+1}^t$. In this procedure, the Jacobian matrix and its inverse are not calculated explicitly. Despite this, Eq. (4.36) serves as an approximation to find the displacement differences, similar to the Newton-Raphson method, in the first line of Eq. (4.27).

It is possible to extend the single-time-step QNLS method to a multi-time-step version in which the data of the previous n time steps $t-1, t-2, ..., t-n$ are reused in order to obtain a better approximation for the coefficient vector $\boldsymbol{\alpha}_i$. A common problem when using a multi-time-step version of the QNLS method is a possibly ill-conditioned matrix $\boldsymbol{V}_i$ in Eq. (4.33). This problem can be reduced significantly by using a filtering technique as proposed in [13] and improved in [14]. After computing the QR-decomposition, small eigenvalues of the matrix $\boldsymbol{V}_i$ can easily be identified in the upper triangular matrix $\boldsymbol{R}$ obtained in the QR-decomposition. If an entry on the main diagonal of $\boldsymbol{R}$ is below a certain threshold, the corresponding row and column of $\boldsymbol{V}_i$ are deleted.

Another improvement to the multi-time-step version is the hybrid method as proposed in [72]. With the hybrid method, the QNLS method uses the multi-time-step version at the beginning of a certain time step until, after a certain number of iterations, a switch to the single-time-step version is performed.

4.6 Coupling framework *comana*

The software framework used in this work for all FSI simulations is the coupling manager *comana* [26]. *comana* is a flexible software library written in C++ that can be used to design a program for solving partitioned multi-physics problems. Many different existing commercial and in-house solvers are coupled to *comana*. For solving structural problems with the FEM, the in-house developed high-order FEM solver AdhoC [73] and the well-known commercial FEM solver ANSYS [74] are (amongst others) coupled to *comana*. For solving fluid problems, the established open source FVM solver OpenFOAM [75] and the BEM solver pan*MARE*, see [24] and [76], are two of the fluid solvers coupled to *comana*. Different solvers for the fluid and structural problems are added as adapter libraries to *comana*. The design and details of this software library are explained in detail in [26], [62], and [77], so that it is sufficient to outline the basic ideas here.

Since multi-physics problems can be composed of more than two subproblems and can also involve volume coupling instead of just surface coupling, the coupling framework *comana* can also be used to solve these more general multi-physics problems, as shown in [62] and [77]. Nonetheless, this work focuses on the common FSI approach with one structural

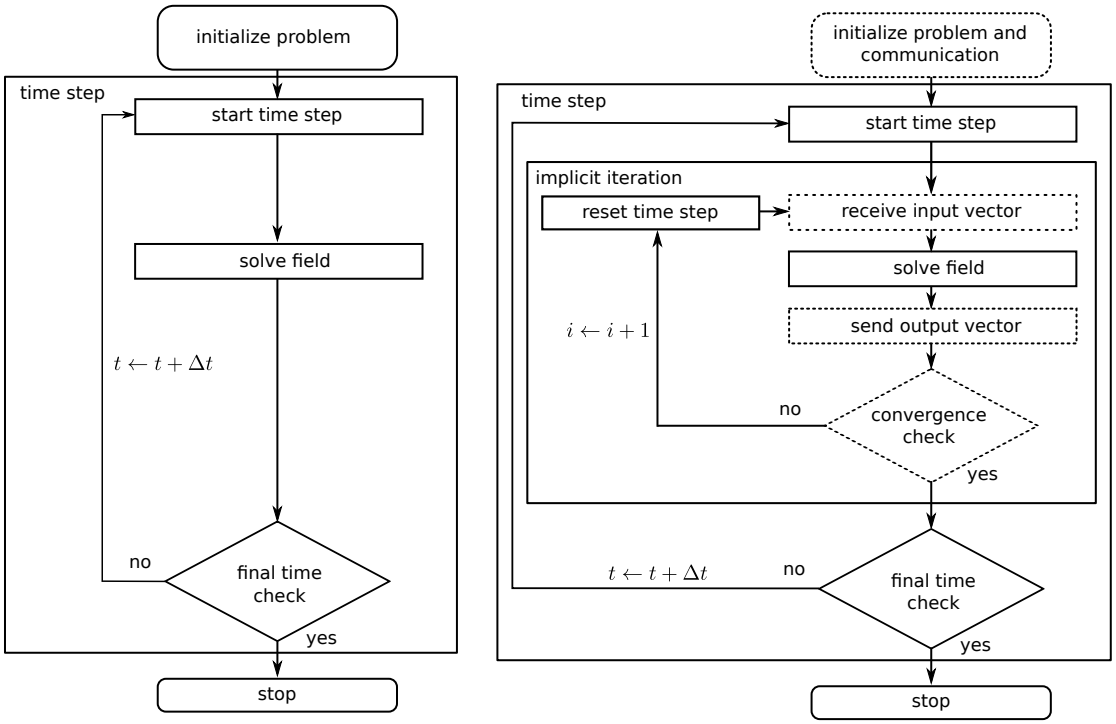


Figure 4.4: Left: unmodified slave process, right: modified slave process

solver and one fluid solver coupled on a common interface – applying this approach to complex and application-related problems.

Following the idea of the partitioned approach, as explained, for example, in [7], the subproblems are treated independently by separated advanced and highly specialized solvers. The coupling requires some modifications to the standard solution process of solvers that were initially not designed for the solution of FSI problems. For partitioned FSI problems with *comana*, three processes are executed if the fluid and the structural solver are both run as single processes. The master process is the program that is built using the coupling library *comana*, and the solution procedure of the master process is visualized in Fig. 4.1. For both the fluid and the structural problem, a slave process is executed, which is controlled by the master process to enable the FSI coupling. The exchange of data between the different processes is realized with the message passing interface (MPI).

As indicated by Eq. (4.1) and (4.2), the solvers are viewed as black boxes. Only small modifications to the standard solution process of these specialized solvers are necessary. The simplified unmodified and the modified slave process for an implicit solver are visualized in Fig. 4.4. Dashed lines indicate steps in which communication with the master process is necessary. After the initialization of the communication and the problem, the time-step loop is entered. The slave solver is now modified to wait for the input vector, which is sent by the master process. In the case of a structural solver, this input vector contains the tractions or forces on the FSI interface, meanwhile, for the fluid solver, it contains the displacements. After applying the input vector as a boundary condition in the solver, a normal solution step can be performed in the slave process. After this step, the computed output vector is sent to the master process. For the structural solver, this output vector

contains the displacements on the FSI interface – and for the fluid solver, it contains the forces or tractions acting on the interface. In the next step, the slave process again waits for the master process to provide information on whether the current time step has converged. If the time step does not converge, the solver must be reset to its initial state at the beginning of the time step. To this end, it is necessary to enable the solver to conduct such a reset – which can be challenging if the source code of the solver is not accessible or modifiable. If the time step does converge, the solver can continue as in the unmodified solution process.

For most of the simulations, the problems are solved in a staggered procedure as indicated in Fig. 4.1, which means that while one of the slave processes is running, the other slave processes are idle, awaiting new input before their corresponding field is solved. A parallel solution is possible and can reduce solution time, but convergence difficulties often arise when the problems are solved in a parallel procedure and require special convergence acceleration methods like a modified version of the QNLS method to work efficiently [78]. If the required solution time for both solvers is in a similar range, the total solution time can theoretically be reduced by up to 50% with this approach [79, p. 871].

5 Floating wind turbine

In the next chapters, the concepts introduced in the last three chapters are applied to complex and application-related problems. The simulation setup for all the maritime application cases is explained and the beneficial results that can be obtained with FSI simulations are presented. This chapter presents the first example from the field of offshore renewable energy: the floating wind turbine.

5.1 Introduction

The floating wind turbine (FWT) considered here [80] was simulated in the scope of the project "Hydrodynamic and structural-mechanical optimization of a semi-submersible for offshore wind turbines" (HyStOH) as part of the subproject "Fluid-structure interaction and optimization of floating platforms for offshore wind turbines" (FSIOpt) [81] funded by the Federal Ministry for Economic Affairs and Energy of Germany (BMWi). The results obtained in the scope of this collaborative research project were previously published in [62, 82, 83, 84].

The development of a sophisticated setup for a detailed simulation of a complex application is usually the result of cooperative work. This also applies to the setup presented here. The setup for the fluid simulation was developed in the "Institute for Fluid Dynamics and Ship Theory" at TUHH, by Netzband et al. [85]. The coupling and the structural setup for the simulation were developed in the workgroup "Numerical Structural Analysis with Application in Ship Technology" at TUHH [84].

The author of the presented thesis contributed to this project by implementing the case-specific coupling and by solving significant implementation issues in the basic setup of the simulation to enable the generation of the results. Furthermore, the author worked on the efficient implementation and parallelization of the presented setup on the high-performance cluster of TUHH, reducing the simulation time by a factor of more than 3. He also helped in the production and interpretation of the presented application-related results.

5.2 Conceptual design

For large water depths of more than 50 m, a classical offshore wind turbine (OWT) design with a foundation in the sea bed is not economical due to the excessive cost for the foundation of the OWT in the sea bed [86, p. 7]. Thus, a different design is necessary to harvest the strong and steady wind far away from the shore in large water depths. In the case of large water depths, an FWT is the most promising alternative to the classic wind turbine design. With the possibility to move in the three translational and three rotational degrees of freedom, predicting the load on the structure and the dynamic behavior in different conditions becomes much more challenging. Here, a turbine with a power output of 6 MW is analyzed as the first prototype for the considered design concept.

The principle setup of the floating wind turbine is shown in Fig. 5.1. The initial design of the floating wind turbine consists of a floating platform composed of four floaters, arranged in a rhombic shape and connected by a hollow structure of welded stiffened steel plates. The use of welded stiffened steel plates for the FWT enables the manufacturing of the structure in a shipyard – which is beneficial since transporting an FWT on land is complicated and expensive. To keep the FWT from capsizing, the platform is filled with ballast water, allowing a shift in the center of gravity for the entire structure.

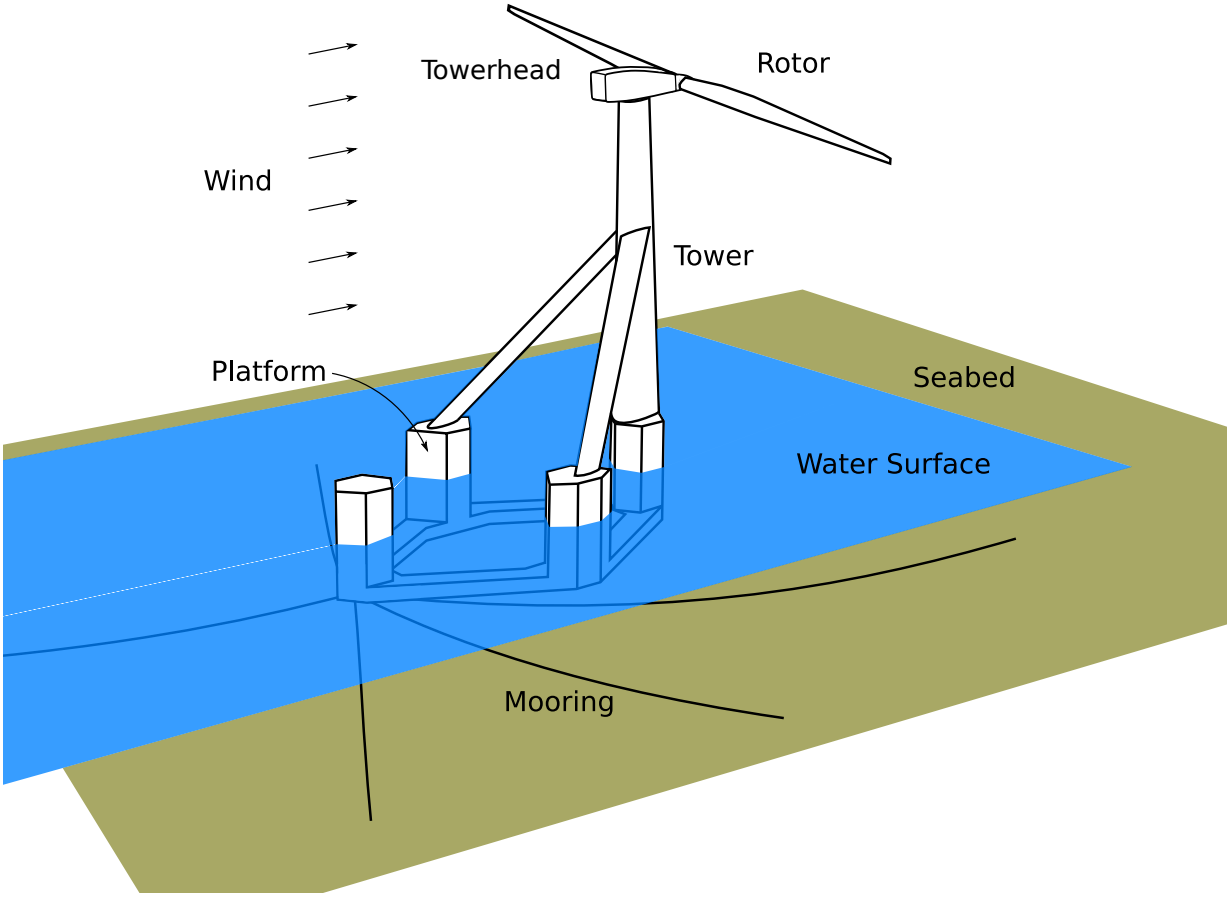


Figure 5.1: Schematic overview of the floating wind turbine

On the floater that is on the upwind side of the FWT, the mooring lines are attached to the structure by a turret buoy that allows the entire structure to rotate freely around the anchor point. The turret buoy also allows for electric power transmission to conduct the energy generated by the FWT. The five mooring lines have a length of approximately 350 m and are arranged at a uniform angle of 72°.

The tower is mounted on the floater on the downwind side of the FWT and is supported with struts that are based on the two floaters in the middle.

Contrary to most existing wind turbines, a fixed head is mounted on top of the tower. A fixed drive is beneficial compared to a yaw drive, as this design allows to significantly reduce the weight and the cost for the towerhead. The weight reduction of the tower head also reduces the weight of the other parts of the FWT, as the counterweight can be minimized, and the load on the supportive structure of the tower head is also reduced. Further weight

reduction is achieved by a two-blade instead of the more common three-blade design for the rotor. These modifications serve to optimize the production costs of the FWT, making the design more economical.

In order to align the rotor blades in the wind, the entire structure can rotate around the anchor point of the mooring. To support the FWT's self-alignment in the wind, the rotor is placed on the tower's downwind side and the tower has an aerodynamically optimized airfoil shape. This shape also reduces wind resistance and disturbance. Reduced wind disturbance is advantageous because the load change on the rotor in the wake of the tower is reduced. The downwind design also helps to reduce the risk of the rotor blades hitting other parts of the FWT. The measures of the platform and the rotors are shown in Fig. 5.2. The height of the tower is approximately 90 m.

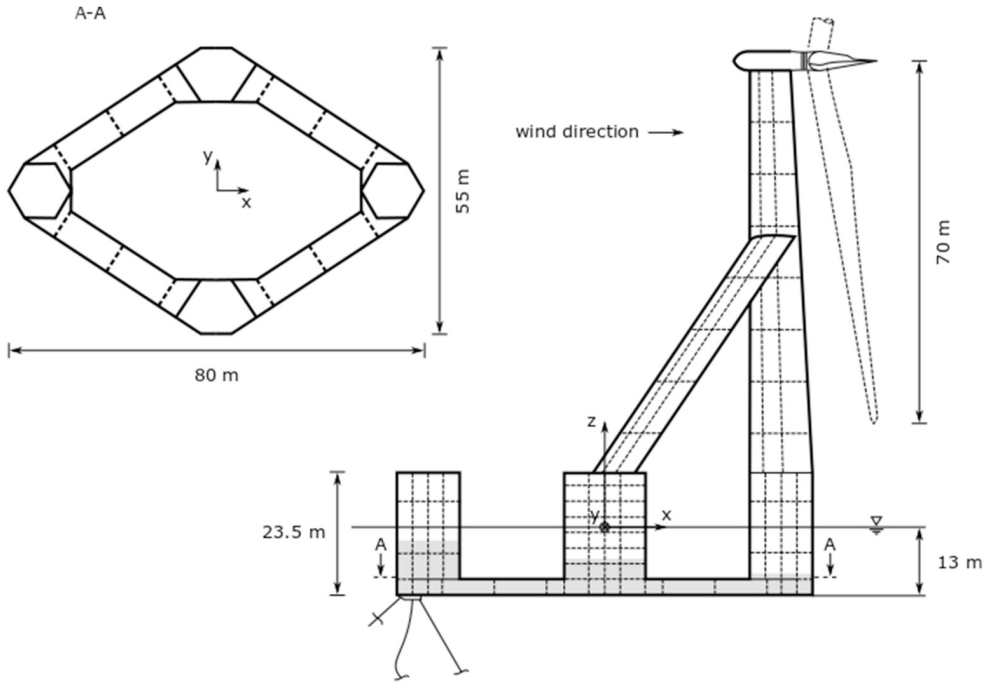


Figure 5.2: Dimensions of the FWT [84]

5.3 Simulation setup

The FWT is simulated using a partitioned FSI simulation with the coupling manager *comana* introduced in Sec. 4.6. For the fluid part of the FSI simulation, the BEM explained in Sec. 3.3 and implemented in *pan*MARE [76] is used.

The fluid simulation can be split into two different parts. The waves of the water interact with the platform of the FWT, and the wind forces interact with the tower and the rotor blades.

Two different scenarios are considered here. The first scenario explores an extreme wave condition in which the highest load expected throughout the lifetime of the FWT is considered. The rotor blades are fixed, and their pitch is adjusted to minimize the load acting on them in this scenario. Since the water waves are expected to be the critical load, the wind forces are neglected. The waves acting in this scenario are based on the Joint

North Sea Wave Project (JONSWAP) wave spectrum, and the wave elevation is shown in Fig. 5.3.

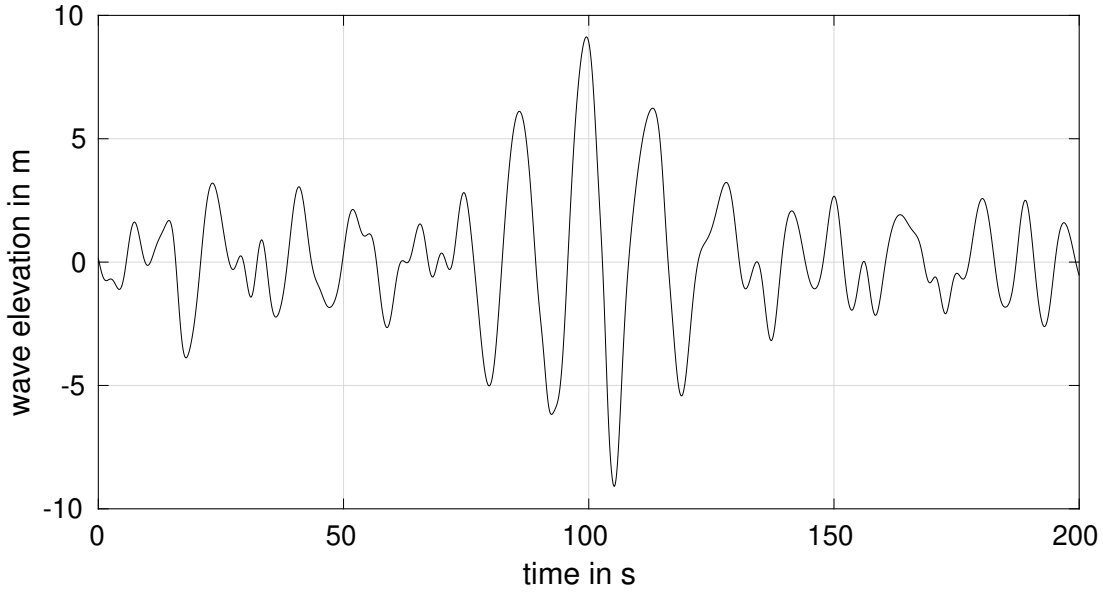


Figure 5.3: Wave elevation over time for extreme wave scenario

The second scenario considers the floating wind turbine in operating conditions under the influence of regular water waves and steady wind. The water waves are modeled with linear wave theory, and Morrison elements are used to account for viscous drag [25, p. 5] in the hydrodynamic part of the simulation.

In the second scenario, the wind velocity varies with the height over the water surface h_z according to

$$v_w = 12.6 \frac{\mathrm{m}}{\mathrm{s}} \left(\frac{h_z}{90\,\mathrm{m}} \right)^{\frac{1}{7}} . \tag{5.1}$$

This variation of the wind velocity is shown in Fig. 5.4.

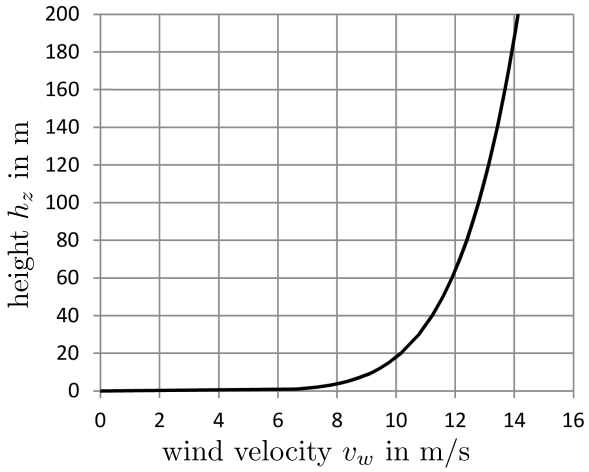


Figure 5.4: Wind velocity over water surface

A multi-point constraint is used to connect the hubs of the rotor blades with the top of the tower. The multi-point constraint allows for a relative rotation between the rotor and

the tip of the tower around the rotation axis of the rotor. All other relative translational and rotational motions between the top of the tower and the hubs of the rotor blades are restricted. To capture the effect of the generator that counteracts the moment caused by the wind forces, a varying generator torque is applied at the multi-point constraint to the rotor blades, depending on the rotation rate of the rotor.

Fig. 5.5 offers a visualization of the discretized fluid mesh in *pan*MARE. A refinement in the platform region close to the water surface is used to capture the wetted hull surface more accurately. The tower is discretized with force elements that do not influence the flow field but capture drag-lifting effects on the structure [85, p. 18]. The rotor wake is discretized with wake panels, as explained in Sec. 3.3.3, to account for lifting forces on the rotor blades and to accurately capture the flow field in the wake of the rotor. The discretization towards the hub and the tip of the rotor blades is refined as well, as can be seen in Fig. 5.5, to enable the accurate simulation of the rollup of the wake. Further parameters for the fluid simulation are listed in Tab. 5.1. [85]

Table 5.1: Fluid parameters of the FWT simulations in *pan*MARE.

variable	value
sea water density ϱ_{sw}	$1025\ \frac{\mathrm{kg}}{\mathrm{m}^3}$
sea water viscosity ν_{sw}	$10^{-6}\ \frac{\mathrm{m}^2}{\mathrm{s}}$
air density ϱ_a	$1.24\ \frac{\mathrm{kg}}{\mathrm{m}^3}$
air viscosity ν_a	$1.48 \cdot 10^{-5}\ \frac{\mathrm{m}^2}{\mathrm{s}}$

A correction of the drag and lift forces on the rotor blades is performed by means of empirical corrections obtained with Reynolds-averaged Navier-Stokes simulations. A validation for the aerodynamic model used in this simulation can be found in [76, p. 131].

The mooring lines are modeled in an extended version of *pan*MARE with concentrated mass nodes connected by spring-damper elements. The mooring model is based on the approach of Hall ([87]) and accounts for inertial, buoyancy, and current effects. More details about the mooring model can be found in [85, p. 19].

For the structural part of the FSI simulation, the commercial FEM solver ANSYS [74] is used. The structural model for the platform and the tower of the FWT was developed by JÖRSS – BLUNCK – ORDEMANN GmbH as part of the DESPOW subproject in the joint HyStOH project [88]. The tower and the struts are modeled with approximately 600,000 linear Reissner-Mindlin-type shell elements with approximately 599,000 nodes. The platform is modeled with around 384,000 linear shell elements with 358,000 nodes. The shell elements are predominantly quadrilaterals and have an edge length of approximately 0.05 m. To prevent locking, a mixed enhanced strain formulation [89] is integrated in ANSYS for linear shell elements [90, p. 601]. The rotor blades are modeled with geometrically nonlinear 2-noded Timoshenko-type beam elements. Point masses are used to model the approximately 5000 t of ballast water inside the platform.

Since the high number of degrees of freedom leads to a computationally expensive simulation, the static condensation introduced in Sec. 2.8 is used to reduce the number of degrees of freedom for the tower and the platform. Since the static condensation is limited to linear problems and the deformation of the rotor is too large to be modeled

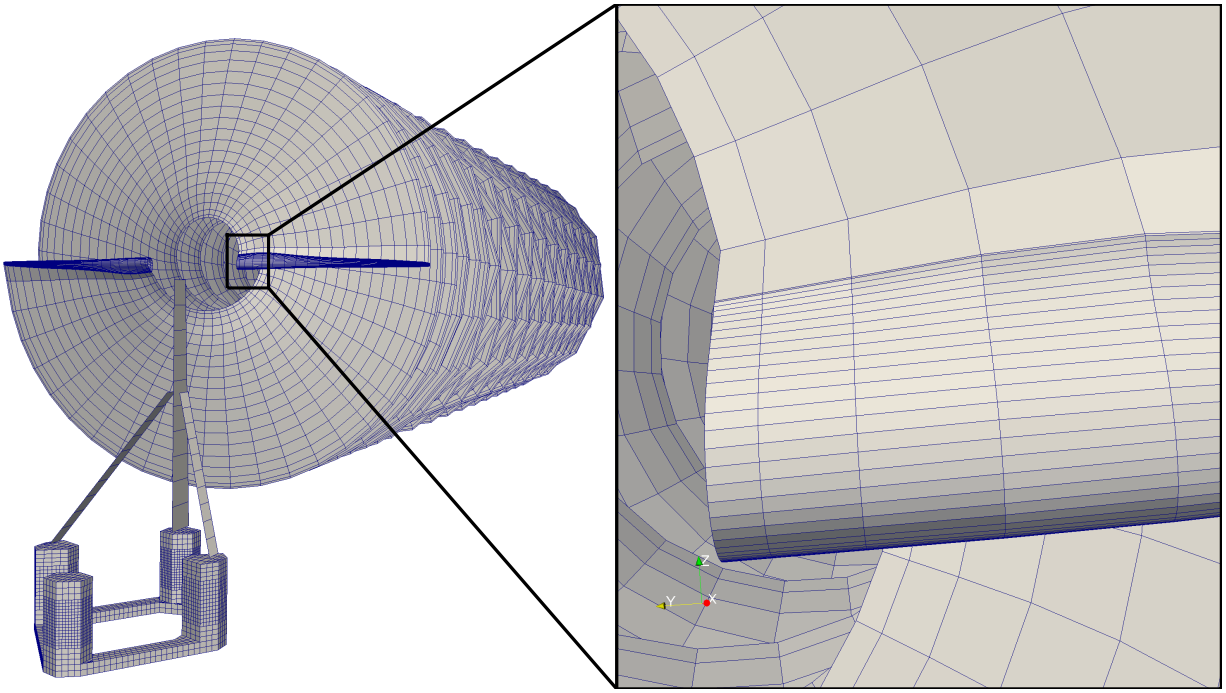


Figure 5.5: Fluid mesh of the FWT.

with a geometrically linear approach, the number of degrees of freedom for the rotor cannot be condensed. Nonlinearities due to large rigid body rotations can be modeled with static condensation in ANSYS since a corotational technique is employed. With static condensation, the number of nodes is reduced to 66 nodes for the tower (reduction of nodes by approximately 99.99%) and to 917 for the platform (reduction of nodes by approximately 99.7%). The steel is modeled as a linear elastic isotropic material according to Eq. (2.8), with the material parameters listed in Tab. 5.2.

Table 5.2: Parameters for the structural part of the FWT simulations.

variable	value
steel density ϱ_s	7850 $\frac{\text{kg}}{\text{m}^3}$
steel Young's modulus E_s	$2 \cdot 10^{11}$ $\frac{\text{kg}}{\text{ms}^2}$
steel Poisson's ratio μ_s	0.3
generalized alpha parameter α_m	0
generalized alpha parameter α_f	0.4
generalized alpha parameter β	0.45
generalized alpha parameter γ	0.9

The generalized alpha method introduced with Eq. (2.62) is used as the time integration method with the parameters listed in Tab. 5.2. The parameters for the generalized alpha method are chosen such that numerical damping is apparent in the simulation. Numerical damping is used to reduce high-frequency oscillations. The verification of certain aspects of the coupled simulation is presented in [84].

The simulation is conducted for up to 200 s with a time-step size of 0.1 s on the high-performance cluster of the Hamburg University of Technology (TUHH).

5.4 Simulation results

In the extreme wave scenario, the load on the FWT platform and tower is expected to be maximal. Since the rotor blade pitch is controlled to minimize wind force and the blades are fixed with a break, wind loads on the blades are not considered in this scenario. The state of the wind turbine when the extreme wave hits the structure is shown in Fig. 5.6. The pressure distribution due to the high wave amplitude leads to a strong heave, surge, and pitch motion of the structure, imposing a high load on the mooring and the FWT structure.

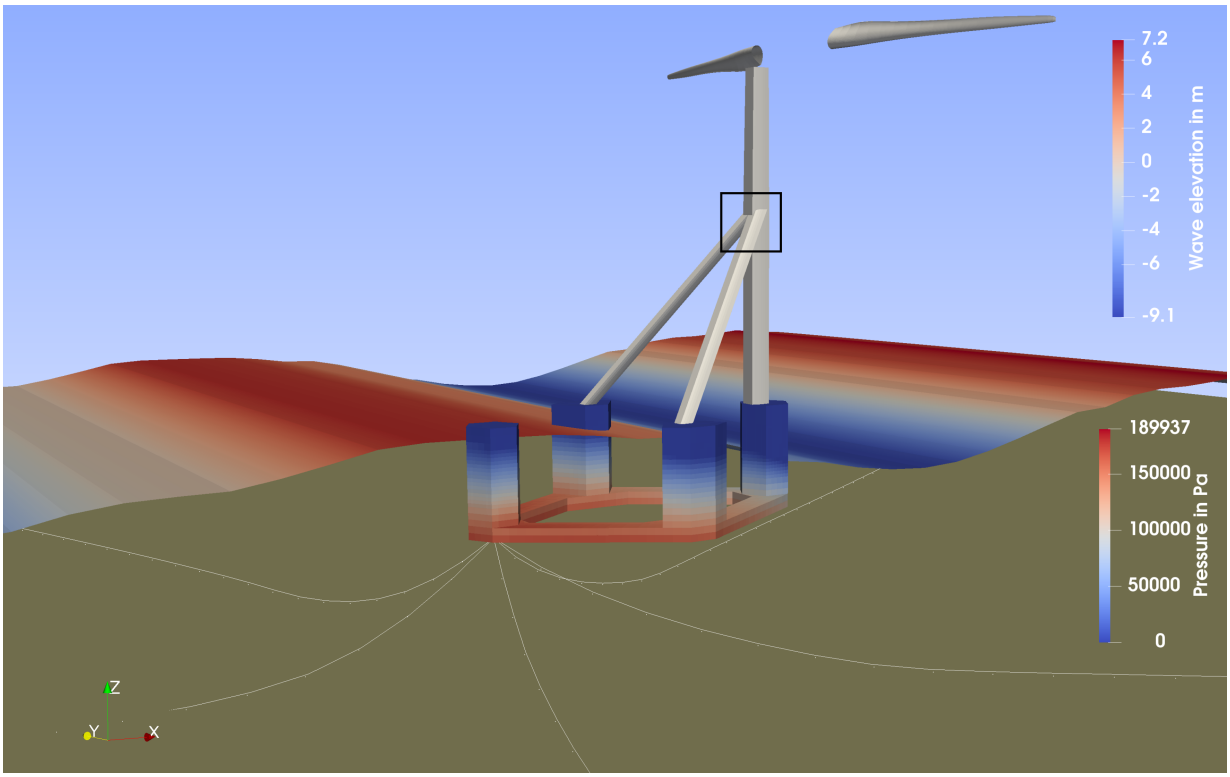


Figure 5.6: Pressure distribution on the platform in extreme wave scenario

After the results of the structural simulation are obtained, the condensed results derived with the reduced system are expanded to provide results for the full system, as described in Sec. 2.8. A critical loading point is the junction between the struts and the tower, marked with a black rectangle in Fig. 5.6. The equivalent stress at this junction is visualized in Fig. 5.8. The color scale represents the von Mises equivalent stress according to Eq. (2.71). Stresses higher than 235 MPa, above the yield stress of standard steel, are indicated in black. The deformation in this area would be in the plastic regime and, thus, the stress in this region cannot be determined accurately with a linear-elastic material model for this type of steel. The point where the stress is maximal is at the stiffeners directly at the junction between the upper parts of the struts and the tower. In Fig. 5.8b, this point is indicated with a yellow circle. The equivalent stress at this point is higher than 1000 MPa.

Further, a large area around this part of the structure is marked in black, indicating that it is considered too high for this scenario. Fig. 5.8a shows additional areas in which the stress is much higher than the yield stress – such as the stiffening structure on the side of the tower and the connection between the lower part of the struts with the tower. The maximum stress exceeds the yield stress in other regions of this scenario, too, like at the connection of the floaters with the struts.

Due to the high loading visible in this scenario, a modification of the structural design was proposed [84, p. 13]. As the platform was not designed stiff enough, a considerable deformation of the platform appeared in the extreme wave scenario. To reduce this effect, additional tubes were introduced at the top region of the floater to increase the stiffness to prevent bending of the structural parts that connect the floaters. The equivalent stress of the FWT with this reinforced design can be seen in Fig. 5.7. Like the rotor blades, the tubes are modeled with Timoshenko-type beam elements and, therefore, are only represented as lines in Fig. 5.7. The diameter of the tubes is 2.5 m, and the wall thickness is 3 cm. The tubes are not considered to be under significant loading due to the water wave, which is why they are not part of the FSI interface.

When looking at the equivalent stress of the platform, a pattern of points with a concentration of stress is visible that is not physically plausible. These points of high stress are caused by static condensation. External forces can only be applied to master nodes, as explained in Sec. 2.8. This means that the pressure force acting on the structure is lumped to the master nodes, and the numerical effect of this force concentration are points that show unphysically high stress. Despite the concentration of stresses at the master nodes and the approximation of the reduced damping and the mass matrix compared to the full system, as explained in Sec. 2.8, the reduced system delivers reasonable results for the stresses, as shown in [82].

Fig. 5.9 shows the equivalent stress of the FWT with the reinforced design from the same perspective as in Fig. 5.8.

With the reinforcement, the maximum equivalent stress in all regions is reduced to below 230 MPa. Since the platform is designed stiffer, the stiffeners at the connection between the tower and the struts are under an uncritical loading. The regions of high stress are also distributed more evenly. A verification of the results obtained for this scenario by comparison of the rigid body behavior between the coupled simulation and a reference simulation in *pan*MARE can be found in [84, p. 5].

The second scenario of the reinforced FWT in operating conditions is shown in Fig. 5.10. This scenario exhibits quite a high stress of up to 200 MPa in the top region of the tower, caused by the wind load acting on the rotor. However, the wind forces for this scenario are overpredicted due to numerical errors, as will be shown in the following.

To investigate the overprediction of the wind forces, a reduced simulation is conducted in which only the rotor of the FWT with a fixed hub is simulated. In this coupled simulation, the rotor is simulated as a rigid body in ANSYS with different time-step sizes. The results are compared to those obtained in a rigid body simulation performed in *pan*MARE. Fig. 5.11 shows the thrust and the moment acting on the rotor due to the wind forces. The thrust is the total force acting on the rotor in the x-direction and the moment refers to the moment acting around the rotation axis (which is also the x-axis). For the simulation with a time-step size of 0.1 s, the thrust is approximately 20 % higher than in the reference simulation. The wind moment is approximately 13 % higher than the reference simulation at the beginning of the simulation, where the rotor rotation rate is fixed, and 5 % higher at

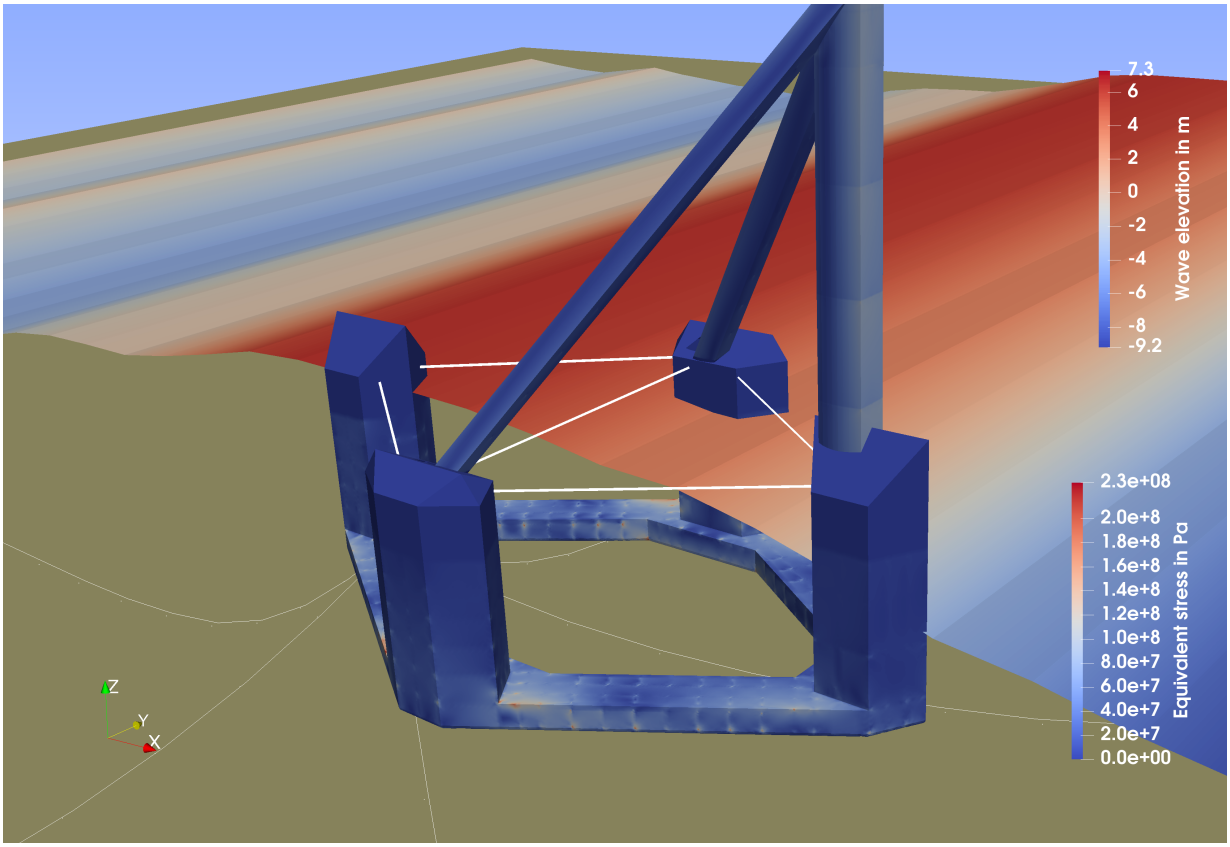
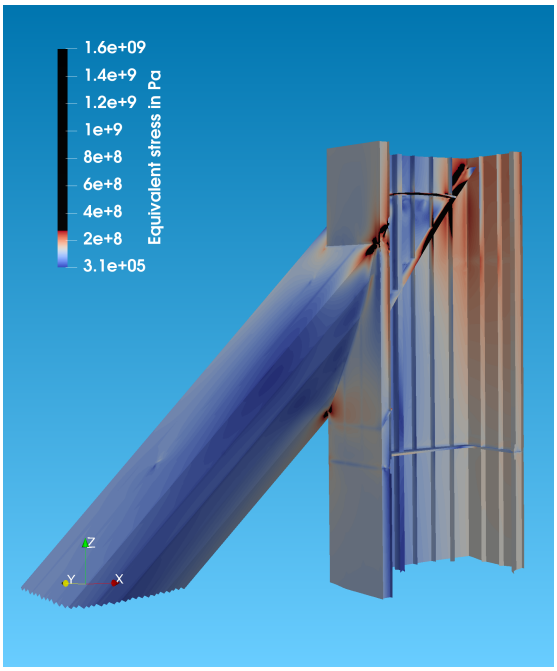


Figure 5.7: Reinforced platform and equivalent stress in extreme wave scenario

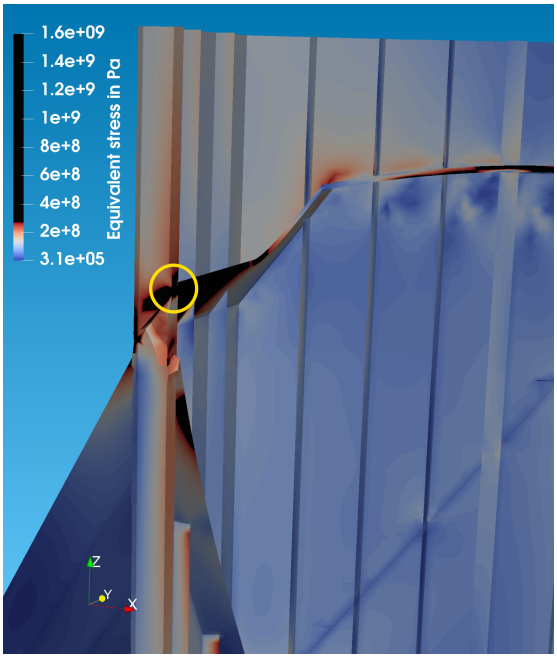
the end of the simulation. The higher wind forces also lead to an approximately 15 % higher rotation rate of the rotor at the end of the simulation for the coupled simulation with a large time step compared to the reference simulation in *pan*MARE. The rotation rate in the *pan*MARE reference simulation is approximately equal to $0.26\frac{1}{s}$, which corresponds to an angle increment of around 10°.

To obtain results with acceptable accuracy for the prediction of the thrust and the moment caused by the wind forces, a time-step size of 0.01 s is required. However, this time-step size would lead to an unacceptably high computational time for the simulation of the full FWT.

The reason for the difference between the coupled simulation and the reference simulation lies in the different calculation of the velocities of the panels in *pan*MARE in both cases. If the rigid body solver implemented in *pan*MARE is used, the velocities of the panels of the spinning rotor are computed based on the exact tangential velocities of a rotation around a certain axis. For the coupled simulation, this approach is not directly feasible if the rotor is modeled with an elastic material. This is because the motion of the rotor in the coupled simulation is not purely rotational, as there is also a significant velocity in the axial direction of the rotor resulting from the deformation of the rotor blades. In the coupled simulation, the velocity is therefore computed as an interpolation of the current and the previous time step in a similar manner as in Eq. (2.49). A possible solution for this problem, without increasing the computational time significantly by a small time-step size is presented later in Sec. 8.1.2.

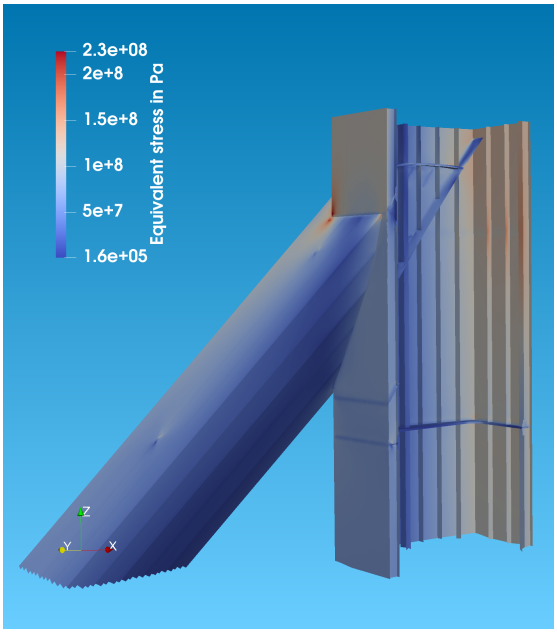


(a) Section view front

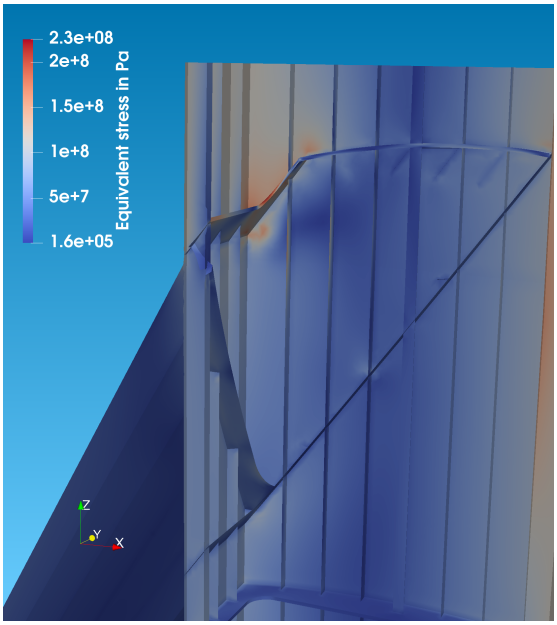


(b) Close-up – the maximum stress is marked with a yellow circle

Figure 5.8: Equivalent stress at the joint between tower and struts for the initial design



(a) Section view front



(b) Close-up

Figure 5.9: Equivalent stress at the joint between tower and struts for the reinforced design

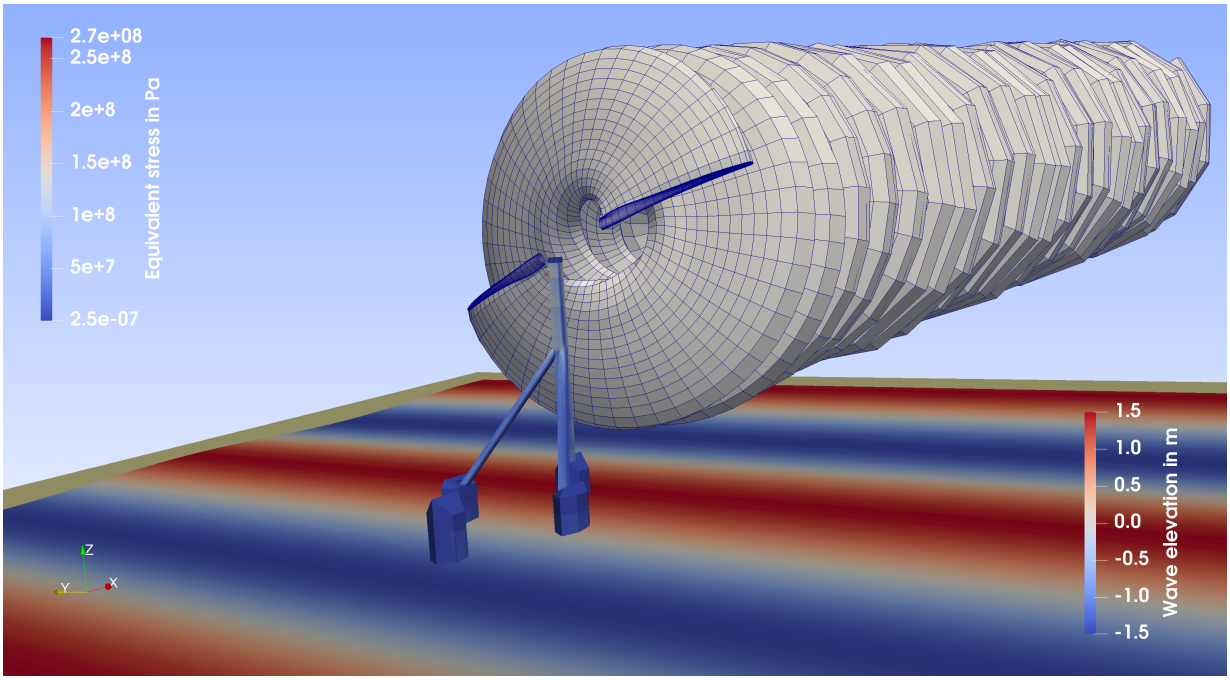
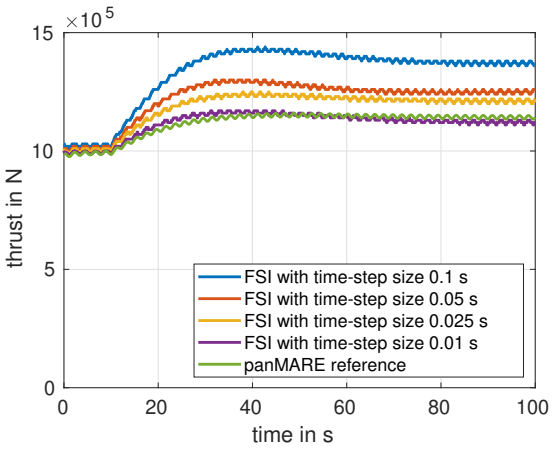
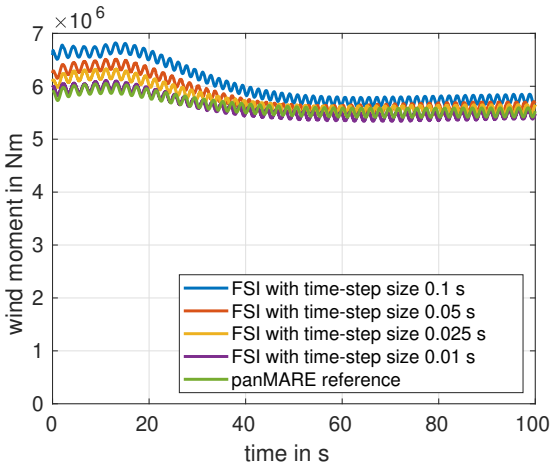


Figure 5.10: Equivalent stress of the reinforced FWT in operating conditions



(a) Thrust on the rotor due to wind forces

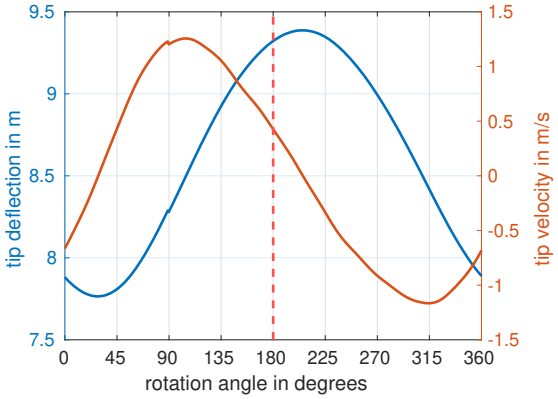


(b) Moment on the rotor due to wind forces

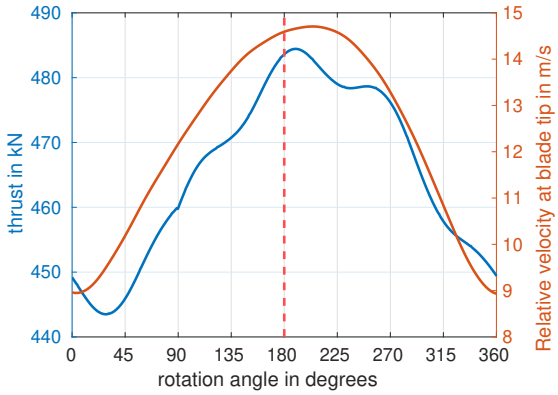
Figure 5.11: Comparison between stiff FSI simulation with different time-step sizes and *pan*MARE simulation

Despite this issue, another physical effect can be observed for the simulation of the flexible rotor blade (without the rest of the FWT) with a fixed rotation axis. For this simulation, the wind distribution shown in Fig. 5.4 is used together with a small time step of 0.01 s.

The difference in the wind velocity with increasing height over the water surface leads to an oscillating motion of each rotor blade in the axial direction, which coincides with the x-direction. Fig. 5.12a shows the deflection and the velocity in the x-direction for a single rotor blade. The blue line corresponds to the deflection, while the orange line refers to the tip velocity. 0° refers to the bottom position, and 180° refers to the top position of the blade tip. The top position of the rotor blade is also indicated by a red dashed line. The rotor blade moves in the positive direction from 0° to 360°.



(a) Tip deflection and velocity in the x-direction.



(b) Thrust and relative velocity between tip and the wind in the x-direction.

Figure 5.12: Deflection, velocities, and thrust on one rotor blade over one rotation; the red dashed line indicates the top position of the propeller.

In Fig. 5.12b, the thrust acting on one entire blade (not only on the tip) in the x-direction is indicated with a blue line. The relative velocity between the tip (represented by an orange line in Fig. 5.12a) and the wind at the tip in x-direction according to Eq. 5.1 is indicated with an orange line in Fig. 5.12b.

If the effects on the side where the rotor blade moves upwards (from 0° to 180°) were the same as on the side where the rotor moves downwards (from 180° to 360°), the plots would be symmetric with respect to the red dashed line. This would, for example, be the case for blades that rotate very slowly. The high thrust in the top position leads, in combination with the fast rotational movement, to a positive velocity at the tip of the blade in the direction of the wind – up to a rotational angle of about 205° (25° after the top position). When the blade moves in the wind direction, the relative velocity between the wind and the blade (indicated with an orange line in Fig. 5.12b) is reduced. On the other hand, the relative velocity is increased when the blade moves against the direction of the wind – which mostly occurs when the rotor moves downwards (from 180° to 360°). Naturally, the relative velocity between the wind and the rotor correlates strongly with the thrust acting on the blade. This correlation is also visible in Fig. 5.12b, even though the total thrust is here compared to the relative velocity only on the tip, which leads to small deviations.

As a result of the effect, the average thrust when the blade moves upwards (from 0° to 180°) is approximately 460 kN, while the average thrust when the blade moves downwards

(from 180° to 360°) is approximately 470 kN. Therefore, the offset between both sides is approximately 10 kN. In the case of an OWT with a foundation in the seabed or an onshore wind turbine, the offset in thrust is effectively resisted by the foundation. In the case of a self-aligning FWT, however, the offset leads to a yaw motion of the whole FWT, causing the rotor to be inadequately positioned in the wind. If the entire FWT is simulated with a larger time-step size, the yaw motion explained above is clearly visible, despite the overestimation of the rotor thrust and moment, as shown in Fig. 5.11.

To overcome the problem of an unintended yaw motion of the FWT, one possible solution is to even out the misalignment by shifting the ballast water in the platform, as detailed in [84, p. 13].

In conclusion, the FSI simulation allows to analyze certain aspects more accurately and allows to observe physical effects that cannot be captured by a structural simulation without FSI or a fluid simulation with a rigid body simulation of the FWT. With classical approaches such as a separate FEM simulation, it is necessary to simplify the dynamic interaction between the structure of the FWT and the sea waves. In this case, the external forces acting on the FWT must be estimated beforehand. The accurate estimation of dynamic loading is challenging and requires a higher safety factor. Furthermore, a correct selection of the Dirichlet boundary conditions is necessary to suppress the rigid body modes of the structure, even if the loading on the structure is estimated correctly.

The rigid body simulation in combination with a fluid solver allows, for the most part, to accurately predict the motion of the FWT. Despite this, the yaw motion of the FWT due to the blade deformation cannot be predicted with this approach.

With the presented approach, it is possible to capture these effects – at the cost of a longer simulation time and increased computational power requirements. For the simulations presented here, a computational time of several days on a high-performance cluster was required. The FSI simulation approach presented here is, therefore, too computationally expensive for early-stage development, as the standard tool to analyze small changes or to perform extensive parameter studies. However, it can be used at a later development stage to reduce the number of prototypes required for the development of a FWT that is commercially viable. In addition to this, the FSI approach becomes more attractive with the anticipated increase in computational resources in the future.

6 Wave energy converter

As part of collaborative research between Isfahan University of Technology (IUT) and TUHH, funded by the Alexander von Humboldt Foundation, a two-body point absorber wave energy converter (WEC) was developed at IUT. Theoretical consideration and extensive experimental tests at IUT determined the optimal hydrodynamic coefficients and the optimal dimensions for the WEC [91]. At the same time, detailed FSI simulations conducted at TUHH in scope of this project served to validate the experimental results and provide deeper insight into the behavior of the full two-body WEC. Validation results of the hydrodynamic coefficients of the WEC are published in [92]. The FSI simulation results presented here are based on the master thesis of Hssini [93].

6.1 Conceptual design and experimental measurements

Various design options are available for WECs. For example, overtopping systems collect water from waves in a large fixed tank on an elevated water level. The collected water, traversed through a turbine, generates electricity. Oscillating water column systems drive a turbine using pressurized air, compressed inside a chamber by the incoming waves. These structures usually require a large fixed structure, located close to the shore, in which water is collected. Fixed or floating pitching devices can generate electricity through a rotational movement of the device, induced by the waves. Some concepts allow placement further away from the shore – but electricity can only be generated if the waves come from the right direction. Two-body point absorber WECs can be placed far away from the shore in deep waters, and, due to their rotational symmetric design, they are independent of the direction of the incoming wave. In comparison to other WEC systems, they are smaller. This advantage allows an easier and more flexible installation in deep waters [94].

Two promising designs for two-body point absorber WECs are *Powerbuoy*, developed by Ocean Power Technologies [95], and *Wavebob*, developed by Wavebob Ltd. [96]. The design analyzed in this work follows the design of *Powerbuoy* PB150.

Fig. 6.1 shows the basic configuration and the chosen measures of the WEC determined in [91]. A submerged body consisting of a heave plate fixed to a spar is connected to a power take-off (PTO) system that is attached to the floater. To generate electricity with a two-body WEC, a relative motion between both bodies is necessary. The floater is supposed to generate lift against the relatively steady submerged body. The submerged body is designed to have high resistance against lifting forces, providing support against the motion of the floater. The WEC is scaled down by a factor of 23 compared to an industrial-scale WEC [92, p. 182] to reduce the manufacturing cost and enable extensive experimental testing in a wave tank.

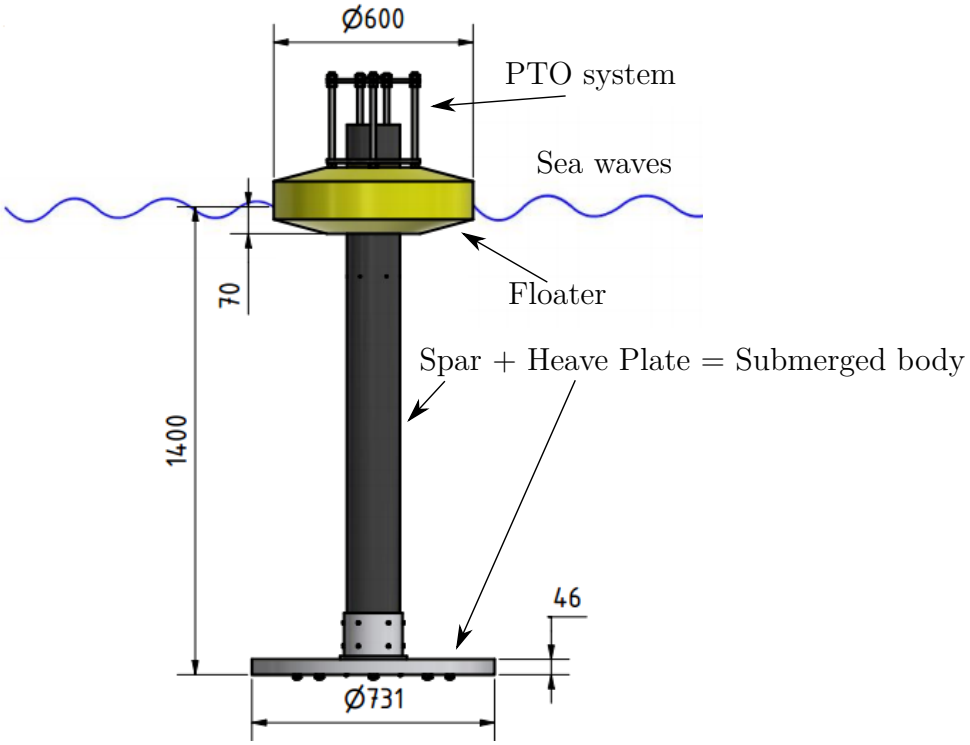


Figure 6.1: Concept and measures of the WEC (modified from [92]).

6.2 Simulation setup

The FSI setup considered here models the WEC as two rigid bodies (floater and submerged body) coupled by a linear spring-damper system.

OpenFOAM [75, 97] is used to simulate the three-dimensional fluid problem with the FVM (see Sec. 3.2). Reynolds-averaged [98, p. 83] Navier-Stokes equations (Eq. (3.17)) based on the k-ε model developed by Jones, Launder and Spalding [99, 100] reduce the computational time, while still capturing turbulent effects, such as dissipation through turbulent vortices.

The PIMPLE Method delivers a solution for the velocity-pressure coupling (see Sec. 3.2.3). The implicit Euler method (defined in Eq. (3.24)) with a time-step size of 0.01 s is the time integration method in the simulation.

The fluid is simulated as a two-phase flow using the volume of fluid method [101]. The volume of fluid method introduces a scalar parameter θ varying between 0 and 1 that defines the phase of each cell in the finite volume mesh. A value of $\theta = 0$ refers to the air phase, while a value of $\theta = 1$ refers to the water phase. Surface tension stabilizes the surface and prevents the mixing of the two fluids. The parameters for water and air used in the simulation are listed in Tab. 6.1.

A structured block mesh, consisting mainly of cuboid cells, is used. The cells are adapted to the geometry of the bodies with snappyHexMesh in OpenFOAM. The bodies are modeled as holes in the domain, and the motion of the bodies is captured by means of the ALE mesh motion approach, which moves the cells and faces at the surface of the body according to the displacements set by *comana*. This mesh motion approach moves the cells inside the

Table 6.1: Fluid parameters of the WEC simulations in OpenFOAM.

variable	value
water density ϱ_w	$1025\ \frac{\mathrm{kg}}{\mathrm{m}^3}$
water viscosity ν_w	$1.09 \cdot 10^{-6}\ \frac{\mathrm{m}^2}{\mathrm{s}}$
air density ϱ_a	$1\ \frac{\mathrm{kg}}{\mathrm{m}^3}$
air viscosity ν_a	$1.48 \cdot 10^{-5}\ \frac{\mathrm{m}^2}{\mathrm{s}}$
surface tension at the interface	$0.07\ \frac{\mathrm{N}}{\mathrm{m}}$

fluid domain according to the displaced boundary based on the Laplace equation [102]. To this end, the interDyMFoam solver [103, p. 88] is used. This solver combines the volume of fluid method with the ALE mesh motion approach. A refinement toward the water surface and the surface of the bodies captures the water surface and the motion of the bodies in the fluid simulation more accurately. Fig. 6.2 shows a visualization of the mesh around the WEC and the phase coefficient θ. Here, blue refers to the water phase, while red indicates the air phase. Cutting off the upper part of the spar prevents pronounced mesh distortions in the gap between the upper part of the spar and the floater. The outer dimensions of the FV-mesh are $3\,\mathrm{m} \times 2\,\mathrm{m} \times 2.5\,\mathrm{m}$, as can be seen in Fig. 6.3.

The waves on the inlet boundary of the domain are generated as third-order Stokes waves [104] with a wave height of 0.1 m and a frequency of 0.5 Hz with olaFlow [105]. Wave absorption introduced with olaFlow at the outlet boundary, which is located on the opposite side of the inlet boundary, prevents reflections of waves at the boundary. The waves travel in the positive x-direction. In Fig. 6.3, the inlet is thus situated on the left side of the fluid domain, while the outlet is located on the right side.

In ANSYS [74], the floater and the submerged body are modeled as rigid bodies by meshing the surface of the bodies and connecting the meshed surface to a pilot node with a point mass and a moment of inertia in all coordinate directions. In Fig. 6.2, the surface mesh of the floater and the submerged body are marked in green. Two yellow points indicate the pilot nodes of the respective body to which the surface mesh is constrained. The green line between both bodies connects both pilot nodes and indicates a linear spring and damper element that represents the PTO system. At the beginning of the simulation, the pilot nodes and the spring and damper elements are located on the z-axis. The stiffness of the spring element k_{PTO} is set to 2.16 kN/m, and the damping resistance R_{PTO} of the damper is set to 1.06 kN s/m. In his master's thesis, Hssini [93] determined these coefficients as the optimal coefficients for the PTO system at a frequency of 0.5 Hz. There, the hydrodynamic coefficients of the floater and the submerged body are determined separately based on non-coupled fluid simulations in OpenFOAM. Later, the hydrodynamic coefficients are used in a one-dimensional model to compute the optimal spring stiffness and damping resistance.

Tab. 6.2 lists the mass of both bodies and the moments of inertia. These values are chosen in a similar manner as in the WEC experiments [92]. Gravitational acceleration, acting in the negative z-direction on both bodies and the fluid, is set to $g = 9.81\ \frac{\mathrm{m}}{\mathrm{s}^2}$.

The time-step size in the simulation is set to 0.01 s, and the simulation is performed for several wave periods.

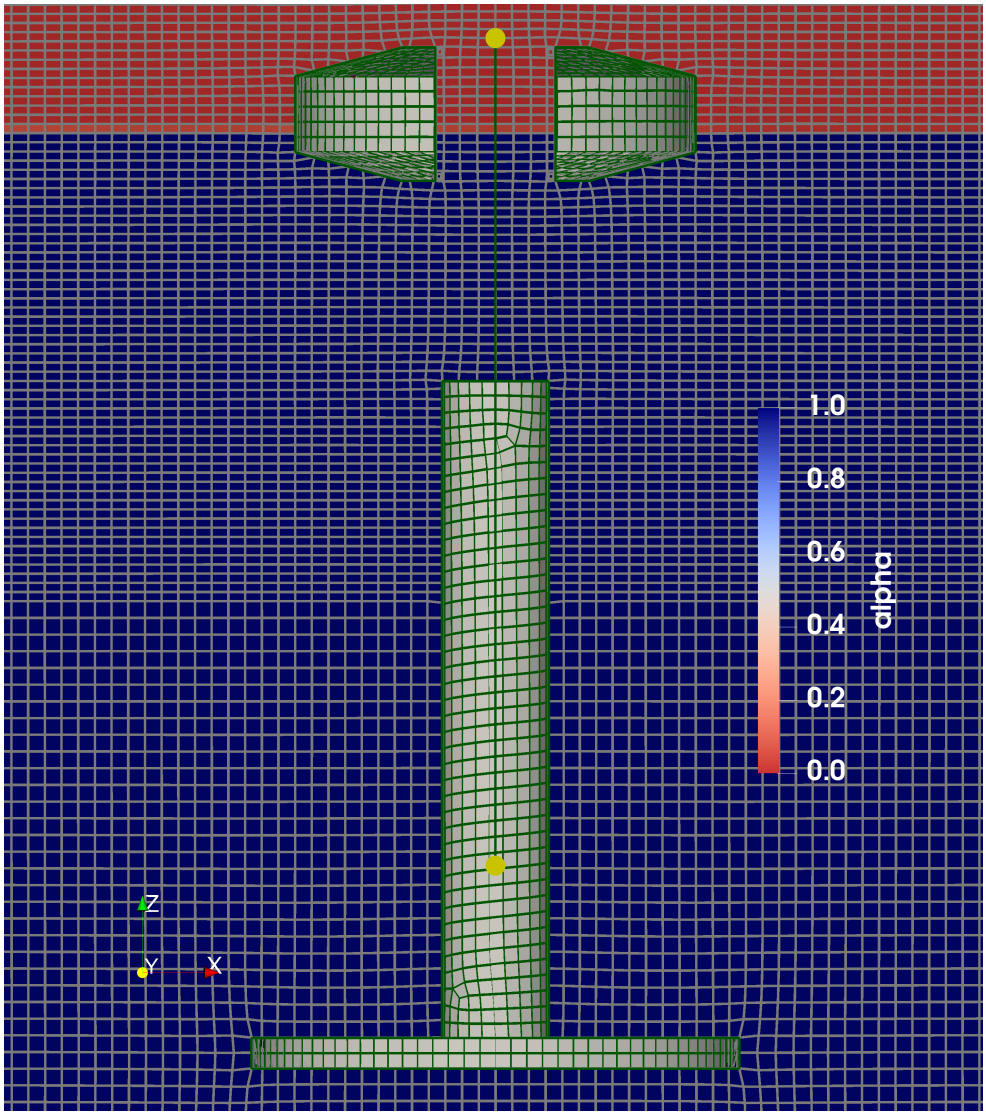


Figure 6.2: Mid-section of the volumetric fluid mesh (in grey) and surface structural mesh (in green) of the WEC and phase coefficient at the beginning of the simulation. The positions of the pilot nodes of both bodies are indicated by yellow points.

To restrain the rigid body modes of the WEC, catenary mooring lines are modeled as 2-noded linear beam elements with low bending stiffness. The mooring lines are connected to the submerged body at the pilot node of the WEC at a distance of approximately 0.5 m to the z-axis. The four mooring lines are attached to the seabed at the anchor nodes in a water depth of 5 m and have an initial length of 10 m. Projected to the flat seabed, the mooring lines span an angle of 24° to the x-axis. The mooring lines have a cross-sectional area of 32 mm^2, a Young's modulus of 164 GPa, and a density of approximately 4350 kg/m^3. In the simulation, each mooring line consists of 44 elements. The mass of the submerged body was distributed along the mooring lines to obtain a more realistic behavior of the mooring lines. Fig. 6.3 shows the mooring elements at the beginning of the simulation. This simulation does not include any interaction between the mooring lines and the fluid simulation.

Table 6.2: Mass and moments of inertia for both bodies of the WEC.

WEC part	floater	submerged
mass	12.7 kg	18.5 kg
moment of inertia around x-axis	0.47 kg · m^2	12.4 kg · m^2
moment of inertia around y-axis	0.47 kg · m^2	12.4 kg · m^2
moment of inertia around z-axis	0.36 kg · m^2	2.1 kg · m^2

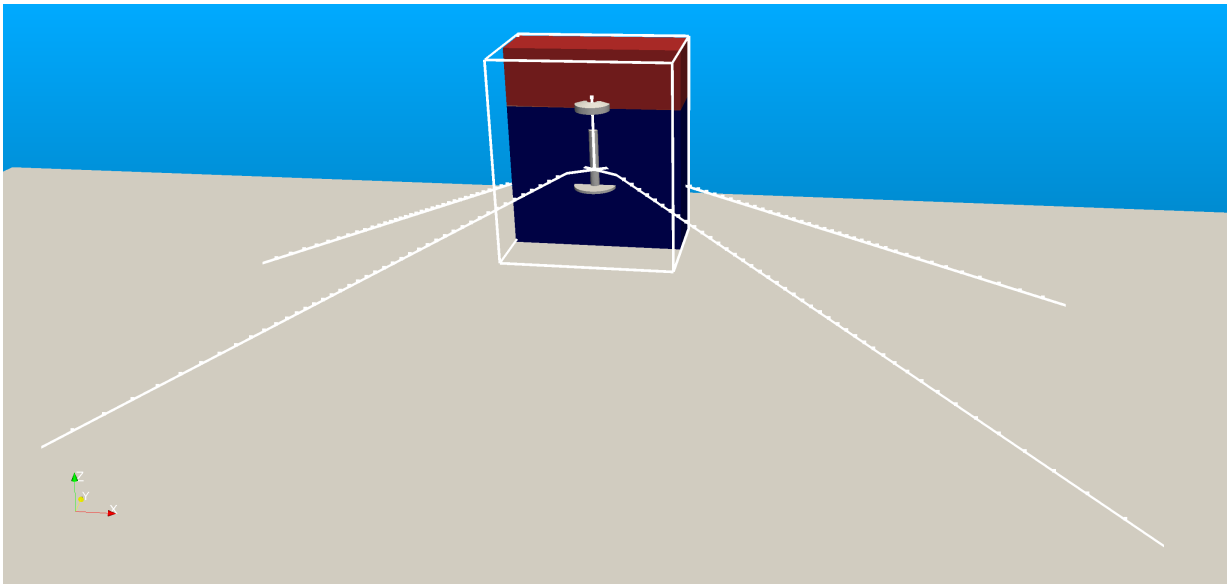
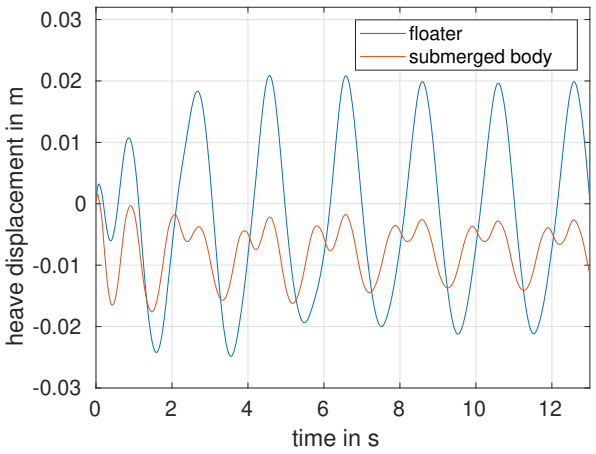


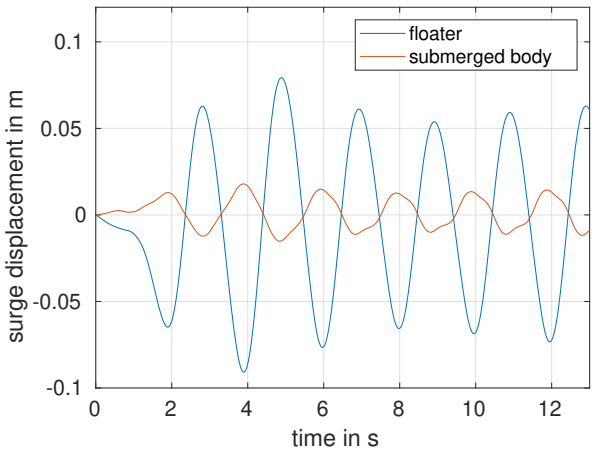
Figure 6.3: Visualization of the mooring lines together with the outline of the fluid mesh.

6.3 Simulation results

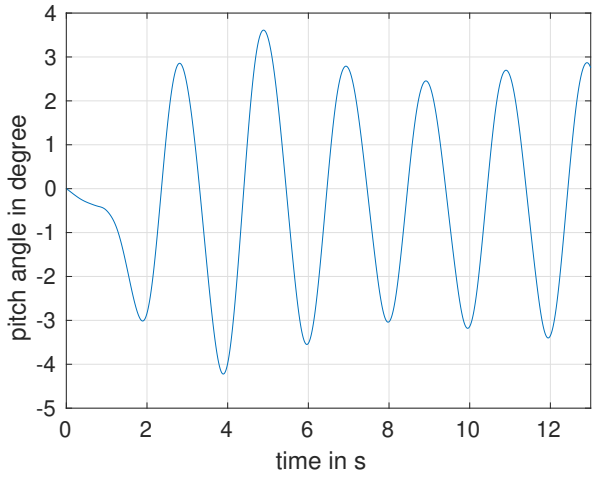
The motion of the WEC bodies at the pilot nodes is shown in Fig. 6.4. The heave and surge displacements in this figure are defined as the displacement of both bodies in z- and x-direction. The pitch angle is the angle of the line connecting the pilot nodes of both bodies to the plane span by the y- and z-axis. The time at which the wave trough of the n-th wave reaches the WEC is approximately equal to $0.5\,\mathrm{s} + 2\,\mathrm{s} \cdot n$. Accordingly, the time at which the wave crest of the n-th wave reaches the WEC is approximately equal to $1.5\,\mathrm{s} + 2\,\mathrm{s} \cdot n$. The floater in Fig. 6.4a shows a symmetric displacement in the heave direction. The motion of the submerged body has a much smaller amplitude compared to the floater – as is desired to achieve a large relative displacement. The surge displacement of both bodies (visible in Fig. 6.4b) is mainly caused by the pitch rotation of the WEC, shown in Fig. 6.4c, since the point around which the WEC rotates is slightly above the pilot node of the submerged body. Due to the rotation around this point, both bodies show a surge motion in opposite directions. Apart from this, almost no surge motion is visible. The asymmetrical displacement of the submerged body in Fig. 6.4a can also be explained by the strong pitch motion, which decreases the heave displacement of the submerged body in the range of higher displacements. The pitch motion of the WEC shows a symmetric oscillation with a pitch angle of approximately $\pm 3°$.



(a) Heave motion.



(b) Surge motion.



(c) Pitch motion.

Figure 6.4: Motion of the WEC.

The distance $d_{PTO,t}$ between the pilot node of the floater $\mathbf{X}_{pn,F,t}$ and the pilot node of the submerged body $\mathbf{X}_{pn,S,t}$ at time step t can be computed in every time step as

$$d_{PTO,t} = \|\mathbf{X}_{pn,F,t} - \mathbf{X}_{pn,S,t}\|_2 \ . \tag{6.1}$$

The velocity $V_{t,PTO}$, with which the spring and damper are compressed and expanded, can be estimated based on a linear interpolation between two time steps as (similar to Eq. (2.49))

$$V_{t,PTO} = \frac{d_{PTO,t+\Delta t} - d_{PTO,t}}{\Delta t}. \tag{6.2}$$

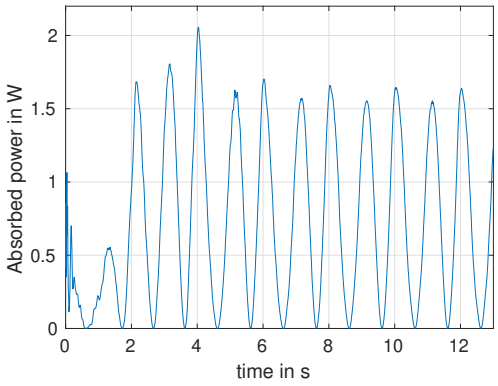


Figure 6.5: Power absorption.

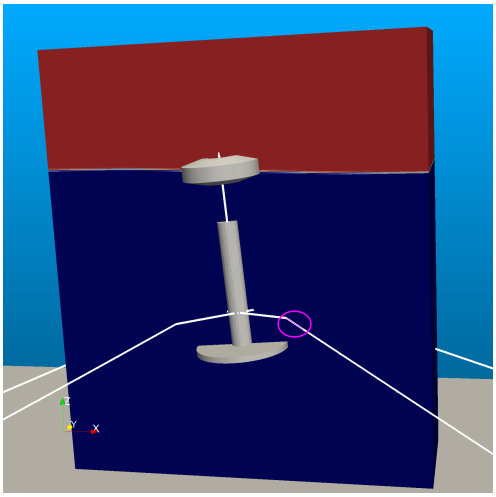
It is possible to estimate the power absorption through the damper (which represents the part of the PTO that generates electricity) based on the damping resistance and the relative velocities as [106, p. 18]

$$P_{t,PTO} = F_{PTO} V_{t,PTO} = \frac{1}{2} R_{PTO} V_{t,PTO}^2 \ . \tag{6.3}$$

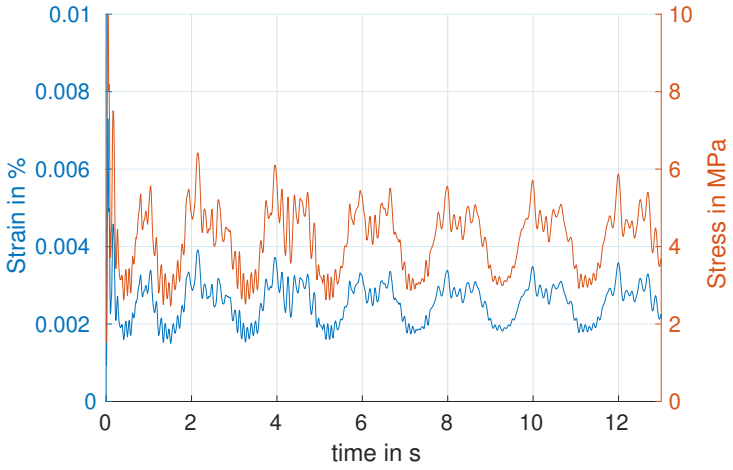
Fig. 6.5 shows the absorbed power $P_{t,PTO}$, computed based on Eq. (6.3), for the considered wave scenario with a wave frequency of 0.5 Hz and a wave height of 0.1 m. For this laboratory-scaled WEC, the mean value for the power absorption after the run-in period of the simulation is $P_{avg,PTO} = 0.78\,\mathrm{W}$. This estimation represents an upper estimate for the absorbed power of this wave scenario since it is important to consider additional losses – such as losses in the PTO mechanism, which strongly depend on the mechanism chosen for transforming the mechanical energy into electricity.

FSI simulations not only allow for the estimation of the power and a possible improvement of the design, but can also be used to compute the strain and stress in the mooring lines with an accuracy that can hardly be obtained with other methods.

Fig. 6.6b shows the strain and the stress for a mooring element attached to one of the fairlead points over time. The corresponding mooring element in the FSI simulation at $t = 12\,\mathrm{s}$ is highlighted and marked with a magenta circle in Fig. 6.6a. Clearly, the stress in the mooring lines stays below 8 MPa during the whole simulation. In addition to the wave period, Fig. 6.6b shows a higher frequency oscillation. The strain and stress are in a similar range in all mooring lines. The strain in the mooring lines becomes maximal in the elements closest to the fairlead point, even though the variation of the strain along the mooring lines stays below 10 %.



(a) Mooring element at fairlead point, highlighted in magenta at $t = 12\,\mathrm{s}$.



(b) Strain and stress over time for mooring element.

Figure 6.6: Strain and stress of mooring line element highlighted on the left-hand side.

The stress in the mooring lines computed here is, of course, not generally representative of a WEC in offshore conditions, as only one low wave height and one frequency have been considered so far. Furthermore, the interaction between the mooring lines and the water is not taken into account. Considering this FSI would probably also have a damping effect on the mooring lines and reduce the oscillation visible in Fig. 6.6b. Despite this, the results show the potential of the FSI simulation, e.g., for a correct dimensioning of the mooring and for choosing the right distribution between the weight of the mooring lines and the weight of the two bodies of the WEC. In order to design a WEC for offshore conditions over the full lifespan, it would be beneficial to also consider an extreme wave scenario, as was done for the FWT with the wave elevation, which can be seen in Fig. 5.3. Even though this could be possible with the presented setup, the dimensions of the wave tank in which the simulation is conducted would have to be scaled up by more than a magnitude (at least in the heave direction) because not only would the larger wave need to fit in the wave tank, but also the mesh deformation would increase drastically. This would increase the simulation time for the given case from several days to (at least) several months and, therefore, would pose a problem. Thus, using the BEM in the FSI simulation is recommended for an extreme wave scenario given the present state of computing power. Another approach to avoid strong mesh deformation could be immersed methods that prescribe the boundary conditions of the structure into the finite volumes of the fluid domain instead of moving the domain boundaries of the fluid domain.

Despite this, using the FVM instead of the BEM on the fluid side of the simulation is beneficial here for the low wave height and frequency: As shown by Rahimi et al. [92, p. 188] for the given WEC, the BEM is not able to accurately capture the added mass effect at different frequencies without modifications. Viscous effects lead to a higher added mass of the WEC bodies for low wave frequencies. The prediction of the added mass with an unmodified version of the BEM results in an added mass that is independent of the wave frequency.

Even though the WEC design presented here is only preliminary (especially with respect to the mooring line dimensioning), the benefits of conducting FSI simulations for this case are apparent. As for the FWT, the FSI simulation allows to predict the motion of this application case and, thus, potentially undesired behavior can be predicted without costly prototypes. Estimating the power absorbed by the PTO mechanism is also important, as this can help tune the WEC behavior to a higher power absorption. If more details about the PTO mechanism are considered, it could also be possible to tune the PTO mechanism in combination with the WEC parameters to obtain an optimal energy output.

7 Submersible mixer

In the scope of the project "Hydroelastic simulation of the acoustic behaviour of ship-propeller configurations with and without cavitation", funded by the German Research Foundation (Deutsche Forschungsgemeinschaft), FSI simulations of a submersible mixer were conducted that are presented in this chapter. The submersible mixer presented here is taken as the first example of a turbo machine rotor and used as a validation case for the propeller cases presented in the next chapter. Submersible mixers are used in industrial applications such as sewage water plants or in chemical process engineering. The simulations conducted are based on the experimental setup taken from Lass [20], who performed extensive experimental investigations measuring a variety of different parameters during the experiments. The FSI simulations presented here are performed with *pan*MARE on the fluid side and ANSYS on the structural side coupled by the coupling manager *comana*. The results presented in this chapter were previously published in [107].

7.1 Experimental setup

Since the experimental setup from Lass [20] is used to validate the FSI simulation, it is briefly explained in this section. A more in-depth explanation can be found in [20]. The experimental measurements of the submersible mixer were performed in a closed-looped channel in which the mixer was mounted. In this setup, an electric engine drives the mixer, which has an electric power consumption of 4.7 kW and a maximum thrust of approximately 2 kN at a rotation speed of $4.0\bar{3}\frac{1}{\mathrm{s}}$ [20, p. 61]. The submersible mixer is composed of multiple layers and has two blades, as can be seen in Fig. 7.1.

The inner layer is a metallic ring that serves as a stiffener for the middle layer. The middle layer is a glass-fiber reinforced polymer (GFRP) layer that defines the main geometry of the mixer. Strain gauges are applied to the GFRP layer in order to measure the strain on the surface between the middle layer and the outer layer. The outer layer is a soft polyurethane resin layer that is molded onto the GFRP layer. This is necessary to isolate the solder pads and the cable system of the strain gauges to prevent short circuits. All materials used for the submersible mixer have no dominant fiber direction and are therefore assumed to be isotropic. The properties of the different layers are listed in Tab. 7.1. These are the structural properties used in the FSI simulation. The outer diameter of the mixer is approximately 0.9 m.

To capture the full strain state on the surface of the GFRP layer, the experiment uses layered strain gauge rosettes [20, p. 67].

In addition to the strain, the inflow velocity (generated by the self-induced flow caused by the rotating mixer) was measured as well – at different locations within a distance of 0.45 m to the rotor plane in the main flow direction (coinciding with the x-direction) – using vane anemometers [20, p. 66]. Even though significant flow components in y- and z-direction are likely in the experiment, due to the complex flow field, the velocity components in these

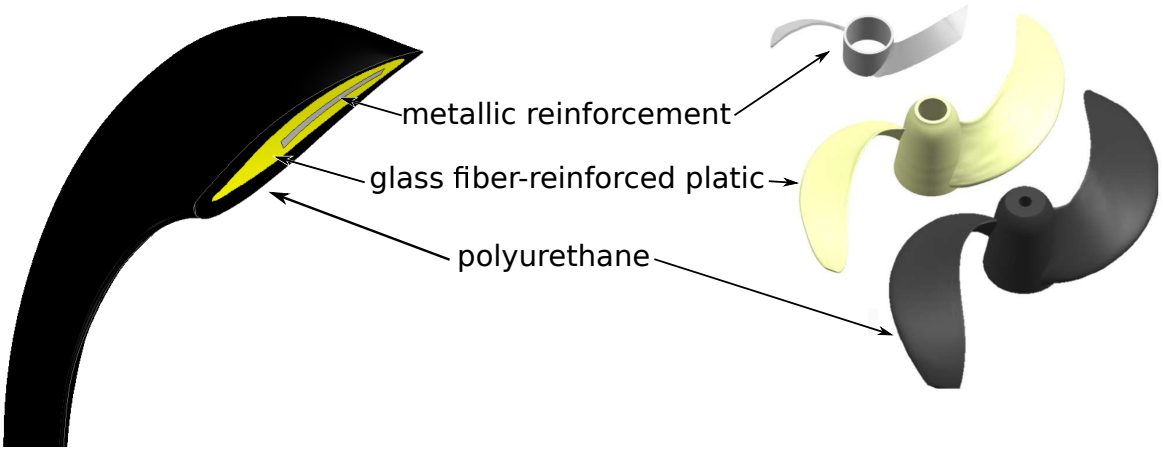


Figure 7.1: Multilayered structure of the submersible mixer (modified from [20, p. 62]).

Table 7.1: Properties of the different materials of the multilayered mixer [20, p. 87].

material	Young's modulus	Poisson number	density
metallic reinforcement	$1.1 \cdot 10^{11} \frac{\mathrm{N}}{\mathrm{m}^2}$	0.28	$7200 \frac{\mathrm{kg}}{\mathrm{m}^3}$
GFRP	$3.3 \cdot 10^{9} \frac{\mathrm{N}}{\mathrm{m}^2}$	0.27	$1500 \frac{\mathrm{kg}}{\mathrm{m}^3}$
polyurethane	$1.4 \cdot 10^{9} \frac{\mathrm{N}}{\mathrm{m}^2}$	0.35	$1200 \frac{\mathrm{kg}}{\mathrm{m}^3}$

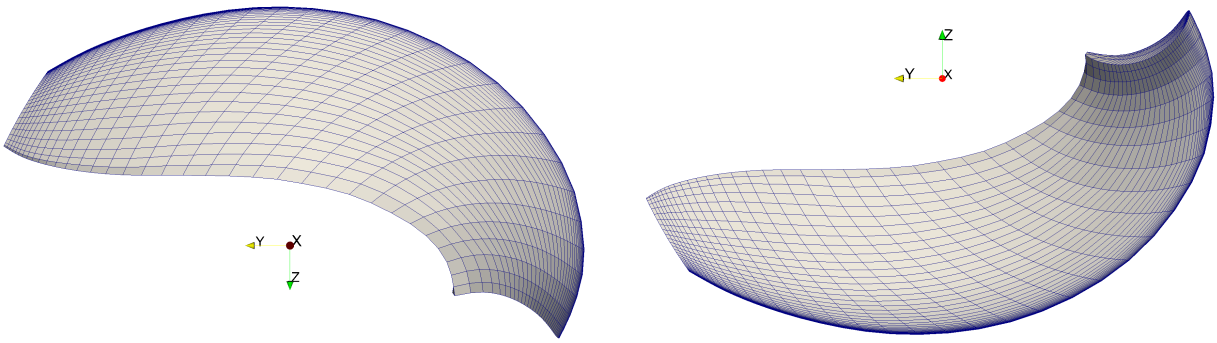
directions are not measured. Based on the measurement of the inflow velocity with the vane anemometers, the averaged axial inflow velocity can be approximated to be equal to $0.28 \frac{\mathrm{m}}{\mathrm{s}}$. With Eq. (3.54), the advance ratio can with this velocity be computed as $J \approx 0.077$.

Furthermore, the thrust acting on the mixer was measured in the experiment with a load cell [20, p. 64]. The high relative uncertainty of the torque measured in the motor unit amounts to 10 % [20, p. 65].

7.2 Simulation setup

The fluid simulation is conducted with *pan*MARE with the advance ratio computed above. With this, the inflow on the propeller is assumed to be uniform and steady. The density of the water flow around the mixer is assumed to be equal to $1000\mathrm{kg/m}^3$. The *Schoenherr line* friction model is used to correct the forces acting on the panels to account for viscous effects in the fluid simulation. The low advance ratio of the submersible mixer in the experiment can, due to self-penetration, lead to problems with the wake discretized by wake panels. Therefore, the relative wake length is set to only 0.45 rotations.

Based on the surface model of the submersible mixer, a regular mesh (visible in Fig. 7.2) is created in *pan*MARE with a resolution of 40 panels in the chord direction and 35 panels



(a) Pressure surface. **(b)** Suction surface.

Figure 7.2: Fluid mesh in *pan*MARE

in the span direction on both sides of the mixer blade. The fluid mesh is refined towards the blade's leading edge, as can be seen in Fig. 7.2.

The simulation on the structural side is performed in ANSYS with a linear elastic material formulation for all three materials. Geometrical nonlinearities (such as large rotations) are captured. The material parameters for the simulation are chosen as listed in Tab. 7.1.

The unstructured mesh in the structural simulation in ANSYS is created with ten-noded tetrahedral elements that are automatically generated with Delaunay mesh triangulation [108]. 3D mesh generation approaches based on Delaunay triangulation have the advantage that the time-consuming mesh generation for a good-quality mesh can be automatized, even for complex multilayered structures such as the submersible mixer. The minimum angle is maximized with this approach for the tetrahedral elements to avoid highly acute or obtuse angles.

One of the quantities that should be used to validate the FSI simulation are the equivalent strains obtained in the experiment based on the response of the strain gauge rosette. The results for the strains in the FSI simulation are sensitive to the resolution of the structural mesh. Therefore, three different FSI simulations with three different resolutions of the structural mesh are conducted. The coarse structural mesh consists of 8816 quadratic tetrahedral elements with a total of 14206 nodes. The medium structural mesh consists of 17294 quadratic tetrahedral elements with a total of 27744 nodes. The fine structural mesh consists of 45117 quadratic tetrahedral elements with a total of 69393 nodes. Fig. 7.3 offers a visualization of the coarse structural mesh – different layers indicated with the same colors as in Fig. 7.1.

The Newmark method as described in Sec. 2.5.2 is employed as the time integration method with the Newmark parameters $\gamma = 0.5$ and $\beta = 0.25$. Oscillations are reduced by means of stiffness-proportional Rayleigh damping according to Eq. (2.46) with a stiffness matrix multiplier of $\beta = 0.0001$. In the Newton-Raphson procedure used to solve the nonlinear equation system in the structural simulation (see Sec. 2.6), the maximum number of equilibrium iterations is increased to 30. The general simulation parameters are listed in Tab. 7.2 and will be explained in more detail in the following.

To control the rotational movement of the mixer, displacements are prescribed by *comana* on the hub of the blades in the structural simulation. These displacements are prescribed in such a way that the hub rotates around the rotation axis of the submersible mixer.

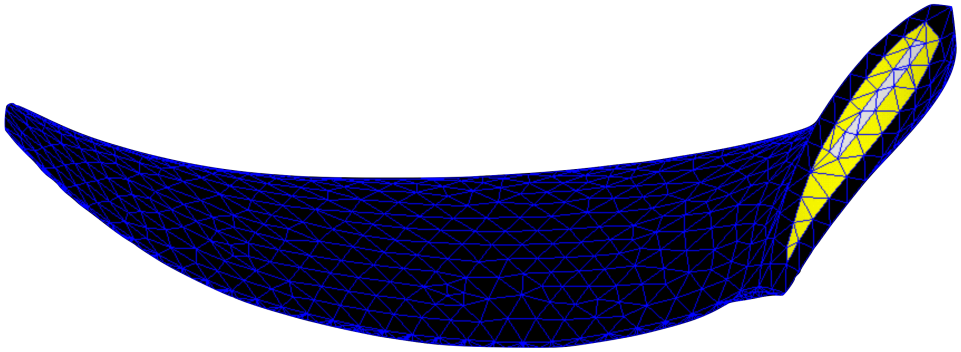


Figure 7.3: Coarse structural mesh of the multilayered mixer created with Delaunay triangulation.

Table 7.2: Parameters for the submersible mixer simulation.

parameter	value
angle increment $\Delta\alpha_f$	$2°$
time-step size Δt	0.00138 s
total simulation time t_e	$1800\Delta t \approx 2.48$ s
time of full rotational speed t_{rot}	$180\Delta t \approx 0.248$ s
traction ramp start t_{st}	$360\Delta t \approx 0.496$ s
traction ramp end t_e	$410\Delta t \approx 0.565$ s

Since the simulation is not initialized with the blades already rotating at full speed, the rotation speed of the propeller is gradually increased at the beginning of the simulation using a cosine ramp that is defined as

$$\Delta\alpha_t = 0.5\,\alpha_f \left(1 - \cos\left(\frac{t}{t_{rot}}\pi\right)\right). \tag{7.1}$$

The full rotational speed of the mixer is obtained after approximately half a rotation at $180\Delta t$. From this time step on, the angle increment is set to $\alpha_t = \alpha_f = 2°$. The angle increment $\Delta\alpha_t$ plotted over the time steps can be seen in Fig. 7.4. An abrupt application of the tractions can lead to strong oscillations, which may cause a convergence failure in the implicit coupling. Therefore, also the tractions that are transmitted to the structural solver by *comana* are multiplied by a scale factor β_{tr}, computed as

$$\beta_{tr} = 0.5\,\left(1 - \cos\left(\frac{t - t_{st}}{t_e}\pi\right)\right). \tag{7.2}$$

Fig. 7.4 shows the traction scale factor β_{tr}. After the full rotational speed is reached, the simulation is continued for one rotation or 180 time steps before the first tractions are transferred to the structural solver. This dampens out vibrations that appear due to the rotational acceleration of the mixer blades and, hence, avoids distorted panels in the BEM solver. A quadratic extrapolation (see Sec. 4.2) is used to initialize the displacements at the beginning of a time step, and the QNLS method explained in Sec. 4.5.3 is used as the

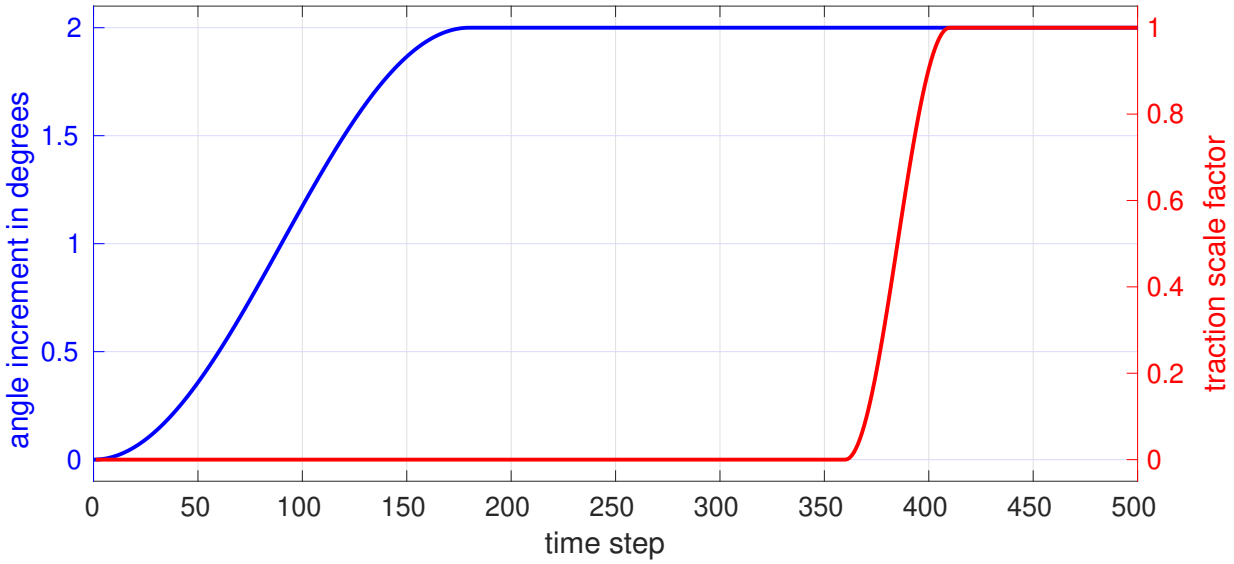


Figure 7.4: Angle increment and traction scale factor at the startup of the simulation.

convergence acceleration method. For the interpolation between the structural and the fluid mesh, mesh-based interpolation techniques as explained in Sec. 4.3.4 are employed.

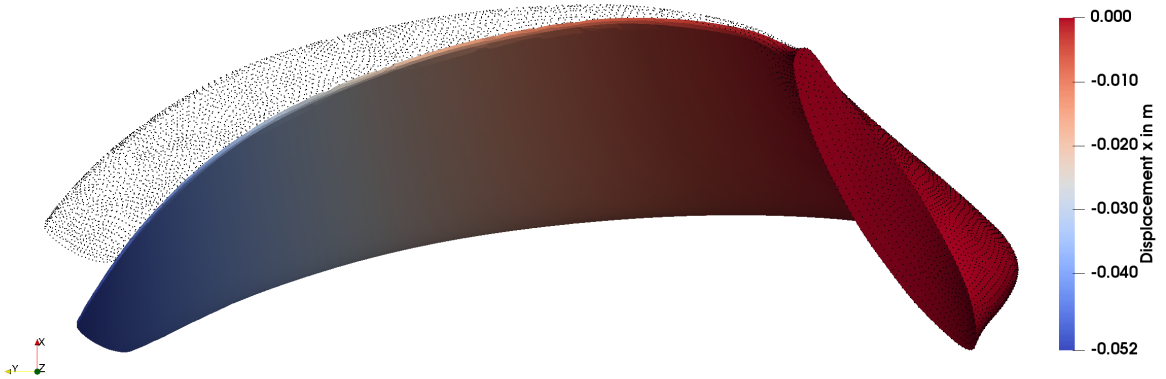
A problem that is apparent in the FSI simulation of the presented case are singular panels located close to the tip of the propeller blade in the fluid simulation. These singular panels are caused by the strong deformation of the blade, which leads to small distorted panels on the tip. The pressure computed by the fluid solver in these panels exceeds realistic values by several magnitudes. To reduce the effect of the distorted panels, the mesh shown in Fig. 7.2 was already optimized in the distribution of the panels. Since the effect is still apparent in some time steps, the traction values, which are equal to the pressure values corrected by frictional effects, are limited to a value of $2 \cdot 10^6$ Pa in *comana* before they are forwarded to the structural solver. With this filter, it is possible to avoid unrealistically high forces acting on small areas of the structural elements, which could lead to highly distorted elements in the structural solver.

These singular panels are only apparent in FSI simulations since the panels in the uncoupled *pan*MARE simulation do not change their shape.

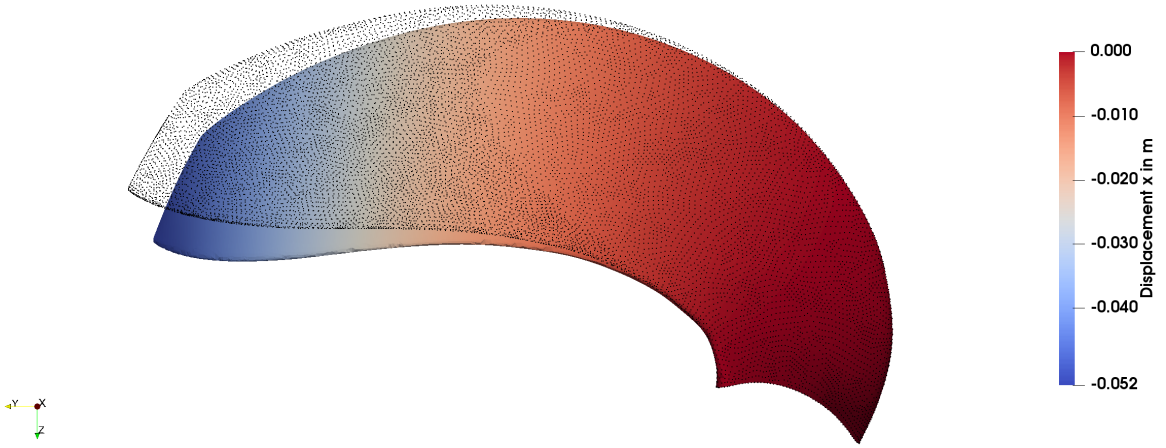
7.3 Simulation results

To visualize the deformation of the blade, the displacements caused by the deformation of the submersible mixer are shown in Fig. 7.5. The color scale only visualizes the displacement in the x-direction due to the deformation. The rigid body motions due to translation and rotation are excluded in this figure. The dominating deformation is the deflection of the blade in the x-direction. Nevertheless, Fig. 7.5b also shows a deformation around the rotation axis of the mixer. At the tip of the blade, the maximum displacement in the x-direction is approximately equal to 0.052 m. The deformation of the blade is especially strong further away from the hub in the region which is not reinforced by the inner metal ring. Validation based on tip displacement is not possible as it was not measured in the

experiment. Despite this, the visualization of the deformation provides insight into the importance of considering FSI effects for flexible materials.



(a) Side view.



(b) Top view.

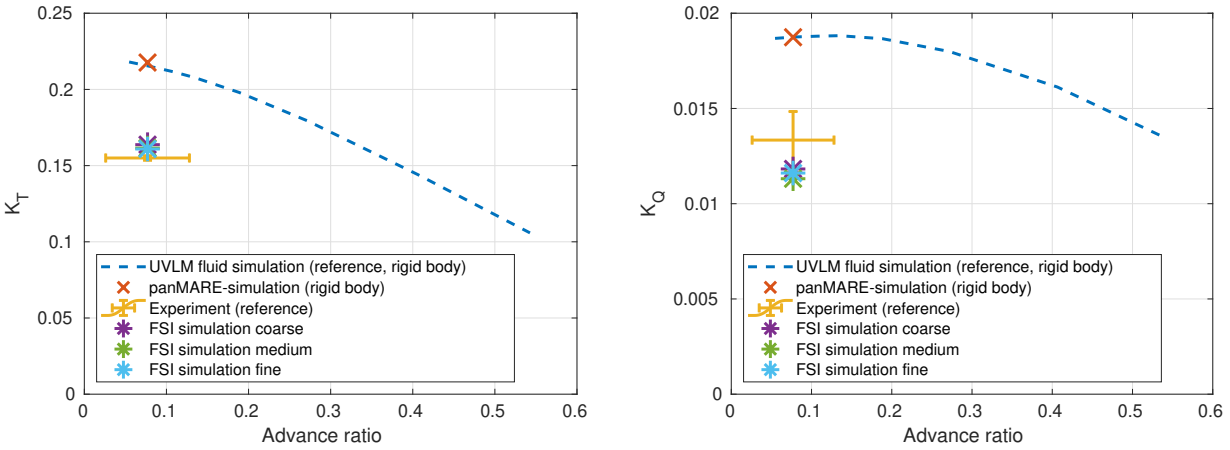
Figure 7.5: Deformation of the mixer blade after 4.75 rotations. The black dots indicate the location of the nodes of the undeformed blade.

For the validation, the thrust and torque coefficients computed in the FSI simulation presented in this work are compared in the next part to the values obtained by Lass [20] in the experimental tests. Lass did not only perform experimental measurement but also numerical simulation with the unsteady vortex lattice method (UVLM). The UVLM method is also based on the potential theory, but compared to the BEM used in this work, the UVLM neglects the thickness of the blade.

For the validation of the FSI simulation, variables that represent the whole simulation are beneficial. Furthermore, it is favorable to validate the variables that are important for the analysis conducted after the validation. In the optimization that follows in the next chapter, efficiency (see Eq. (3.57)) and thrust (see Eq. (3.55)) are two of the most important variables for the optimization. Since the efficiency is also dependent on the torque coefficient (see Eq. (3.56)) and both variables are available from the experimental

measurements, the first step of the validation is performed by comparing the thrust and torque coefficients.

The thrust for the FSI simulation in this work is computed in *pan*MARE by summing up the forces computed at each panel center in the x-direction. The corresponding torque is computed similarly by computing the contribution to the torque of each panel and summing up the resulting torque values. After computing the thrust and the torque, Eqs. (3.55)-(3.56) are used to compute the thrust and torque coefficients. The problem of the small distorted panels (mentioned at the end of the previous section) also causes deviations in the trust and torque coefficients for some time steps. To reduce this effect of the distorted panels, the thrust and torque coefficients are averaged over two full rotations after outliers are excluded by a filter.



(a) Thrust coefficient. **(b)** Torque coefficient.

Figure 7.6: Validation results for the submersible mixer. The reference results taken from Lass [20] are marked with 'reference' in the legend.

Fig. 7.6 shows the resulting thrust and torque coefficient. In this figure, the blue dashed lines indicate the rigid body UVLM fluid simulation performed by Lass [20, p. 93]. The red cross indicates the rigid body simulation performed in *pan*MARE. Obviously, it is not possible to directly validate the rigid body *pan*MARE simulation based on the UVLM approach, as both results are obtained with numerical methods and both methods are based on the potential theory. In addition to this, the UVLM approach further simplifies the problem since the thickness of the blade and frictional effects are not considered. Despite this, matching results between both methods are an indicator of accurate fluid simulations with both methods and can be seen as a form of verification. Fig. 7.6a shows that the thrust coefficient computed in *pan*MARE is only 1 % higher than in the UVLM method.

The FSI simulations, performed with the three different structural meshes introduced in the previous section, are labeled as coarse, medium, and fine in the legend and marked with an asterisk in the graph. The difference in the thrust coefficient between the coarse and the fine mesh is approximately 1.7 %. The difference in the thrust coefficient between the medium and the fine mesh is approximately 0.4 %. Due to the small difference between the fine and the medium resolution, the two corresponding asterisks are almost indistinguishable.

The experimental results are marked with error bars that indicate the systematic measurement uncertainties mentioned in Sec. 7.1. The systematic measurement uncertainty

for the thrust is relatively low (0.2 % of the result) [20, p. 65]. The difference between the thrust measured in the experiment and the thrust computed with the fine FSI simulation is approximately equal to 3.8 %. Even though the measurement uncertainty for the thrust is small, the measurement uncertainty of the inflow velocity is high. The vane anemometers have a systematic measurement uncertainty of up to 2 % of the maximal measurable value ($5\,\frac{\mathrm{m}}{\mathrm{s}}$) [20, p. 65]. For the low averaged inflow velocity of $0.28\,\frac{\mathrm{m}}{\mathrm{s}}$, this leads to an uncertainty range for the velocity of approximately $0.18-0.38\,\frac{\mathrm{m}}{\mathrm{s}}$. In addition to this, the inflow velocity is assumed to be constant in the FSI simulation, which can lead to differences in the results. Lass computed the uncertainty range for the advance ratio as $J = 0.08 \pm 0.051$ [20, p. 76]. This range is also indicated in Fig. 7.6. In view of this, the differences between the FSI simulation and the experiment are well within the expected limits.

Fig. 7.6b visualizes the results for the torque coefficient in the same way as Fig. 7.6a. Again, the difference between the two rigid body simulations is very small. However, for the experimental results, a high difference of approximately 13 % is visible between the fine FSI simulation and the experiment. Despite the considerable differences, the FSI simulation results are still close to the tolerance range of the experiments since the systematic measurement uncertainties for the torque measurements are 10 %.

Overall, the results of the torque and thrust coefficients for the FSI simulations show a good agreement with the experiments. The differences between the experimental results and the simulations are within the expected range.

The large difference of approximately 26 % for the thrust coefficient between the *pan*-MARE rigid body results and the FSI simulation results emphasizes the importance of considering FSI effects for the presented case.

To verify that not only global variables but also local variables are computed correctly in the FSI simulation, the local strains measured in the experiment with strain gauges are compared to the strains computed in the FSI simulation.

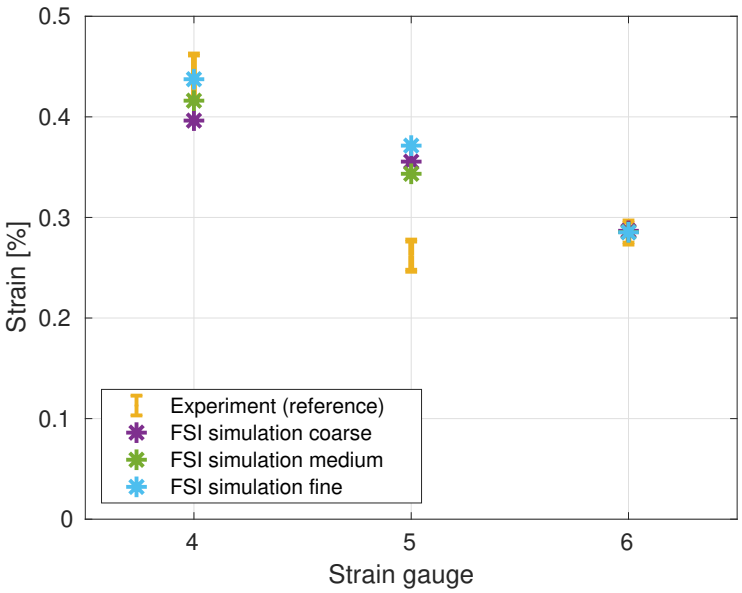
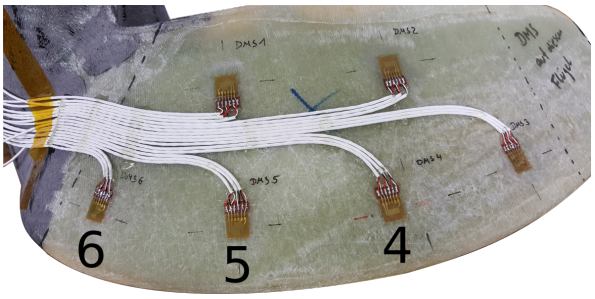


Figure 7.7: Comparison of strains between the experiment and the FSI simulations.

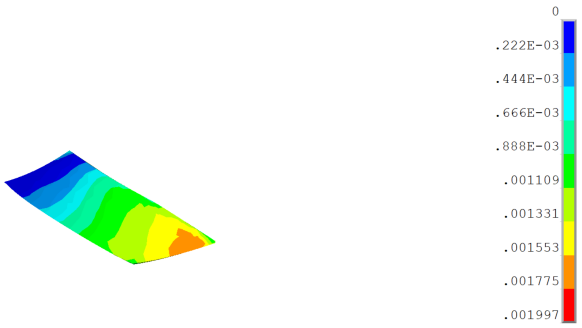
To compare the strains in the experiment with the strains in the FSI simulation, the von Mises equivalent strain according to Eq. 2.71 is used. In ANSYS, a corotational formulation

is used to account for the large rotations of the propeller blade. With this corotational formulation, it is possible to use the engineering strain tensor to compute the von Mises equivalent strain. Since the strain in the experiment is measured with strain gauge rosettes, only the strain state tangential to the surface can be measured directly. Since the strain in the surface normal direction cannot be determined in the experiment, Lass [20, p. 68] used an equation from Keil [109, p. 292] which is based on the plane stress condition to compute the full von Mises equivalent strain.

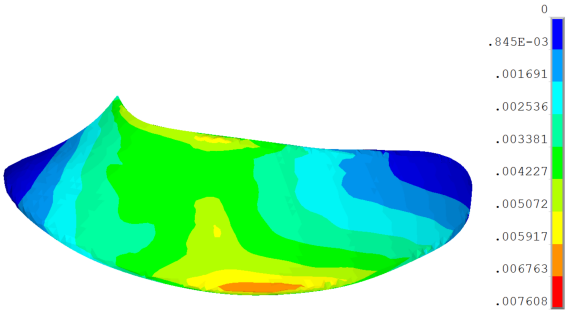
Fig. 7.7 shows a comparison of the strain in the experiment and the strain computed in the FSI simulations. The three strain gauges on the GFRP layer can be seen in Fig. 7.8a (before the polyurethane resin layer is molded on this layer).



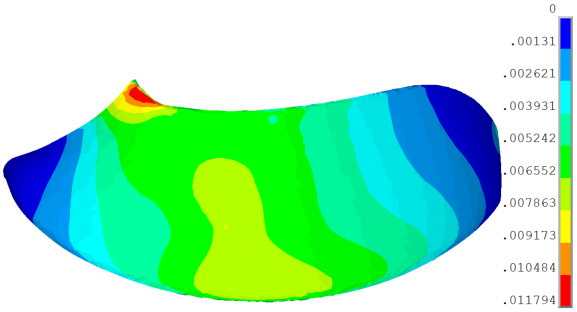
(a) Strain gauge locations in experiment [20, p. 67].



(b) Strain on the metal reinforcement.



(c) Strain on the GFRP layer.



(d) Strain on the polyurethane layer.

Figure 7.8: Visualization of the strain together with the strain gauge locations.

In Fig. 7.7, it can be seen that the results in the experiments agree very well with the FSI results for strain gauge 6. The results for strain gauge 4 are also in good agreement with the results from the fine FSI simulation. For strain gauge 5, a significant deviation between the experimental results and the FSI simulation results is visible.

Verifying the FSI simulation with the mesh convergence study shows that the results for the three different mesh resolutions agree well for strain gauge 6. For strain gauge 5, the results for the coarse and the fine mesh resolution are closer together than the results for the medium and fine resolution. This shows that the convergence behavior is not optimal. However, since the FSI simulation contains many numerical uncertainties, an ideal convergence behavior was not to be expected. Also, the results for all three mesh resolutions are close to each other – which shows that the numerical error due to the mesh is small. The results for strain gauge 4 indicate a better convergence behavior than the results for strain gauge 5.

The numerical simulations allow for a visualization of the strain at every location. Fig. 7.8 provides an overview of the strain on the surface of all layers together with the strain gauge location on the GFRP layer in the experiment. The maximum overall strain is experienced by the polyurethane layer since it is also the most flexible layer (see Tab. 7.1). On the polyurethane layer, the maximum equivalent strain is below 1.2 %. With this, the assumption of a linear elastic material model is valid. When comparing Fig. 7.8b-7.8d, it shows that the metal reinforcement leads to a maximum of strain (and with this also to a maximum of stress) on the GFRP layer in the spanwise region in which the metal reinforcement ends. Strain gauge 5 is located in this region as well, which could explain the deviation between the FSI simulations and the experiments. The region around the strain maximum shows high gradients in the strain. The experiments might exhibit an uncertainty concerning the exact location of the strain gauges – which could, together with the high gradients, explain the differences for strain gauge 5 between the experiment and the FSI simulation.

8 Ship propeller

The ship propeller is the last and most comprehensively analyzed application case in this work. As for the previous application case, the simulations for ship propellers were conducted in the scope of the project "Hydroelastic simulation of the acoustic behavior of ship-propeller configurations with and without cavitation", funded by the German Research Foundation.

The KRISO container ship (KCS) propeller KP505 [110] was chosen as the first model for a ship propeller in this chapter. The following section serves to introduce the simulation setup for the KCS propeller in more detail. The influence of the Young's modulus on the dimensionless propeller parameters at different advance ratios is investigated. Also, this is underpinned by analyzing the influence of different coupling parameters such as convergence tolerances or convergence acceleration methods on the computational time, the necessary number of coupling iterations, and the solution accuracy.

In the following section, a new propeller is developed based on the KCS propeller. In the first step, only the propeller geometry is optimized for a rigid propeller in order to combine the targets of maintaining the thrust, keeping the efficiency high, and avoiding cavitation and cavitation variation. Besides leading to damage (e.g. through cavitation erosion [111, p. 245]), cavitation is also the primary reason for noise caused by a ship propeller. After this, an FSI simulation model for a flexible fiber-reinforced propeller is developed to improve the operational properties of the propeller further. The influence of the fiber direction and the propeller stiffness on the propeller's performance parameters is investigated. In the last step, the propeller geometry is optimized in conjunction with fiber direction and overall stiffness.

8.1 KCS propeller

The KRISO container ship (KCS) propeller KP505 [110, 112] (referring to the "Korea Research Institute of Ships and Ocean Engineering") is a 5-bladed propeller with a diameter of 7.9 m. The KCS propeller is chosen here since it is a widely used design for scientific analysis – even though it was never used or built in full scale. As a result, there is a significant amount of experimental and simulation data available for the KCS propeller.

8.1.1 Simulation setup

The FSI simulation setup for the KCS propeller is similar to the simulation setup introduced in the previous chapter. The fluid simulation is conducted with *pan*MARE, and the structural simulation is conducted with ANSYS, while the coupling is realized with *comana*. The parameters for the simulation are listed in Tab. 8.1. The ramps for the startup are implemented in the same way as in the simulation presented in the previous chapter in Sec. 7.2. The time-step size is modified to keep the angle increment per time step to

2° when full rotational speed is reached. The settings for extrapolation and convergence acceleration are chosen in the same way as in the previous application case (see Sec. 7.2). Also, the filters introduced in the previous chapter are used with the same settings to avoid singularities due to distorted panels. The KCS propeller is simulated in full scale with a rotation rate of 1.5 rotations per second.

Table 8.1: Parameters for the open water FSI simulation of the isotropic KCS propeller.

parameter	value
rotation rate	$1.5\frac{1}{\mathrm{s}}$
angle increment $\Delta\alpha_f$	$2°$
time-step size Δt	$\frac{1}{270}\,\mathrm{s} \approx 0.003704\,\mathrm{s}$
time of full rotational speed t_{rot}	$180\Delta t \approx 0.66\,\mathrm{s}$
wake rollup enabled t_w	$370\Delta t \approx 1.37\,\mathrm{s}$
traction ramp start t_{st}	$540\Delta t \approx 1.33\,\mathrm{s}$
traction ramp end t_e	$590\Delta t \approx 1.52\,\mathrm{s}$
total simulation time t_e	$1800\Delta t \approx 6.66\,\mathrm{s}$

The structural mesh for this simulation is not constructed with tetrahedral elements but with linear hexahedral elements. The mesh is generated with an in-house meshing tool based on the surface geometry of the propeller. Fig. 8.1 displays the structural mesh of the KCS propeller for the FSI simulations in this section.

B-splines [113, p. 486] describe the blade surface – which is distinguished into the hub surface, the tip surface, the pressure side surface, and the suction side surface. The tip of the propeller blade is cut off to avoid poorly shaped elements close to the tip. The B-splines are used to create two structured two-dimensional surface meshes on the suction- and pressure-side surfaces. These two meshes have the same number of elements in the span and chord directions. To create a three-dimensional volume mesh based on these two surface meshes, an interpolation between the nodes of both surface meshes is performed in which the layers with the intermediate nodes are inserted. Only one layer with intermediate nodes is inserted for the mesh shown in Fig. 8.1. The structural mesh has 20 elements in the span direction, 10 elements in the chord direction, and 2 elements in the thickness direction – leading to a total of 400 elements for the structural part of this FSI simulation. As in the previous chapter (see Sec. 7.2), the rotational movement of the propeller is realized by prescribing the displacements at the hub of the structural simulation, and the Newmark method is used as implicit time integration method.

This meshing approach leads to degenerated elements on the leading and trailing edge of the propeller blade, as can be seen in Fig. 8.1. The degenerated elements, being prism-shaped, are fully supported by ANSYS for linear hexahedral elements [90, p. 730]. The benefit of creating a layered structural volume mesh is that the different layers of the anisotropic KCS propeller, which will be introduced later, can be represented more accurately with this meshing approach.

Fig. 8.2 shows the fluid mesh for the FSI simulation of the KCS propeller. The suction and the pressure surface both have 18 panels in the chord direction and 25 panels in the span direction. In total, this leads to approximately 900 panels. To better capture the

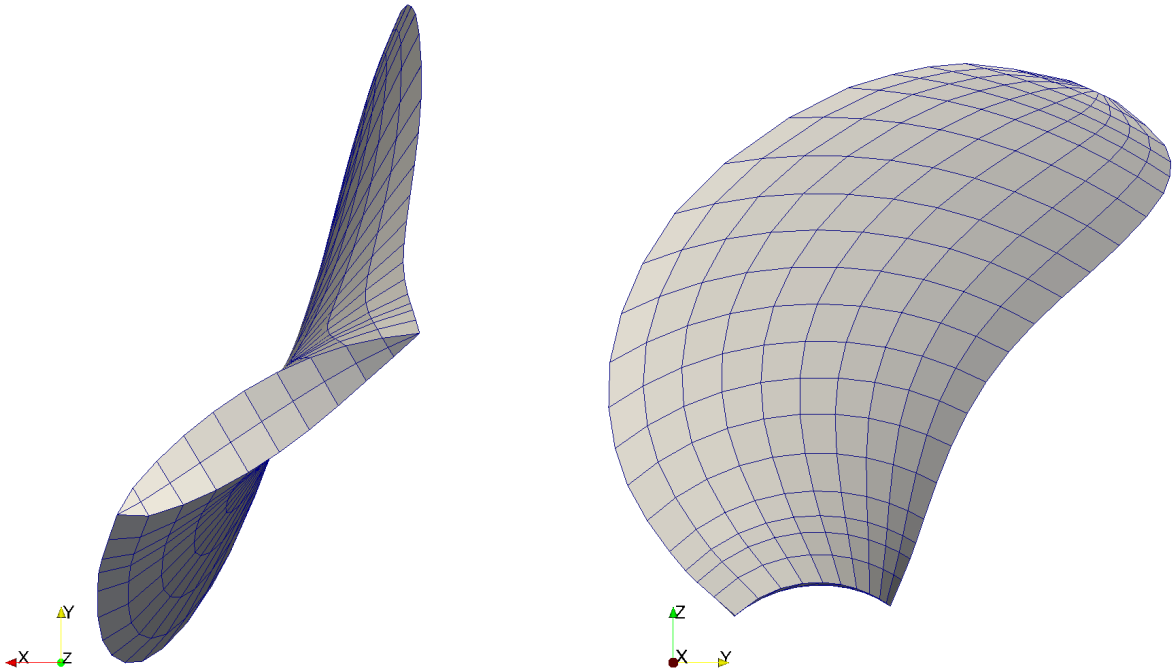


Figure 8.1: Structural mesh for the KCS propeller.

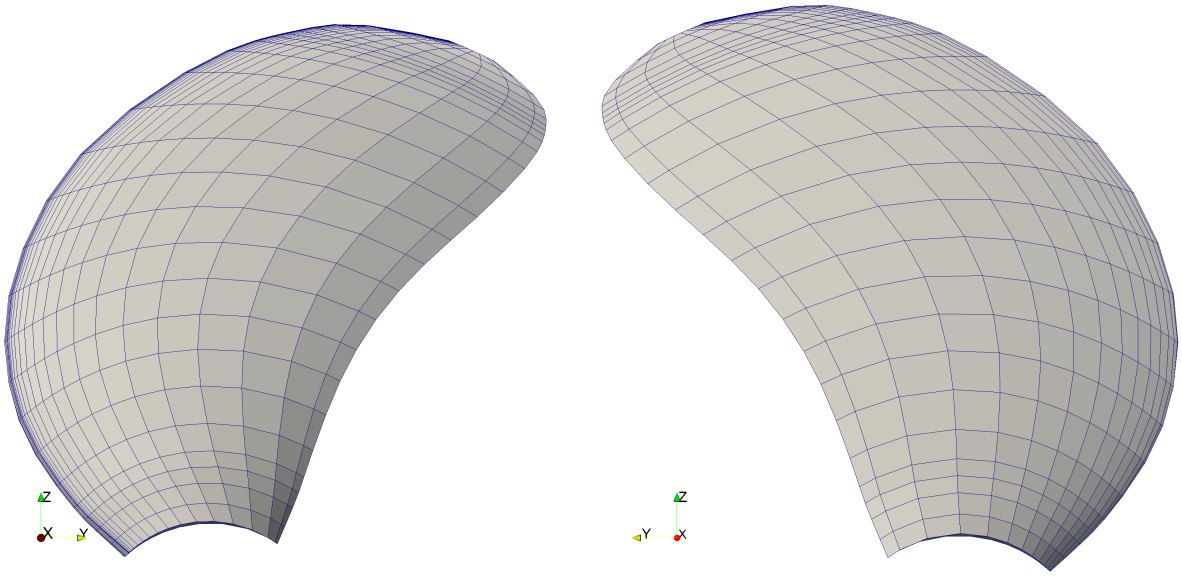
strong pressure gradients at the leading edge, a refinement in chord direction is introduced there. *pan*MARE is able to simulate the wake roll-up in the fluid simulation. To this end, it is helpful to include a refinement towards the tip in the span direction, to capture the wake roll-up more accurately. More details about the implementation of the wake roll-up realized through a wake deformation can be found in [114, p. 22].

The fluid simulation setup presented here was further validated on a model scale propeller with a geometric scale ratio of $2.639 \cdot 10^{-2}$. The experimental results for the validation stem from Otzen et al. [115], and the results of the validation are published in Lund et al. [83, p. 5 f.]. Gonzalés et al. [116] performed another validation of the fluid-structure interaction setup based on the experimental results of Savio et al. [117].

8.1.2 Numerical errors due to rotational movement

A problem in the FSI simulation that became apparent in the first ship propeller simulations is an overestimation of approximately 1.5% to 2% of the thrust and torque coefficient in the coupled FSI simulation with a rigid propeller compared to an uncoupled fluid simulation in *pan*MARE. This overestimation is reduced if the time step (and, correspondingly, the angle increment) is smaller. This is the very same effect that was already observed and briefly explained in Sec. 5.4 for the rotor of the FWT. To find a solution to this problem, the effect is analyzed in more detail here.

The reason for the different results between a coupled FSI simulation with a rigid propeller and the uncoupled *pan*MARE simulation is that the velocity and acceleration at the panel's nodes in *pan*MARE are computed differently in these two setups. In *pan*MARE,



(a) Pressure surface. **(b)** Suction surface.

Figure 8.2: Fluid mesh for the KCS propeller.

the rigid body motion of a propeller rotating around an axis is computed with a motion model that computes the velocity and the acceleration based on the prescribed rigid rotational movement at every time step. The velocities for a rotational movement are therefore computed tangential to a circle, as the black arrow in Fig. 8.3 indicates. If the displacements are set by *comana*, the only information available to the fluid solver is the displacements at the nodes. At this point, the rotational components of this motion cannot be known to the fluid solver. The computation of the velocities and the accelerations in the fluid solver can, therefore, only be performed based on the combination of the displacements of the current and the previous time steps – similar as in Eq. (2.50) and Eq. (2.52). In Fig. 8.3, the resulting computed velocity $v(t)$ for a purely rotational movement is indicated with a blue line.

The computed velocities have a delay of half a time step, due to the fact that they are the averaged velocities between the current and the previous time step. Similar to the velocities, the accelerations are not computed accurately enough either. The deviation caused by this effect increases strongly with larger time steps and correspondingly larger angle increments, as can be seen in Fig. 5.11, where the deviation in thrust is around 20% for an angle increment of around 10°. In contrast to the rigid propeller, it is not possible to describe the displacements of the deformable propeller in the FSI case purely as rigid body modes.

A corotational formulation, in which the rotational displacements are treated separately from the displacements due to the deformation, overcomes this problem. The displacements that are forwarded from *comana* to *pan*MARE are only the displacements due to the deformation of the blade. To this end, the rigid body modes are subtracted from the displacements in *comana*. In *pan*MARE, the velocities are now computed as a combination of the velocities captured accurately with the motion model and the velocities due to

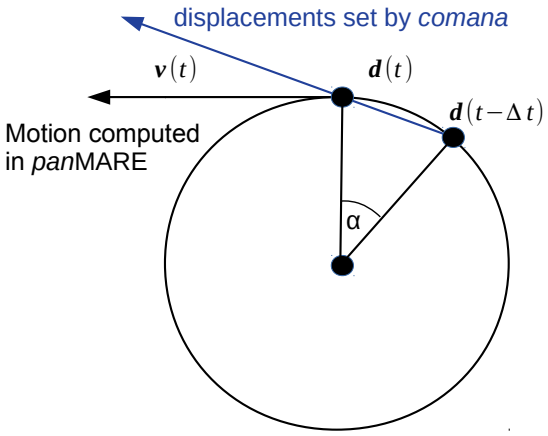


Figure 8.3: Numerical error due to the approximation of the velocity.

the deformations from *comana*. Analogously, the accelerations are computed in the FSI simulation. With this modification, the error due to the differing computation of the velocities and the accelerations is reduced significantly without increasing the computational time.

8.1.3 Stiffness variation

To investigate the influence of the flexibility of the blade on the performance of the propeller, a parameter study is performed with different Young's moduli for the propeller blade. This is done to investigate the effect of the flexibility of the blade on the thrust coefficient, the torque coefficient, and the efficiency (computed according to Eqs. (3.55)-(3.57)). Since the number of revolutions per second f, the diameter D, and the density are the same for all simulations, a change in the thrust coefficient or the torque coefficient is always due to a different thrust or torque acting on the propeller in the simulation. The density is set to $7750\frac{\mathrm{kg}}{\mathrm{m}^3}$, and the Poisson's ratio is set to 0.3.

In this parameter study, also the advance ratio is varied between 0.4 and 1.0 in steps of 0.1 by modifying the translational velocity in the flow direction. This is equivalent to modifying the inflow velocity in Eq. (3.54). The fluid simulation in this section is performed in open water conditions. Therefore, the loading on the blades and the resulting deformation is constant over time. A different density of the propeller material should thus not have a significant impact on the result. In order to select a plausible range for the Young's modulus of a ship propeller, the different materials typically used for propellers are introduced in the next paragraph.

Ship propellers are typically made of copper alloys (such as brass and bronze), cast iron, and cast steel [118, p. 426]. Also, CFRP propellers have become increasingly popular in recent years, since they are lighter and have better acoustic properties [118, p. 432]. Typically, the Young's modulus of copper alloys lies in the region of 100 GPa − 150 GPa. Cast steel and iron have a Young's modulus of around 200 GPa.

Thermoplastic polymers, which are often used as the matrix material for fiber-reinforced polymers, usually have a Young's modulus in the range of 2 GPa to 5 GPa range, see [119,

p. 38] [120, p. 10]. The Young's modulus of carbon fibers can be even higher than the Young's modulus of steel [120, p. 16]. The Young's modulus of CFRPs and other fiber-reinforced polymers, which is strongly anisotropic and depends on the exact composition and manufacturing process, usually falls in the range of 5 GPa to 100 GPa [119, p. 58] [120, p. 10].

As a result of this analysis, the Young's modulus of the propeller blade in the structural simulation is set to 2 GPa, 5 GPa, 10 GPa, 20 GPa, 50 GPa, 100 GPa, 200 GPa, and rigid (uncoupled fluid simulation in *pan*MARE). Fig. 8.4 shows the results for the thrust coefficient, the torque coefficient, and the efficiency for these simulations for an advance ratio between 0.4 and 1.0.

The first observation is that, for a Young's modulus of 100 GPa and 200 GPa, the plots are almost indistinguishable from the simulation of the rigid propellers. The commonly used approach for fluid simulations of ship propellers to assume the propeller is rigid therefore seems to be justified for metal propellers.

For the more flexible propellers with a Young's modulus down to 10 GPa, the thrust is slightly increased for low advance ratios and slightly decreased for high advance ratios. The propeller with a Young's modulus of 10 GPa already shows a reduction in thrust for an advance ratio of 0.4. For an advance ratio of 0.8, the thrust is approximately the same for all Young's moduli down to 5 GPa. For the torque coefficient, similar observations can be made. The torque for the flexible propeller with a Young's modulus of 10 GPa compared to one of the stiff propellers is around 9% higher for a low advance ratio of 0.4. The resulting thrust for the same comparison is only around 4% higher. In combination, this results in a more than 4% lower efficiency for the propeller with a Young's modulus of 10 GPa compared to the stiff propellers at low advance ratios.

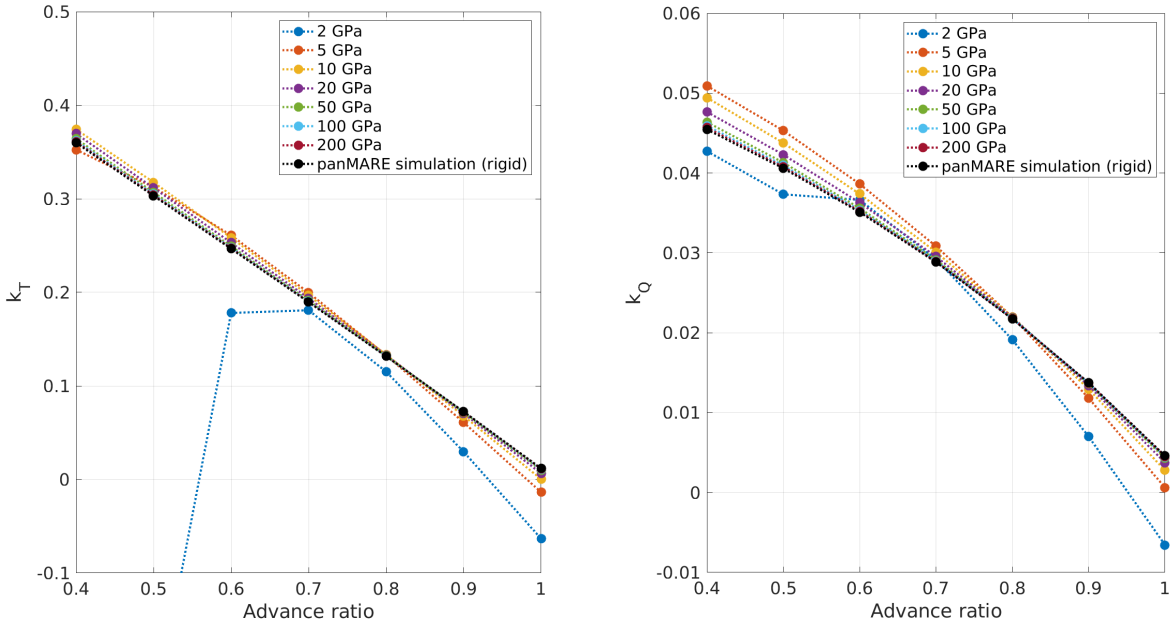
Surprisingly, almost all propellers perform quite similarly at an advance ratio of 0.8. At a higher advance ratio of 0.9, a decrease in all coefficients can be observed for higher flexibility. For an advance ratio of 1.0, the more flexible propellers are unable to generate positive thrust for the given setup, and efficiency is significantly reduced with higher flexibility for those propellers that still generate thrust.

The propeller with a Young's modulus of 2 GPa (corresponding to a propeller made of a flexible polymer) works reasonably well at an advance ratio of 0.7 and 0.8, but is still worse in this region than all the other stiffer propellers. For lower and higher advance ratios, the propeller becomes more and more inefficient. The propeller is unable to generate thrust below an advance ratio of 0.6.

In the next step, the deformation of the propeller is analyzed. Tab. 8.2 shows – for a point at the top – the magnitude of the displacement vector (computed as L^2-norm) from which the rigid body rotation and translation have been subtracted. This point marks the intersection of the leading edge with the tip surface. For most simulations, the deformation is maximal at this point.

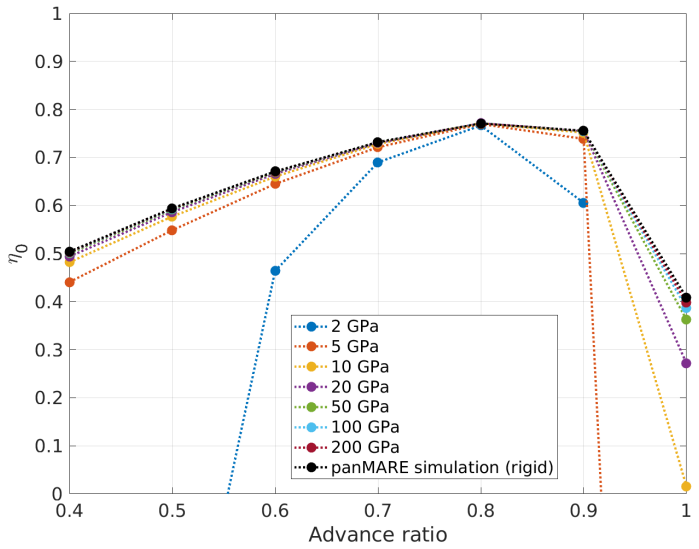
The stiff propeller with a Young's modulus of 200 GPa shows a deformation of less than 10 mm even for the lowest advance ratio. For the stiffer propeller with a high Young's modulus, the deformation is almost proportional to the inverse of the Young's modulus – indicating that the force acting on the propeller does not change significantly in this range, which concurs with the observation that the thrust coefficient does not change significantly in this range either.

Combining the information from Tab. 8.2 with the performance of the propeller shown in Fig. 8.4, suggests a significant loss of efficiency and thrust at low advance ratios, which



(a) Thrust coefficient.

(b) Torque coefficient.



(c) Efficiency.

Figure 8.4: Non-dimensionalized coefficients in the FSI simulation of the KCS propeller for difference advance ratios and Young's moduli.

Table 8.2: Displacement magnitude due to deformation in millimeter at the tip.

Young's modulus	J=0.4	J=0.5	J=0.6	J=0.7	J=0.8	J=0.9	J=1.0
2 GPa	847.9	719.2	577.3	421.3	250	70.8	110.2
5 GPa	383.2	317.4	251.9	187.4	124.4	62.3	10.4
10 GPa	190.3	156.3	124	91.6	62.6	35.3	10.3
20 GPa	92.2	75.5	59.8	45.1	31.4	18.5	6.6
50 GPa	36.1	29.6	23.5	17.8	12.5	7.6	3
100 GPa	18	14.7	11.7	8.9	6.3	3.8	1.5
200 GPa	8.9	7.4	5.8	4.4	3.1	1.9	0.8

is only visible when the KCS propeller is deformed several hundred millimeters at the tip. Another observation is that even though the dimensionless coefficients remain almost unchanged at an advance ratio of 0.8, the displacement at the tip increases significantly with lower Young's moduli.

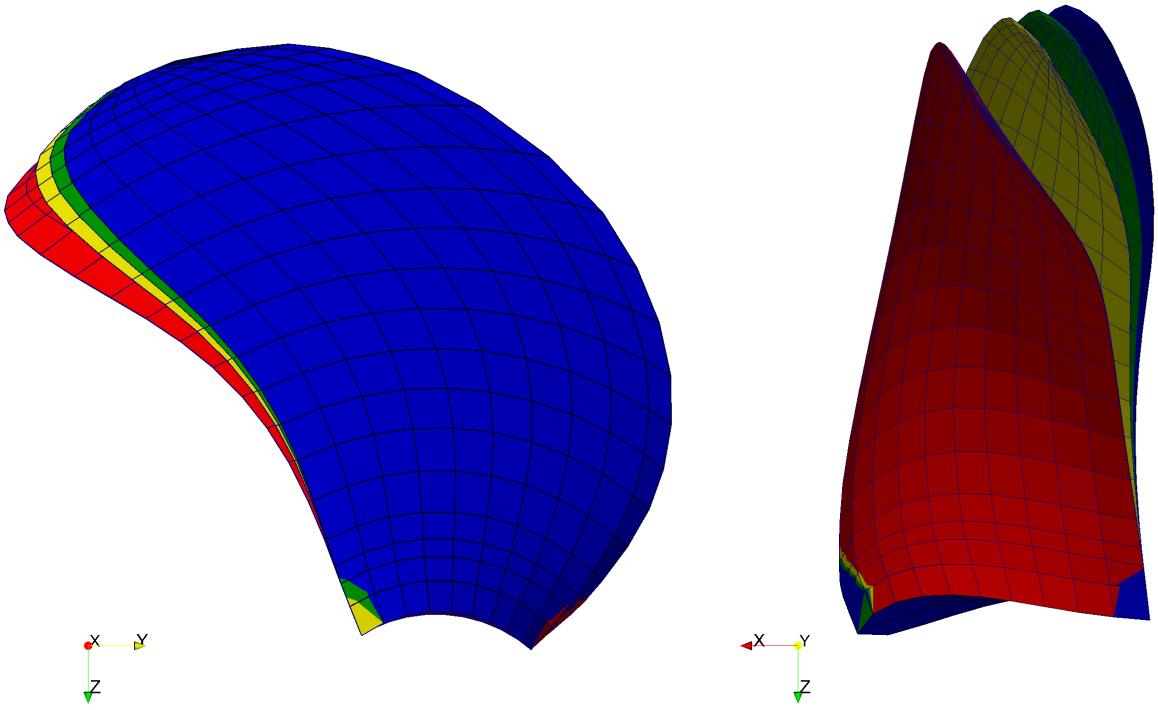


Figure 8.5: Unscaled deformation of the KCS propeller at an advance ratio of 0.4 for different Young's moduli (blue: 200 GPa, green: 10 GPa, yellow: 5 GPa and red: 2 GPa).

To visualize the deformation at an advance ratio of 0.4 for different Young's moduli, the unscaled deformation of one propeller blade is shown in Fig. 8.5. Only the more flexible propellers are shown there, along with the stiffest propeller, since the deformation for the stiffer propellers is very low. The propeller with a Young's modulus of 200 GPa, shown

in blue, serves as a reference. The Young's moduli are, respectively, 10 GPa for the green propeller, 5 GPa for the yellow propeller, and 2 GPa for the red propeller.

The propellers appear to undergo more pronounced deformation with higher radii. The only propeller that loses its initial shape completely is the most flexible propeller. This loss of shape might also explain why the propeller is unable to generate thrust for low advance ratios. The deformation direction of the propeller blade is most pronounced in the x-direction (axial direction) but there are also significant displacement components in the other two directions. For example, the direction of the vector of deformation for the yellow propeller (Young's modulus 5 GPa) is $[0.8\ -0.5\ 0.34]^T$. For most of the other propellers (except for the most flexible propeller), this direction is barely changed with varying Young's moduli at low advance ratios.

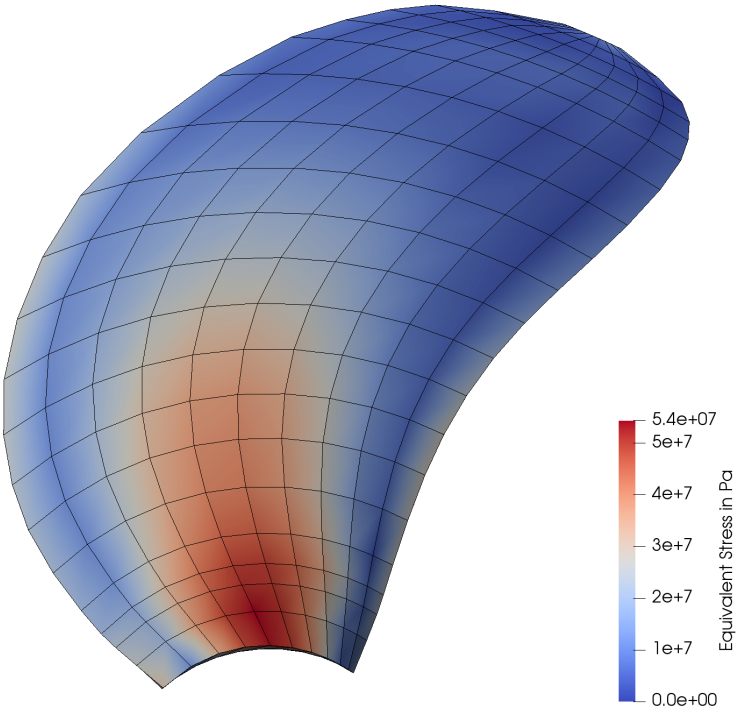


Figure 8.6: Equivalent stress of the KCS propeller (advance ratio 0.4, Young's modulus 200 GPa).

Here, the FSI simulation offers another benefit, namely that the von Mises equivalent stress according to Eq. 2.68 can be computed directly in the structural solver. With this, it is possible to test new propeller designs for structural stability also under realistic inflow conditions. A visualization of the von Mises equivalent stress for the stiffest propeller for an advance ratio of 0.4 is shown in Fig. 8.6. With around 54 MPa, the maximum stress is in an uncritical range for most materials.

Important criteria that have to be set correctly to obtain a good balance between solution accuracy and computational time are the tolerances in the absolute convergence criterion (according to Eq. 4.18) and the relative convergence criterion (according to Eq. 4.19).

Tab. 8.3 lists the influence of the chosen tolerances on the number of coupling iterations and the computation time for a simulation case with a Young's modulus of 5 GPa and with an advance ratio of $J = 0.4$. This table also lists the differences in the thrust coefficient k_T and the torque coefficient k_Q of the two higher tolerances compared with the low tolerance (which is the reference here).

Table 8.3: Influence of convergence criteria tolerances on computation time, number of coupling iterations, and difference in the thrust coefficient k_T and the torque coefficient k_Q (compared to the reference).

tolerance	high	middle	low (reference)
abs. tol. ϵ_a	10^{-6}	10^{-10}	10^{-14}
rel. tol. ϵ_r	$5 \cdot 10^{-3}$	$5 \cdot 10^{-7}$	$5 \cdot 10^{-11}$
average coupling iterations	6.9	10.4	14.2
computation time	5.6 h	7.8 h	11.1 h
Difference to reference k_T	0.07%	0.03%	-
Difference to reference k_Q	0.06%	0.04%	-

The tolerances are set in such a way that the factor between the high and the middle tolerance as well as the factor between the middle and the low tolerance is 10^4. The simulations are (like the previous propeller simulations) conducted on the high-performance cluster (HPC) of TUHH. The computation time is increasing approximately linearly with the necessary number of implicit coupling iterations (averaged over the entire simulation).

The low tolerance is chosen as a reference to compute an approximation for the numerical error due to the convergence criterion. If the tolerance is set to low, the computation time and the necessary number of implicit coupling iterations are approximately twice as high as for the high tolerance. Despite the much higher computation time, the error lies in the region of below 0.1%. Since this numerical error is insignificant compared to other discretization and modeling errors, high tolerance is used for the subsequent propeller simulations.

8.1.4 Comparison of convergence acceleration methods

The setup introduced at the beginning of this chapter is now modified to a simulation setup with a more realistic inflow to the propeller. Based on this setup, the convergence acceleration methods introduced in Sec. 4.5 are now compared in different stages of the simulation, with regard to different Young's moduli and different convergence tolerances.

An inhomogeneous ship wakefield is used as the inflow field to capture the varying forces acting on the propeller blades during the loading cycle. The symmetric inflow field shown in Fig. 8.7 is used to capture the varying forces acting on the propeller blades during a realistic loading cycle. This inflow condition was computed based on CFD simulations around the hull of a ship. The central area is omitted as the propeller hub is not modeled in the presented simulations. For the comparison of the different convergence acceleration methods, this setup with varying forces is more suitable – since, otherwise, the forces acting on the propeller blades are almost constant (after the run-in period) with only a change in the direction of the forces due to the rotation.

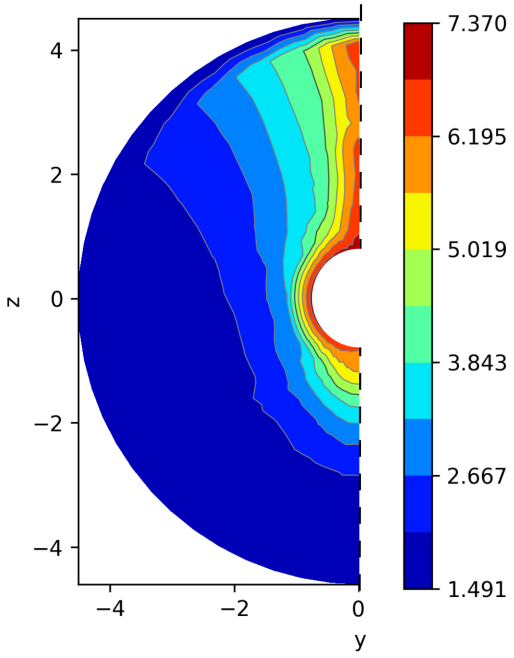


Figure 8.7: Wake field (inflow to propeller).

Most of the other settings remain unchanged in comparison with the previous section. The general simulation parameters are listed in Tab. 8.4. The run-in period is shortened here to save computational time and to achieve a shorter total simulation time in comparison with the previous ship propeller FSI simulations.

The unmodified advance ratio in the simulation is now set to $J = 0.896$. According to Eq. (3.54), this leads to an unmodified translational inflow velocity in the x-direction of the ship propeller of $12.35\,\frac{\mathrm{m}}{\mathrm{s}}$. The velocity field visible in Fig. 8.7 is subtracted from this to compute the inflow velocity to the panels of the ship propeller. The settings are chosen in such a way that they match the settings of the CFD simulation to compute the inflow to the ship propeller. In comparison to the simulation settings published in Lund [121], the frequency and the advance ratio are increased slightly to correct the settings to more realistic flow conditions. The general conclusions regarding the performance of the different convergence acceleration methods for the presented case are not changed due to these modifications in the simulation setup.

The first convergence acceleration variant considered is the unmodified fixed point iteration ($\tilde{\boldsymbol{d}}_{i+1}^t = \mathcal{A}(\boldsymbol{d}_i^t) = \boldsymbol{d}_i^t$ according to Eq. (4.20)) which is labeled "no accelerator" here. The second convergence acceleration variant is the "static relaxation" according to Eq. (4.21), with a relaxation factor of $\omega = 0.5$. The third variant is the Aitken method according to Eqs. (4.23) and (4.24), termed "Aitken 2" with an intermediate relaxation factor of 1. The fourth variant is the "Aitken 3" method, proposed by Irons and Tuck and introduced in Eq. (4.5.2) with an initial relaxation factor of 0.5. The single time-step version of the quasi-Newton least squares "QNLS" method (which was so far used as the convergence acceleration method in this chapter) is the fifth variant considered here and is explained in Sec. 4.5.3. The Broyden method according to Eqs. (4.28) and (4.29) with the unit matrix set as initial approximation $\boldsymbol{B}_0$ is the last variant considered here.

Table 8.4: Parameters for FSI simulation with ship wake used to compare different convergence acceleration methods.

parameter	value
rotation rate	$1.745\frac{1}{\mathrm{s}}$
angle increment $\Delta\alpha_f$	$2°$
time-step size Δt	$\frac{1}{180 \cdot 1.745}\,\mathrm{s} \approx 0.003184\,\mathrm{s}$
time of full rotational speed t_{rot}	$90\Delta t \approx 0.29\,\mathrm{s}$
traction ramp start t_{st}	$135\Delta t \approx 0.43\,\mathrm{s}$
traction ramp end t_e	$180\Delta t \approx 0.57\,\mathrm{s}$
wake roll-up enabled t_w	$370\Delta t \approx 1.18\,\mathrm{s}$
total simulation time t_e	$900\Delta t \approx 2.87\,\mathrm{s}$

The different tolerances used in this section are listed in Tab. 8.5. In the first step of the analysis, the convergence acceleration methods are compared using a stiff ship propeller with the Young's modulus of steel ($2.1 \cdot 10^{11}\,\mathrm{Pa}$) and the high tolerance as specified in Tab. 8.5. This is the same tolerance that was chosen as a good trade-off between accuracy and computational effort in the previous section. If the necessary number of implicit coupling iterations to obtain the chosen tolerance reaches 50, the implicit iteration loop for the current time step is stopped, as the coupling does not fully converge for some time steps.

In Fig. 8.8, the necessary number of implicit coupling iterations is plotted over the time step of the simulation. In the FSI simulation, different ranges can be identified based on the settings listed in Tab. 8.4. These different settings also lead to the different regions shown in Fig. 8.8. Range 1 starts at time step 135, where the first tractions are transferred from the fluid to the structural solver, and ends where the full tractions are transferred at time step 179. Consequently, Range 2 begins subsequently at time step 180, where the full tractions are transferred, and ends before the wake roll-up is enabled at time step 369. Thus, Range 3 extends from time step 370 to the end of the simulation.

Table 8.5: Relative and absolute tolerances (Eq. (4.18) and (4.19)) used in the simulations.

tolerance	high	middle	low
abs. tol. ϵ_a	10^{-6}	10^{-11}	10^{-15}
rel. tol. ϵ_r	$5 \cdot 10^{-3}$	$5 \cdot 10^{-8}$	$5 \cdot 10^{-11}$

No implicit iterations are necessary before applying the tractions at time step 135, as only the displacements are transferred from the structural to the fluid solver. Despite this, the coupling is already active before that time step (as visible in Fig. 8.8). In this case, a few time steps are necessary to fulfill the convergence criteria. In this range, the necessary number of implicit iterations is highest for the static relaxation with a relaxation factor of $\omega = 0.5$. The structural solver already computes the correct displacements in the first iterations, and the displacements remain unchanged in all subsequent iterations. Despite this, the static relaxation computes the displacements according to Eq. (4.21), and,

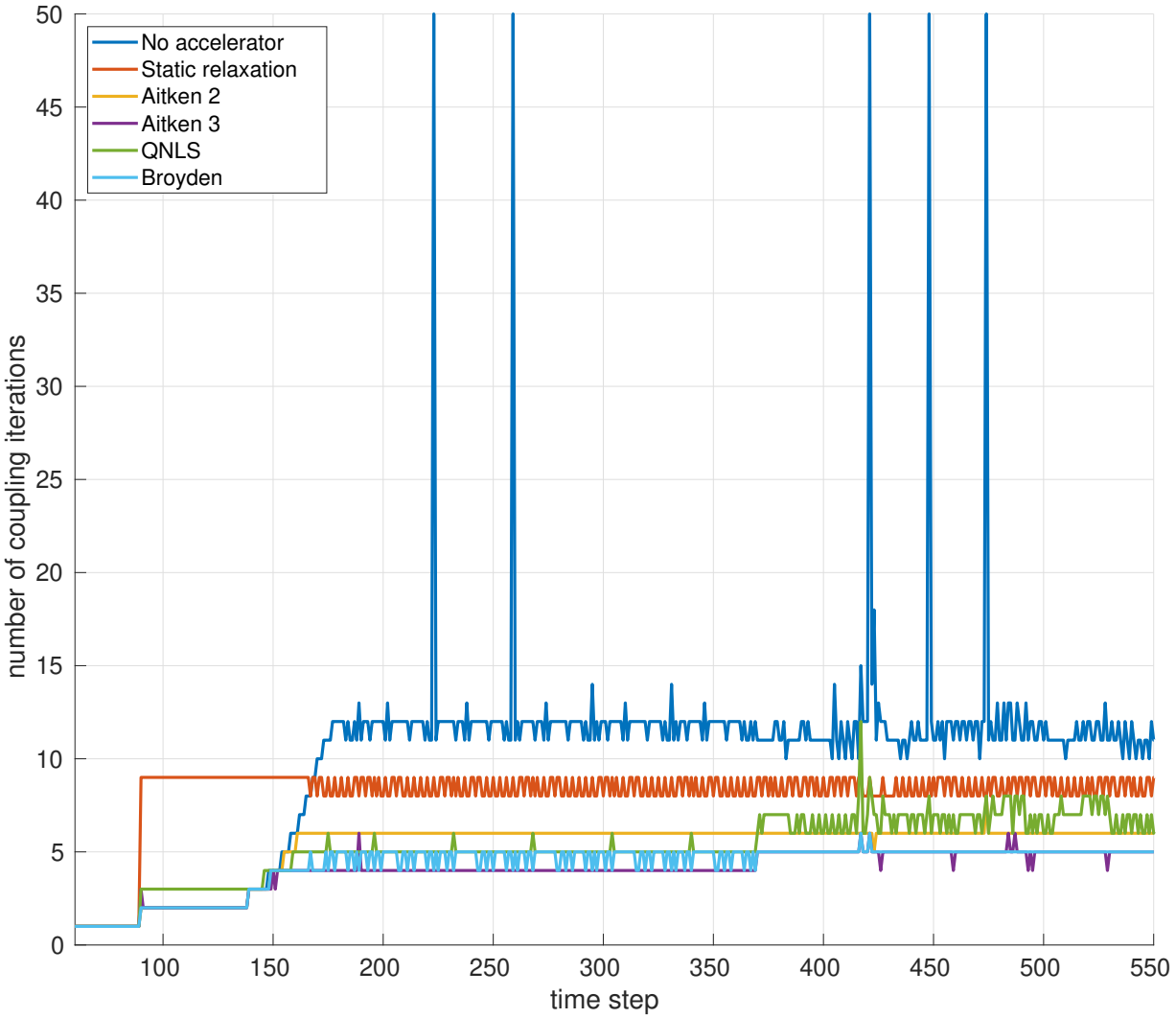


Figure 8.8: Comparison convergence acceleration methods over time for ship wake inflow, a Young's modulus of 210 GPa, and the high tolerance.

thus, the displacements sent to the fluid solver are only changed slowly. Therefore, the error in every iteration is only reduced by 50% in comparison to the previous iterations. In this scenario, the relative convergence criterion according to Eq. (4.19) is fulfilled first. Since the error in the first iteration is the reference for this criterion, the relative error in the subsequent iterations is 0.5^{i-1}, with i being the number of implicit iterations. The relative tolerance of $5 \cdot 10^{-3}$ is obtained accordingly after 9 iterations. This is quite a poor performance for the case where the displacements given to the static relaxation convergence acceleration method are constant. On the other hand, in contrast to all other convergence acceleration methods, the performance of the static relaxation does not deteriorate when the coupling is activated. In Tab. 8.6, the average number of coupling iterations in the different ranges introduced previously are listed for the simulation shown in Fig. 8.8. Here, it can also be seen that the static relaxation performs quite similarly in all three ranges.

The worst overall performance can be attributed to the variant without any convergence acceleration method active. As shown in Tab. 8.6, the necessary number of implicit coupling iterations in this variant is approximately doubled compared to the more advanced con-

Table 8.6: Implicit coupling iterations for different ranges, a Young's modulus of 210 GPa and a high tolerance. The lowest numbers of coupling iterations are highlighted.

	range 1	**range 2**	**range 3**
No accelerator	6.3	12.2	12.0
Static relaxation	8.8	8.4	8.4
Aitken 2	4.6	6.0	6.1
Aitken 3	3.6	4.0	5.1
QNLS	4.3	5.0	7.3
Broyden	3.7	4.7	5.1

vergence acceleration methods. Furthermore, the variant without convergence acceleration method fails to converge in certain time steps, as indicated by the peaks in Fig. 8.8. The more advanced convergence acceleration methods (Aitken and quasi-Newton methods) all perform quite well, although the Aitken 2 performs slightly worse than the other methods in range 2. In range 3, the QNLS performs slightly worse than the other advanced convergence acceleration methods. This might be caused by the numerical instabilities introduced by enabling the wake rollup. The wake rollup is enabled by a deformation of the wake that can lead to a self-penetration of the wake in the tip vortex region.

In conclusion, the Aitken 3 method and the Broyden method perform best in the first case considered here, while the Aitken 2 and the QNLS method show a performance that is only slightly worse.

In the next step, the convergence acceleration methods are compared compared for increasingly flexible propellers. The Young's moduli chosen here are 2 GPa, 5 GPa, 20 GPa, and 210 GPa. The high tolerance according to Tab. 8.5 is used for this comparison. The different Young's moduli are listed in Tab. 8.7 for comparison. In the case of a more flexible propeller with a Young's modulus of 20 GPa or lower, the version without a convergence acceleration fails – which once again highlights the importance of convergence acceleration methods. The performance of the static relaxation seems to be again almost independent of the varying input parameters. This makes the static relaxation one of the best-performing methods for the very flexible propeller with a Young's modulus of 2 GPa. Despite this, the static relaxation performs worse than all other convergence acceleration methods for a stiff propeller. Of the more advanced convergence acceleration methods, the Aitken 3 and the

Table 8.7: Implicit coupling iterations for a varying Young's modulus in range 2 and a high tolerance.

Young's modulus	**2 GPa**	**5 GPa**	**20 GPa**	**210 GPa**
No Accelerator	failed	failed	failed	12.2
Static relaxation	7.3	7.4	8.3	8.4
Aitken 2	11.9	9.0	6.1	6.0
Aitken 3	7.0	7.0	5.7	4.0
QNLS	9.8	9.6	8.0	5.0
Broyden	8.5	8.0	5.7	4.7

Table 8.8: Implicit coupling iterations for different tolerance and a Young's modulus of 20 GPa in range 2.

tolerance	high	middle	low
No Accelerator	failed	failed	failed
Static relaxation	8.3	24.9	34.9
Aitken 2	6.1	17.4	23.4
Aitken 3	5.7	12.5	16.8
QNLS	8	14	32.7
Broyden	5.7	13	18.9

Broyden method again perform best over the whole range of Young's moduli. The Aitken 2 method seems to struggle with the more flexible propellers.

In the last step, the different convergence acceleration methods are compared for the different tolerances listed in Tab. 8.5 for the propeller with a Young's modulus of 20 GPa in range 2. The results of this comparison are shown in Tab. 8.8.

The simulations without a convergence acceleration method fail again, regardless of the chosen tolerance. The static relaxation performs worse than all more advanced convergence acceleration methods. This method is very sensitive to the chosen tolerance – which can be explained by the effect mentioned earlier in this section: even for an unchanged displacement feedback from the structural solver, the error is only reduced by 50% in every iteration. The QNLS method performs quite well for the middle tolerance. In general, the QNLS method needs a certain number of implicit coupling iterations to perform well. This can be seen when comparing the QNLS method with the Aitken 2 method for high and middle tolerances. For a high tolerance and a correspondingly low number of implicit iterations, the Aitken 2 method performs better. For a middle tolerance and a correspondingly higher number of implicit iterations, the QNLS method performs much better. The relatively bad performance for the low tolerance can here be explained by the fact that the QNLS method does not fully converge in approximately a third of the time steps considered for this simulation. In these time steps, the procedure is stopped after 50 implicit iterations, and the next time step is started. If the unconverged time steps are excluded, the QNLS method requires an average of around 24 implicit iterations for the scenario shown in Tab. 8.8, performing approximately as well as the Aitken 2 method.

From among the convergence acceleration methods, the Aitken 3 method once again exhibits the best performance for the varying tolerances. The Broyden method is only slightly less effective compared to the Aitken 3 method. Here, the low tolerance leads, in comparison with the analysis presented in Tab. 8.3 (where the QNLS method is used), to a much higher number of implicit iterations. The reason for this is the more realistic varying load that leads to a different deformation due to the tractions in every time step.

In conclusion, the Aitken 3 method performs best for the chosen setup and also offers the benefit of a relatively simple implementation compared to the quasi-Newton methods. With the Aitken 3 method, the number of implicit coupling iterations is reduced by 20% to 30% compared to the QNLS method used so far. While the QNLS method performs a little bit better for lower tolerances, as shown in Tab. 8.3, the high tolerance still leads to a sufficiently high accuracy for the presented application.

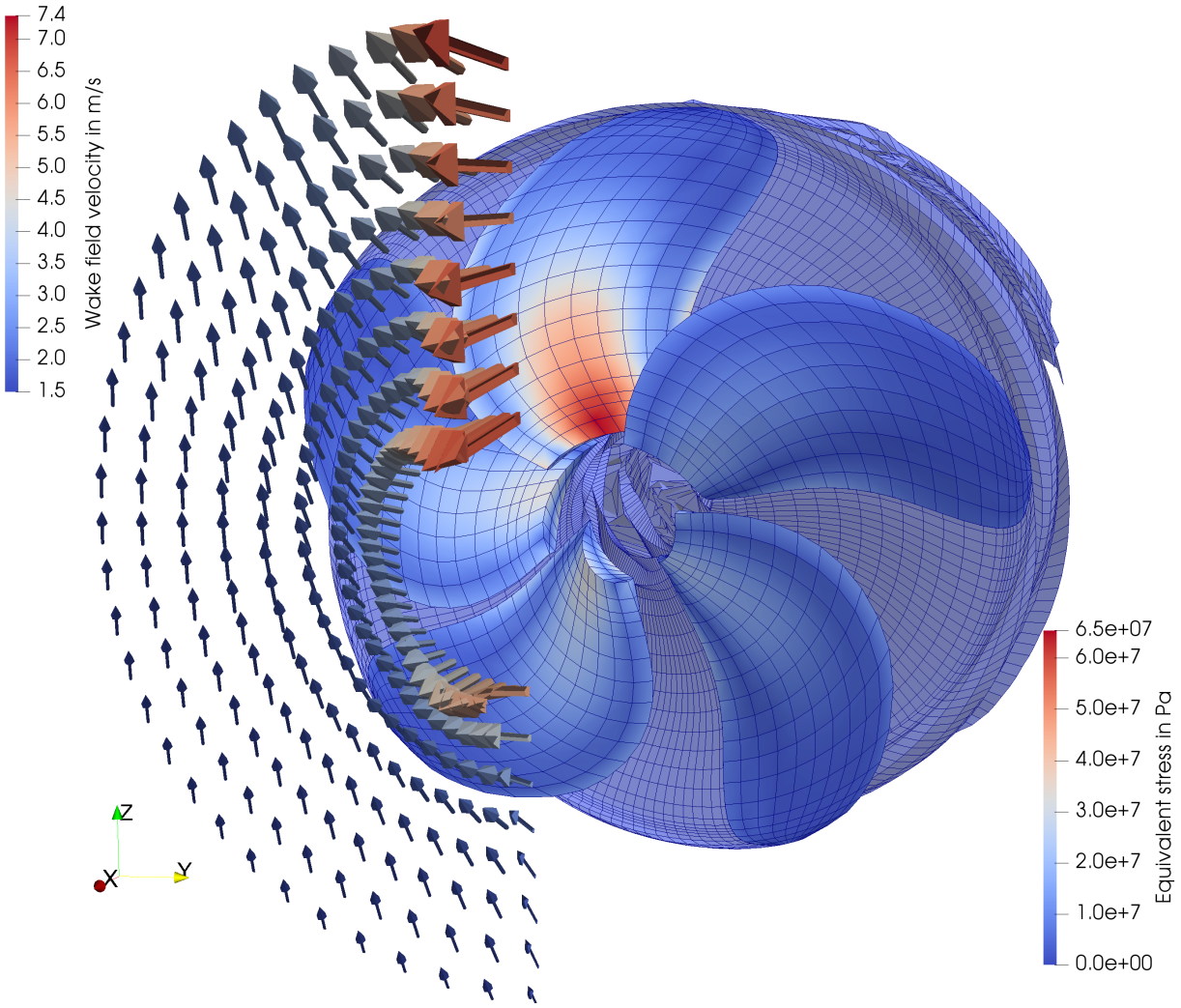


Figure 8.9: Full simulation after wake roll-up is activated with wakefield velocities indicated with arrows and equivalent stress of the different propeller blades.

The performance of the QNLS method could possibly be improved, e.g., by choosing a parallel coupling scheme instead of the staggered coupling scheme used here. In contrast to the staggered coupling scheme, the parallel coupling scheme solves both fields simultaneously. Radtke showed in [77] that the necessary number of implicit coupling iterations with the QNLS method can be reduced with a parallel coupling scheme. Furthermore, the performance of the QNLS method could be improved by using the multi-time step version and fine-tuning the settings for filtering (see Sec. 4.5.3). These more advanced versions of the QNLS method have the disadvantage that more parameters, such as the number of reused time steps, have to be fine-tuned for the given case. As a general conclusion, the comparison in this section highlights the importance of using advanced acceleration methods in partitioned coupled simulations to obtain a stable and fast converging solution.

Fig. 8.9 shows the full simulation of the stiff propeller (Young's modulus 210 GPa) which moves in positive x-direction in the simulation. Additionally, the varying flow velocity due to the ship wake inflow condition according to the wakefield introduced in Fig. 8.7 is visualized. As can be seen, the wake panels behind the propeller exhibit roll-up at the

tip. The varying inflow condition leads to varying traction acting on the propeller and, subsequently, to pronounced differences in the equivalent stress for the individual blades. It is quite challenging to accurately capture the loads acting on a propeller blade during the full loading cycle using traditional design methods. This again highlights the benefit of the fully coupled FSI simulation for this application case.

It is essential to select a well-performing convergence acceleration method to reduce the computational time for the optimization of the ship propeller in the next part since the optimization requires a high number of FSI simulations. As the performance of the Aitken 3 method is best for the presented setup, it will be used for the optimization in the following.

8.2 Optimized propeller

In the following, the aim is to optimize the KCS propeller introduced in the previous section. The optimization here aims to improve the propeller performance regarding low noise emissions, high efficiency, and a thrust that is close to the target thrust at the operational point. The thrust and the efficiency can be obtained directly from the simulation, as done in the previous section and chapter. The main source of noise emissions is the pressure variation in the proximity of the propeller, which leads to sound radiation. If cavitation is present, the major source of this pressure variation is cavitation on the propeller blade [114, p. 10]. Thus, a common approach to reduce noise emission is to reduce cavitation as much as possible [122, p. 268], and this is also the approach that will be followed here.

8.2.1 Optimization methods

Every optimization is composed of a set of input and output parameters. The input parameters $\boldsymbol{p}$ are the variables that are controlled to reach the optimization goal, meaning that the right input parameters are the sought quantity. In the case of the ship propeller, the input parameters in the first stage are the geometry of the ship propeller, and in the second stage (in addition to the geometry) also the material parameters. The output parameters $\boldsymbol{p_o}$ are the variables to be optimized.

To this end, the output parameters must be combined in a function that rates the quality of the output parameters with respect to the aim of the optimization. This combination of the output parameters is the target function $\mathcal{T}$ (also called cost, fitness, or objective function), and the result is the fitness of the chosen design f_{fit}. For the optimization of the ship propeller, the output parameters are the thrust T, the thrust variation over time T_σ, the efficiency η, and the sound emissions. Since the sound emissions are dominated by the effects of cavitation, the cavitation area k_{cav} on the blade surface and its variation $k_{cav,\sigma}$ over time are taken as the variables representing the sound emissions. The variations T_σ and $k_{cav,\sigma}$ are computed as the standard deviation of the thrust and the cavitation area over the time steps of one rotation. With these variables, the target function $\mathcal{T}$ can be written as

$$f_{fit} = \mathcal{T}(\boldsymbol{p_o}) = \mathcal{T}(T, T_\sigma, \eta, k_{cav}, k_{cav,\sigma}). \tag{8.1}$$

The target function includes weighting factors that rate the importance of the different output parameters. For the presented application, the target function is chosen such that the fitness f_{fit} is minimized by the optimization.

The output parameters are determined by a numerical simulation $\mathcal{S}_{sim}$ based on the input parameters:

$$\boldsymbol{p_o} = \mathcal{S}_{sim}(\boldsymbol{p}). \tag{8.2}$$

Combined with Eq. (8.1), the fitness can be computed with the simulation as

$$f_{fit} = \mathcal{T}(\boldsymbol{p_o}) = \mathcal{T}(\mathcal{S}_{sim}(\boldsymbol{p})). \tag{8.3}$$

The basic setup for the simulation $\mathcal{S}_{sim}$ was already introduced in this chapter. Before the target function $\mathcal{T}$ is specified based on the input parameters, the optimization algorithm should be introduced in the next section.

Optimization algorithm

Traditional optimization approaches are often gradient-based algorithms that require the computation or approximation of gradients and other derivative operators [123, p. 195]. When considering complex problems, such as the numerical simulations $\mathcal{S}_{sim}$, it is usually not possible to compute the gradients directly, and approximation based on, e.g., finite differences is often prohibitively expensive. Furthermore, gradient-based algorithms tend to converge to local minima. Therefore, other optimization techniques are often more attractive for these highly nonlinear and non-differentiable problems.

Optimization problems generally require certain properties of the optimization algorithm – such as general applicability, parallelizability, as well as robust and fast convergence to the global minimum of the target function, according to Eq. (8.3) [124, p. 342].

Evolutionary algorithms are a good choice for the presented problem since the simulation setup is strongly nonlinear and very costly, as a single simulation run requires hours of computation time on a node of the HPC. The evolutionary algorithms are inspired by the theory of evolution, which is ubiquitous in nature and which postulates the process of "natural selection" [125, p. 77], often referred to as "survival of the fittest". The idea of an evolutionary algorithm will be briefly outlined in the following.

For each input parameter p_i, a range is defined for the optimizer. Based on this range and the population size, the optimization is started with an initial population in generation 1. This population has a number of individuals, each referring to a single simulation, and it can be created in a structured manner or randomly. After all simulations for a population have been completed and the target functions have been evaluated, the best individuals are selected from the population and transferred to the new population in the next generation [126, p. 12364]. In the next generation, the individuals of the population are modified based on concepts of mutation and crossover.

Fig. 8.10 visualizes an exemplary evolutionary algorithm with crossover and mutation for a population size of m individuals and a set of n input parameters. The set of input parameters for individual j at generation i is called $\boldsymbol{p}_{Ij,Gi}$. The corresponding scalar input parameter k is $p_{k,Ij,Gi}$. The fitness of an individual is computed according to Eq. (8.3) as

$$f_{fit,Ij,Gi} = \mathcal{T}(\mathcal{S}_{sim}(\mathbf{p}_{Ij,Gi})). \tag{8.4}$$

For every individual, a trial $\boldsymbol{t}_{Ij,Gi}$ is performed for every generation to obtain a better fitness $f_{tfit,Ij,Gi}$, computed as

$$f_{tfit,Ij,Gi} = \mathcal{T}(\mathcal{S}_{sim}(\mathbf{t}_{Ij,Gi})). \tag{8.5}$$

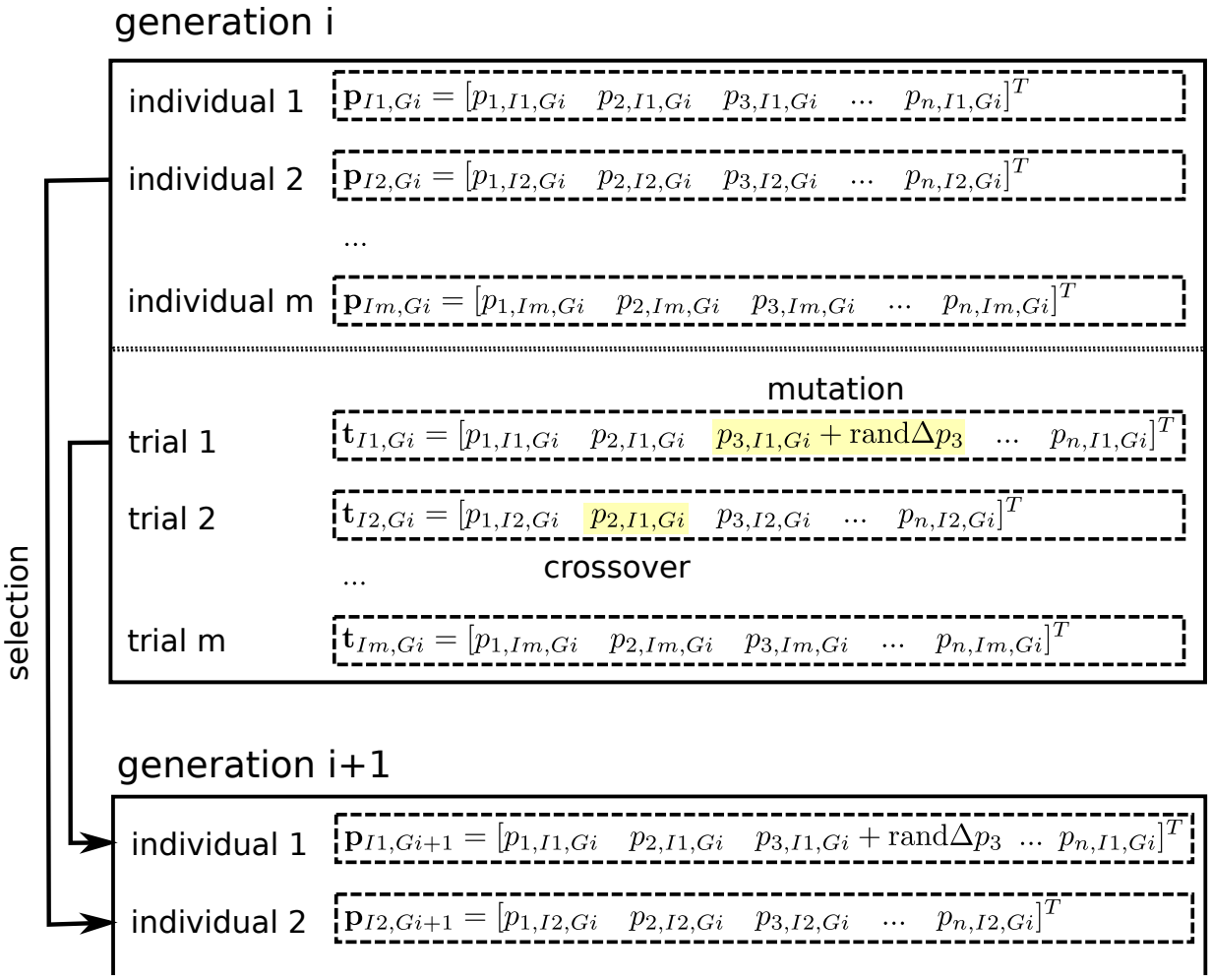


Figure 8.10: Visualization of an exemplary evolutionary algorithm.

If the fitness of the trial is better than the fitness of an individual ($f_{tfit,Ij,Gi} < f_{fit,Ij,Gi}$), the trial is selected as the individual for the next generation. This is exemplified for trial 1 in Fig. 8.10. If the fitness of the trial is worse than the individual ($f_{tfit,Ij,Gi} > f_{fit,Ij,Gi}$), the individual is kept in the selection step for the next generation. This is shown for trial 2 in Fig. 8.10. Individuals that are unchanged in the new generation do not have to be recomputed in the next step. Even though more complex decision mechanisms for the selection are possible, this approach, also known as the "greedy criterion", is fairly common [124, p. 344].

Mutation refers to a stochastic change of some of the input parameters of an individual. One possibility to realize a mutation is included for input parameter $p_{3,I1,Gi}$ in the example. For this mutation, a random number rand is multiplied with the difference Δp_3 between two input parameters of parameter 3 from randomly chosen individuals of the current generation. The purpose of a mutation is often to perform a local search around an individual [127, p. 25]. Therefore, the random number is commonly chosen to be smaller than 1. Instead of the random number, it is also possible to use a constant factor, as randomization is also included by choosing arbitrary individuals for the computation of Δp_3 [124, p. 344].

Here, "crossover" refers to the random exchange of parameters between individuals of a population [126, p. 12364]. The crossover rate and mutation rate of individuals can be influenced by stochastic parameters that increase or decrease the likelihood that input parameters of individuals are mutated or exchanged due to crossover. As an example, the input parameter $p_{2,I2,Gi}$ is exchanged with $p_{2,I1,Gi}$ in Fig. 8.10.

The probability that a parameter of an individual is changed by mutation or crossover is called, respectively, "mutation rate" and "crossover rate" [127, p. 25]. Crossover and mutation can be modified and combined in many different ways, the explanation above is just a simplified introduction to the concepts used for this type of algorithm.

For the implementation of the evolutionary algorithm, the software package Dakota [128] is chosen as the optimization code with the optimizer *coliny_ea* to realize the optimization with an evolutionary algorithm.

8.2.2 Automatic framework

The simulation operator $\mathcal{S}_{sim}$ connecting the input and output parameters is composed of more than just the FSI simulation with *comana*. A flowchart that covers the entire automatic framework for the simulation is shown in Fig. 8.11. Bash and Python scripts are used to control this process.

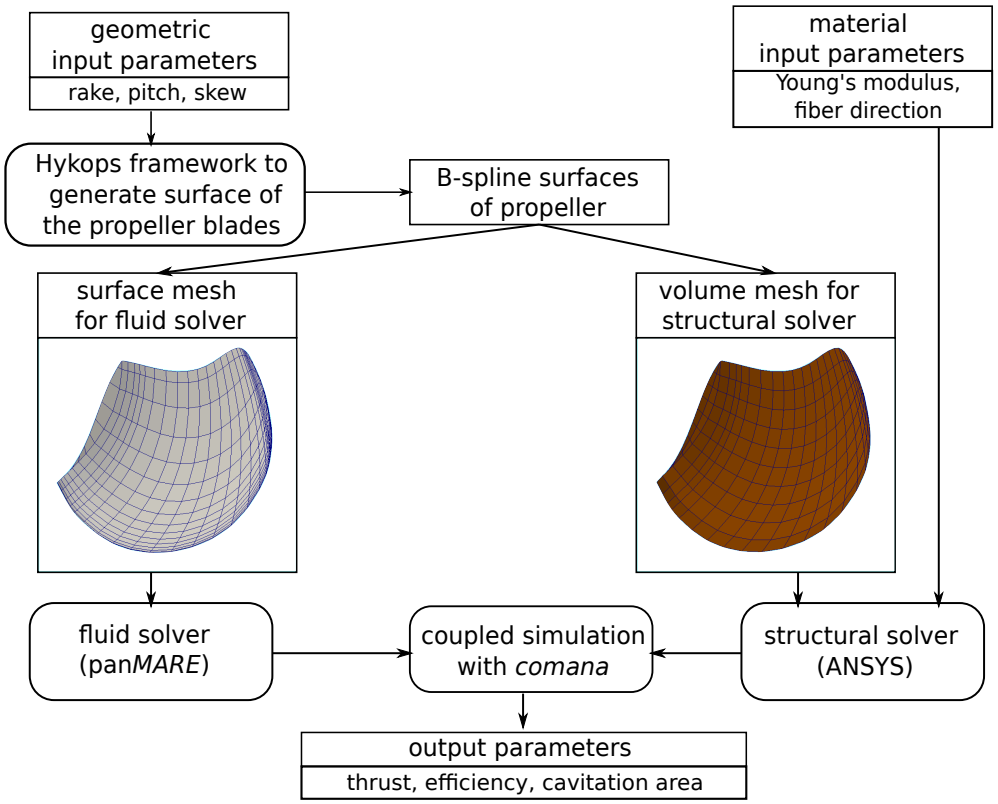


Figure 8.11: Automatic simulation framework $\mathcal{S}_{sim}$ to connect the input with the output parameters.

To modify the geometry of the propeller blades, the description of the geometry modifications to the KCS propeller must be reduced to a small set of input parameters. This is done

with the software framework HYKOPS [129, 130]. HYKOPS modifies the geometry of the propeller blades by changing the rake, pitch, and skew at different locations on the blade. With HYKOPS, it is also possible to generate tip rake ship propellers to reduce tip vortex effects by a high rake angle close to the tip. More information about the implementation details of the HYKOPS framework can be found in Stoye et al. [130].

Based on the resulting surfaces (described as B-splines), a volume mesh for the structural solver is generated with an in-house meshing code in the same way as described in Sec. 8.1.1. The surface mesh for the fluid solver is generated in a similar manner. The material input parameters (explained in more detail later) are directly set in ANSYS.

After the simulation is completed, the target function according to Eq. 8.1 is evaluated and the resulting fitness f_{fit} is sent back to Dakota.

8.2.3 Anisotropic material

In this section, the transversely isotropic material model introduced in Sec. 2.2.2 is applied to the ship propeller simulation to enable the simulation of ship propellers made of fiber-reinforced polymers. In the first part, the bend-twist coupling is demonstrated in an exemplary simulation in ANSYS. In the next part, the material model is applied to the ship propeller.

Bend-twist coupling

Ship propellers made of anisotropic material (like carbon fiber-reinforced polymers) offer additional advantages to the aforementioned benefits of reduced weight and potentially lower noise emissions. They also have the advantage that the deformation of the ship propeller can be used to improve the efficiency. This is possible due to an effect called bend-twist coupling.

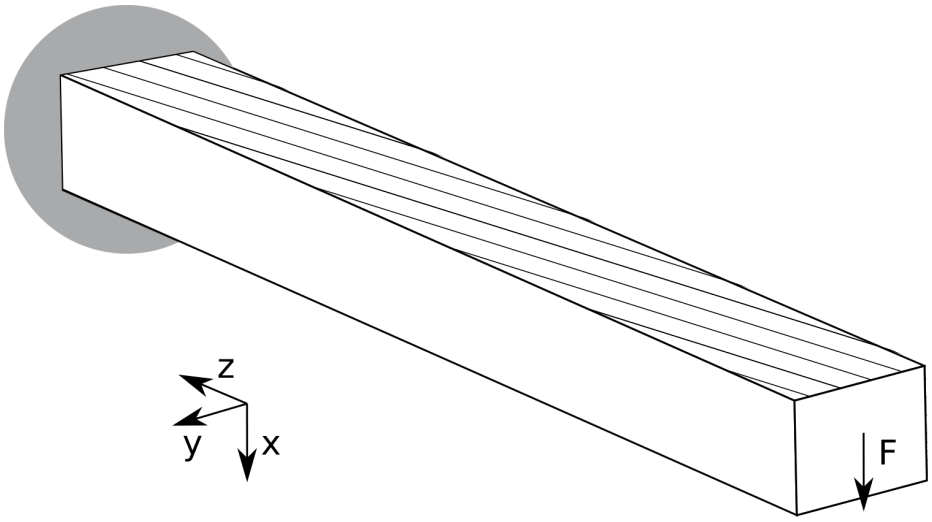
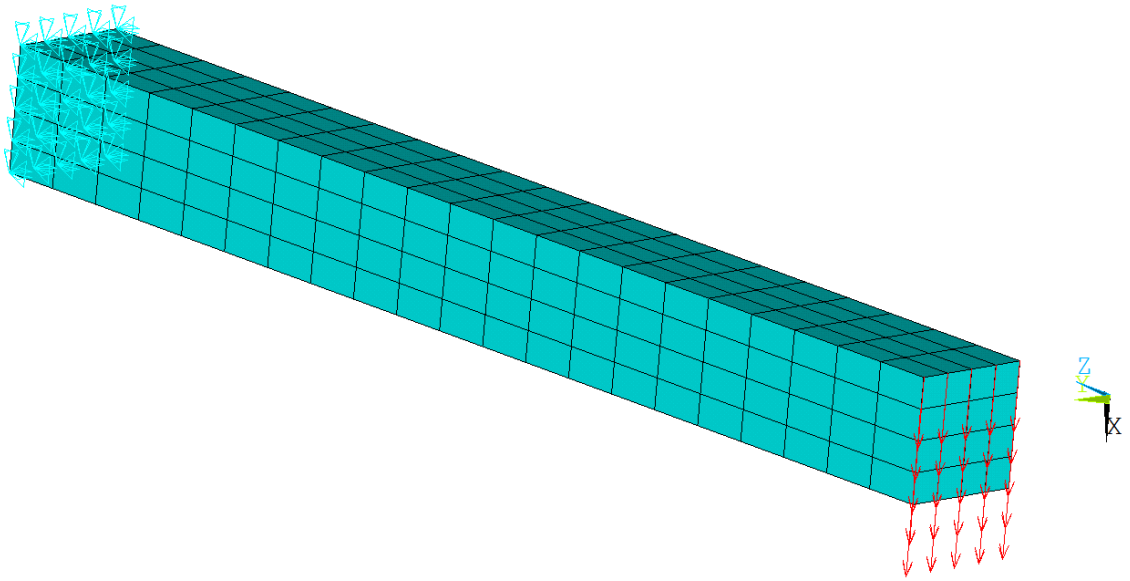
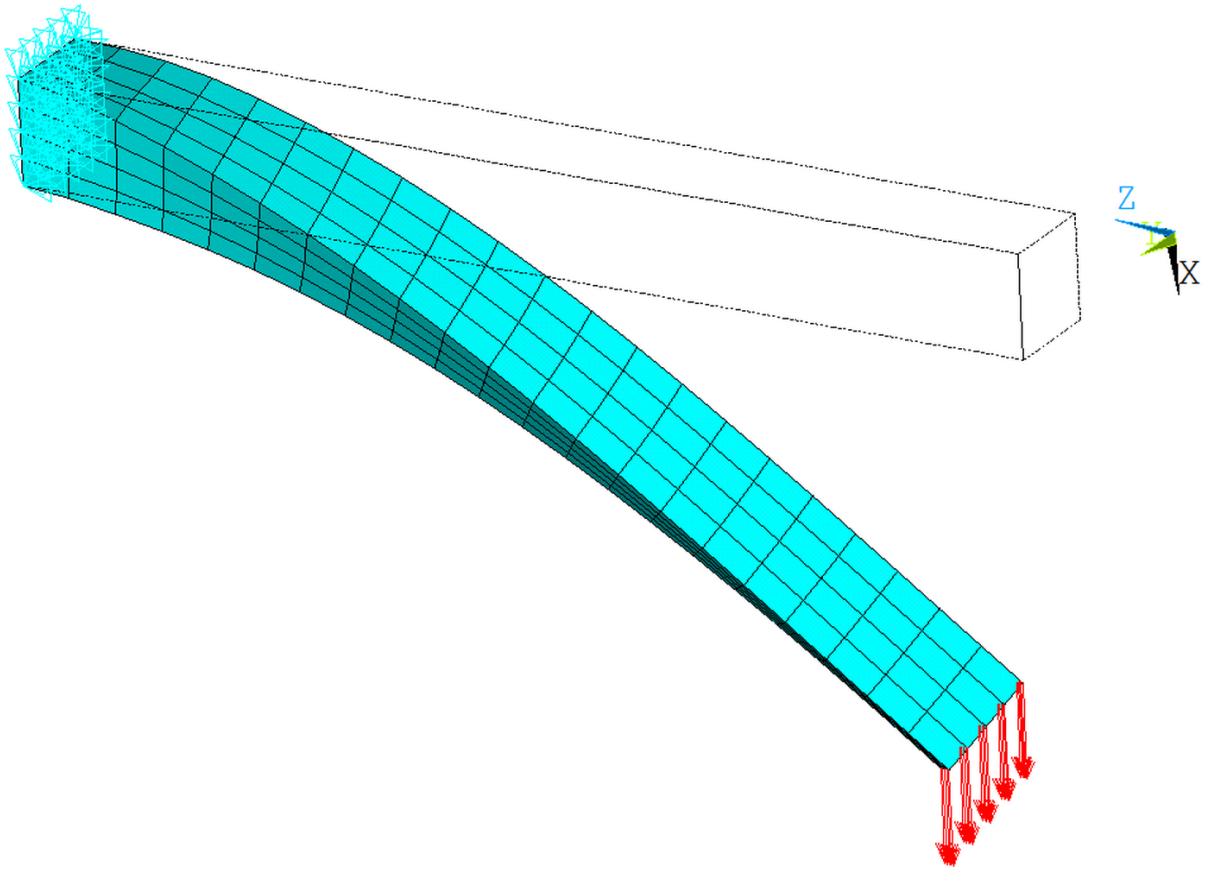


Figure 8.12: Setup for bend-twist coupling simulation test.



(a) Initial state



(b) Deformed state.

Figure 8.13: Bend-twist coupling simulation of a beam in ANSYS.

This effect can be visualized by means of a nonlinear simulation in ANSYS with a transversely isotropic material model. To demonstrate this effect in the simulation, a Young's modulus is selected that is two magnitudes higher in the fiber direction than in the other two directions. The simulation setup is inspired by Rohde et al. [22] and, as shown in Fig. 8.12, contains a beam that is clamped on the rear (marked by the grey area in the figure) and loaded by a distributed force on the other side. The fiber direction in the beam is shifted by 18° from the z-direction towards the positive y-direction, as illustrated for the top surface of the beam.

Bend-twist coupling is an effect that can be explained by the shifted shear center of the beam due to the slightly rotated fiber direction. The shear center is the location of a cross-section of a beam at which a loading produces pure bending and no torsion [131, p. 251]. Details on the computation of the shear center for isotropic materials with an unsymmetrical cross-section can be found, e.g., in Yoo [132, p. 288 ff.]. In these cases, the bend-twist coupling can be explained by an imbalance of the shear stresses in the cross-section, necessitating the beam to twist to maintain the balance of linear and angular momentum [22, p. 1614].

In ANSYS, a regular hexahedral mesh for the setup shown in Fig. 8.12 is generated, as shown in Fig. 8.13a. The fiber direction of every element is shifted separately by means of the local coordinate system to enable a layered mesh with different fiber directions in every layer in the later simulations with the ship propeller. The loading in the x-direction leads to strong torsion around the z-axis, as visible in Fig. 8.13b. If the simulation setup were to be modified to shift the fiber orientation slightly to the positive instead of the negative y-direction, the rotation due to the twist effect would be reversed in direction.

This effect can be utilized for ship propellers to modify the pitch of a propeller based on the external loads acting on the propeller, aiming to achieve better efficiency under operating conditions as shown by Young [21], Liu et al. [5], and Mulcahy et al. [23].

Model for anisotropic ship propeller

This section serves to introduce a model for an anisotropic ship propeller that can be used for further optimization – taking advantage of the bend-twist effect and the reduced noise emissions of fiber-reinforced polymers.

The material properties of fiber-reinforced polymers strongly depend on the exact manufacturing process of the material. Hildebrandt et al. [133, p. 2327] obtained ranges for Young's and shear moduli for CFRP materials based on experimental investigations. The unscaled Young's modulus for the CFRP material used in the anisotropic ship propeller simulation was chosen based on these experimental results.

Table 8.9: Material parameters of CFRP taken from [133, p. 2327].

	E_f	E_t	ν	G_{ft}
lower limit	90 GPa	5 GPa	0.3	2.6 GPa
best parameters	110 GPa	5 GPa	0.3	2.98 GPa
upper limit	110 GPa	30 GPa	0.3	3.4 GPa
FSI simulation	110 GPa	5 GPa	0.3	3 GPa

The corresponding material parameters determined in the experiment [133, p. 2327] are listed in Tab. 8.9 together with the parameters that were chosen for the FSI simulations of the CFRP ship propeller.

The transverse isotropic material model as detailed in Sec. 2.2.2 is used here with the material parameters Young's modulus E_f in the fiber direction, Young's modulus E_t in transverse directions, Poisson number ν, and the shear modulus G_{ft}. The Poisson number is assumed to be independent of the direction. The shear modulus G_{tt} for the FSI simulation can be computed with Eq. (2.13) as $G_{tt} = 1.92\,\text{GPa}$. With this, all variables to compute the elasticity matrix (or its inverse) according to Eq. 2.12 are given and the anisotropic properties are fully determined.

The aspect of fiber orientation in a realistic CFRP propeller is complex and, like the material parameters, depends strongly on the manufacturing process. Since the study presented here focuses on the principle effects rather than on the manufacturing process, the unmodified fiber direction is set to the element orientation in the span direction (from hub to tip). In Fig. 8.14, the corresponding unmodified fiber direction is indicated as a red line in the close-up view.

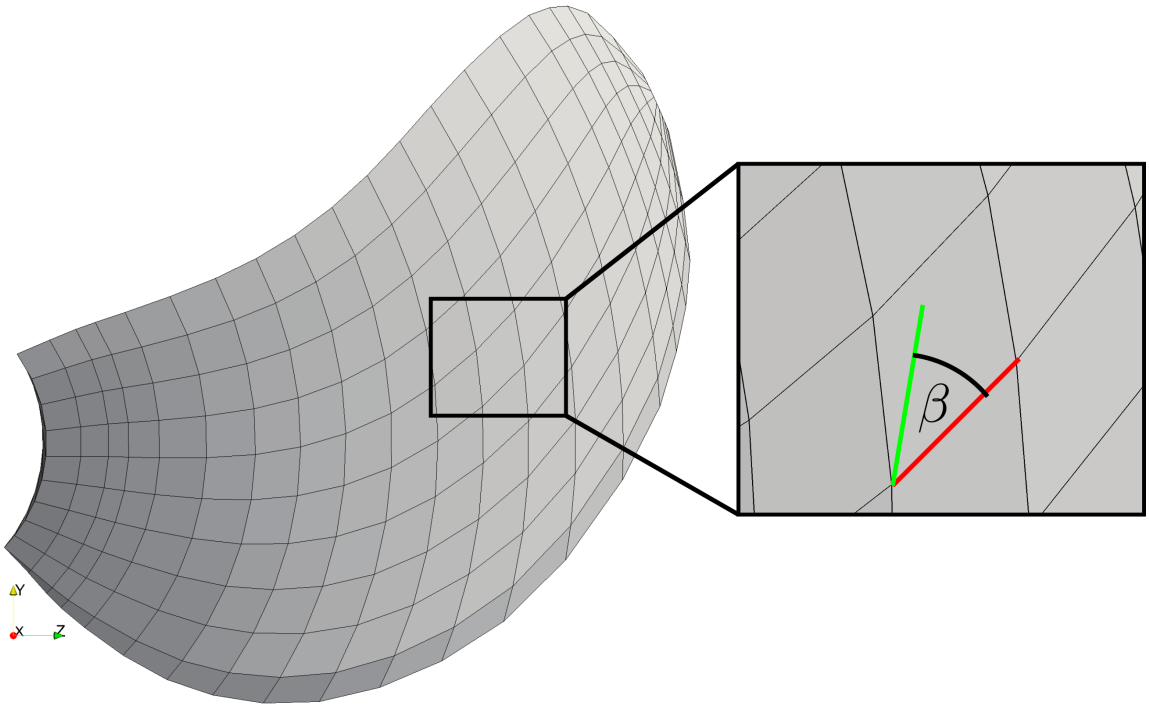


Figure 8.14: Fiber orientation of the anisotropic propeller.

To enable a modification of the fiber direction, an angle β is defined that rotates the fiber direction to a modified fiber orientation. In Fig. 8.14, the modified fiber is represented by a green line in the close-up view. The same fiber angle is used for all elements in the simulation of a propeller.

The angle β can be used as an input parameter in the optimization that influences the bend-twist coupling introduced in the previous section.

To scale the stiffness, a scaling factor s_E is defined. All Young's and shear moduli in the simulation are multiplied by this scaling factor. The scaling factor is set to be smaller or

equal to one, to ensure that it can be used to increase the overall flexibility of the blade. A corresponding increase in flexibility could be obtained in the manufacturing process by changing the composition of the material (e.g., a fiber packing with a lower density) or using a fiber and matrix material with a lower Young's modulus. A lower Young's and shear modulus should be easier to realize than a higher modulus.

With this simplified model, it is possible to investigate the optimization potential of a flexible anisotropic ship propeller.

8.2.4 Parameter study of anisotropic material

To analyze the bend-twist effect for the KCS propeller, another parameter study serves to investigate the influence of the stiffness scaling factor s_E and the fiber direction β on the thrust coefficient, the torque coefficient, the efficiency, and the cavitation area.

The area of a panel in *pan*MARE is assumed to be part of the cavitation area if the pressure of the panel is below the vapor pressure. The vapor pressure for water at a temperature of $20 - 21\,°\mathrm{C}$ is around $2.4\,\mathrm{kPa}$ [134, p. 6-10].

For this parameter study, the ship propeller inflow is assumed to be uniform (open water condition), and the advance ratio J is set to 0.8. The other parameters of the simulation are listed in Tab. 8.10.

Table 8.10: Parameters for the anisotropic KCS propeller simulation in open water condition.

parameter	value
rotation rate	$1.5\frac{1}{\mathrm{s}}$
angle increment $\Delta\alpha_f$	$2°$
time-step size Δt	$\frac{1}{270}\,\mathrm{s} \approx 0.003704\,\mathrm{s}$
time of full rotational speed t_{rot}	$90\Delta t \approx 0.33\,\mathrm{s}$
traction ramp start t_{st}	$135\Delta t \approx 0.5\,\mathrm{s}$
traction ramp end t_e	$180\Delta t \approx 0.67\,\mathrm{s}$
wake roll-up enabled t_w	$370\Delta t \approx 1.37\,\mathrm{s}$
total simulation time t_e	$1800\Delta t \approx 6.66\,\mathrm{s}$

The parameters that are varied in this study are the stiffness scaling factor and the fiber direction angle introduced in the previous section. The scaling factor for the Young's modulus is varied to $s_E = 0.1, 0.2, 0.5$, and 1.0. The fiber direction angle is varied to $\beta = 0°, 10°, 20°, 40°, 60°, 80°, 100°, 120°, 140°, 160°$, and $170°$. This variation in angle is sufficient, as the fiber angle of 170°, for example, is equivalent to a fiber angle of -10°. A fiber angle of 180° or more would therefore be repetitive.

The fiber angles of 10° and 170° are added since the bend-twist effect is expected to be most significant for fiber angles up to 25° [22, p. 1615].

The simulations here are not entirely realistic since the open water condition is used. The open water condition (uniform inflow) has the benefit that the deformation remains unchanged after the run-in of the simulation – which makes it easier to analyze the twist of the blades. Compared to the ship wake inflow condition, the efficiency is higher and cavitation is much less apparent for the open water condition. A selection of the simulation

Table 8.11: Results for varying scaling factor and fiber angle of anisotropic ship propeller.

scaling fact.	fiber ang.	thrust coeff.	torque coeff.	efficiency	cav. area [%]	color
stiff	stiff	0.132	0.022	0.77	<0.01	grey
0.1	0°	0.109	0.018	0.79	<0.01	
0.1	10°	0.118	0.019	0.80	0.03	
0.1	20°	0.135	0.022	0.80	0.07	
0.2	40°	0.201	0.035	0.73	0.37	red
0.2	60°	0.217	0.049	0.56	2.12	
0.1	80°	0.055	0.010	0.69	0.42	
0.5	100°	0.129	0.021	0.77	<0.01	
1	100°	0.134	0.022	0.77	<0.01	
0.1	160°	0.093	0.015	0.78	0.06	blue

results is listed in Tab. 8.11. If the cavitation area of the propeller is below 0.01 %, it is summarized in the table as <0.01 since small cavitation areas always occur, due to numerical errors. The simulation results for all parameters (excluding the simulations that failed due to convergence issues) are listed in Appendix A.1.

The first row contains the results of a reference simulation with a stiff KCS propeller. The output parameter of the flexible anisotropic propellers can be compared to this reference. When comparing the two propellers with a high stiffness scaling factor and a fiber angle of 100° to the stiff propeller, only small differences are noticeable. The cavitation area for those two propellers is very low, like for the stiff propeller, the efficiency is almost unchanged, and only a small variation in the thrust is visible. This pattern is similar to most of the other propellers with a high stiffness scaling factor, as can be seen in Tab. A.2.

The two propellers that produce the highest thrust have a stiffness scaling factor of 0.2 and fiber angles of 40° and 60° respectively. These two propellers also exhibit a high cavitation area and a low efficiency. A high cavitation area and a low efficiency can also be observed for the propeller with a stiffness scaling factor of 0.1 and a fiber angle of 80°, even though this propeller produces only very low thrust. This shows that the thrust seems to be quite sensitive to the input parameters, as the lower scaling factor and the small change in the fiber angle leads to a completely different thrust. Again, this propeller produces quite bad results with regard to cavitation area and efficiency.

To visualize the bend-twist effect of the blades, the deformation of three of the propellers listed in Tab. 8.11 is plotted in Fig. 8.15 from three different perspectives. The colors used for the blades in this figure are also listed in Tab. 8.11. The propeller moves in positive x-direction, and the thrust acting on the blade acts, accordingly, in positive x-direction as well. Therefore, the red propeller has a pitch that is significantly increased compared to the grey (stiff) propeller. The increase in pitch is only apparent under loading and is caused by the previously introduced bend-twist effect. In addition to the modified pitch, a deformation in positive x-direction (caused by the thrust acting on the blade) is also visible.

The increased pitch leads to an approximately 50% higher thrust of the red propeller, but the necessary torque to maintain the frequency is increased even more, resulting in an overall lower efficiency. The deformation also leads to an increase in the cavitation area. An opposite effect with respect to thrust, torque, and efficiency is visible for the blue

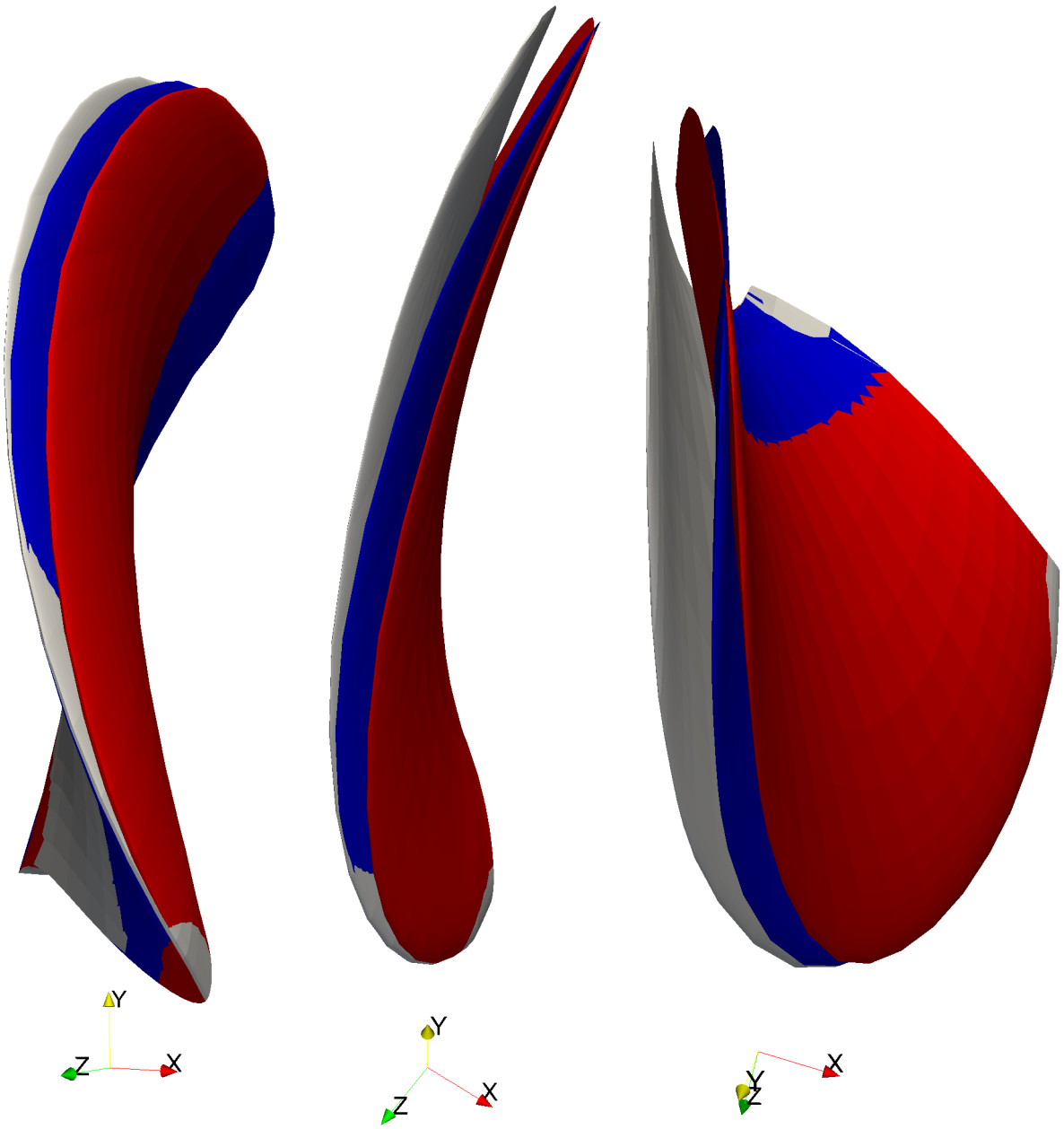


Figure 8.15: Modified pitch under loading visualized for different anisotropic propellers from three different perspectives.

propeller. This is caused by the reduced pitch angle, as shown in Fig. 8.15. The cavitation area for this propeller is slightly increased.

In conclusion, it can be seen that the output parameters (thrust, efficiency, and cavitation area) are sensitive to the input parameters (scaling factor and fiber angle). The desired effect demonstrated in the beam example (see Sec. 8.2.3), where the twist is controlled by the fiber angle, can be transferred to the ship propeller. Hence, this configuration setup is suitable to take advantage of the bend-twist effect. The next step is to optimize a propeller under the assumption of a realistic ship wake inflow condition.

8.2.5 Ship propeller optimization

In the first step of the optimization, the propeller geometry of the stiff KCS propeller is optimized with HYKOPS. This optimization was carried out by Neitzel et al. [135] with HYKOPS and *pan*MARE. In the second step, the anisotropic flexible propeller is optimized by modifying the fiber direction and the stiffness scaling factor as well as one of the geometry parameters. To this end, the automatic framework introduced in Sec. 8.2.2 is used together with the anisotropic material model explained in Sec. 8.2.3. The author of this work carried out the optimization of the flexible propeller. These results are also published in Gonzalés et al. [116].

The fitness function introduced in Eq. 8.1 is the first and one of the most important factors to control the optimization. The fitness function rates the importance of the different output parameters against each other. To this end, a few additional parameters are introduced. The cavitation area limit $k_{cav,lim}$ is used to change the importance of the cavitation area and fluctuation in the optimization. In addition to this, a penalty is introduced if the cavitation area is above this limit. The target thrust T_{target} is the thrust of the reference KCS propeller. The thrust weight c_T is used to increase the penalty of deviations from the target thrust. With the efficiency weight c_η, the penalty for the efficiency can be modified. With this, the fitness function for the ship propeller is defined as

$$f_{fit} = c_\eta \frac{1}{\eta} + \frac{k_{cav}}{k_{cav,lim}} + \frac{k_{cav,\sigma}}{k_{cav,lim}} + c_T \frac{|T - T_{target}|}{T_{target}} + c_{Ts} \frac{T_\sigma}{T_{target}}. \tag{8.6}$$

For the efficiency η, a plausible range is defined by using a lower limit η_l and an upper limit η_u. If the efficiency is out of this range, a penalty is added to avoid unrealistic high or low values. The values for the additional parameters used in Eq. (8.6) are listed in Tab. 8.12. This fitness function was initially used by Neitzel et al. [135, p. 5]. The chosen fitness function places more emphasis on the reduction of cavitation.

Table 8.12: Additional parameters for the definition of the fitness function for the propeller optimization.

c_η	η_l	η_u	$k_{cav,lim}$	T_{target}	c_T	c_{Ts}
1	0.65	0.9	2 %	2028 kN	4	4

The mutation rate for the optimization is set to 0.3 and the crossover rate to 0.1 (see Sec. 8.2.1 for definitions). With this, the focus is more on random modifications of the individuals rather than on the mixing of different individuals. The population size m is set to 24 individuals.

Optimization of stiff propeller

In the scope of the joint research project "Hydroelastic simulation of the acoustic behavior of ship-propeller configurations with and without cavitation", Neitzel et al. [135] performed an optimization of the geometry of the stiff KCS propeller with HYKOPS and *pan*MARE.

The optimization published by Neitzel et al. [135] is intended to serve as the basis for the optimization of the anisotropic flexible propeller, which will be presented in more detail later in this work. Therefore, the optimization of the stiff propeller will be outlined in brief here.

An input parameter that is used in geometry optimization of the stiff propeller is the pitch $\mathcal{P}_{0.6}$ of the propeller at a relative radius of 0.6 of the maximal radius. The pitch $\mathcal{P}_{0.6}$ of the KCS propeller is modified in the range from -6° to 6°. Fig. 8.16 shows the influence of this parameter on the KCS propeller. As shown in the previous section, the pitch strongly affects the thrust of the propeller – and, in combination with the bend-twist effect, it is important to optimize it such that the correct load-dependent pitch for the ship wake inflow condition is realized. Therefore, this input parameter will also be used in the optimization of the flexible propeller later on.

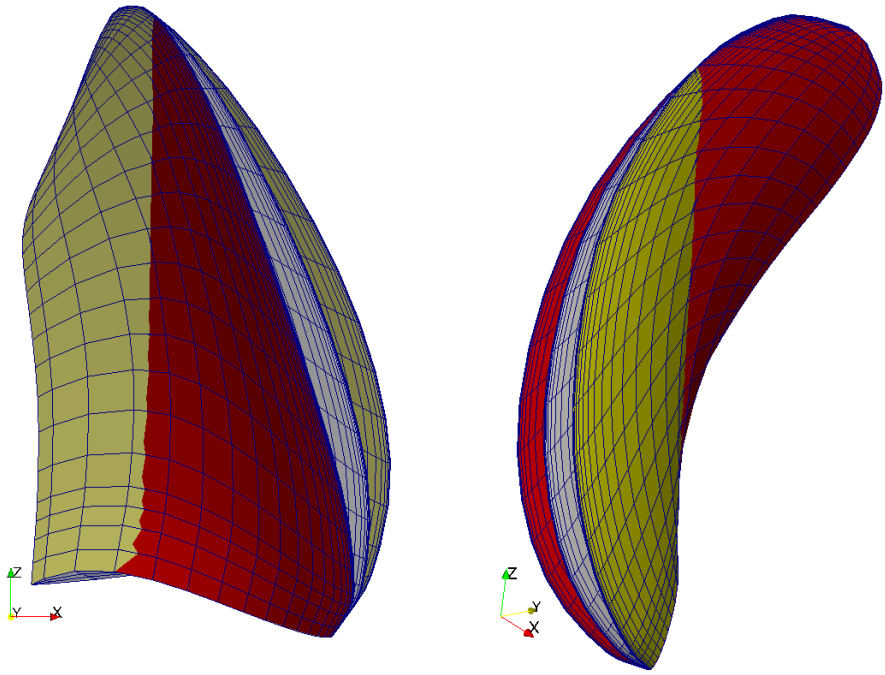


Figure 8.16: Pitch variation of the KCS propeller with HYKOPS (red:-6°, grey: initial KCS propeller, yellow: 6°).

Tip vortex cavitation is an important type of cavitation that causes noise emissions from ship propellers [111, p. 224]. Further, tip vortex cavitation can also cause damage to the rudder [136, p. 365]. A design that can be used to reduce this tip vortex cavitation and to improve the performance of a ship propeller is a tip rake modification [137].

The tip rake modification is also used to optimize the stiff propeller. Fig. 8.17 visualizes the rake modification on the tip of the KCS propeller with HYKOPS.

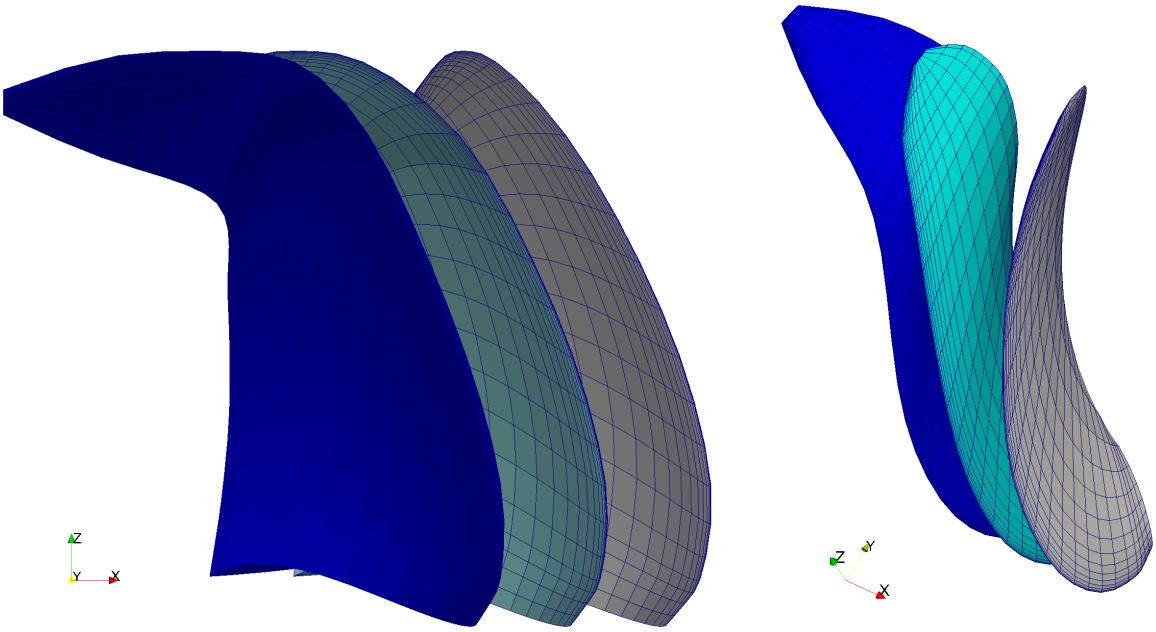


Figure 8.17: Tip rake variation of the KCS propeller with HYKOPS (dark blue: maximum tip rake modification, light blue: intermediate tip rake modification, grey: initial KCS propeller).

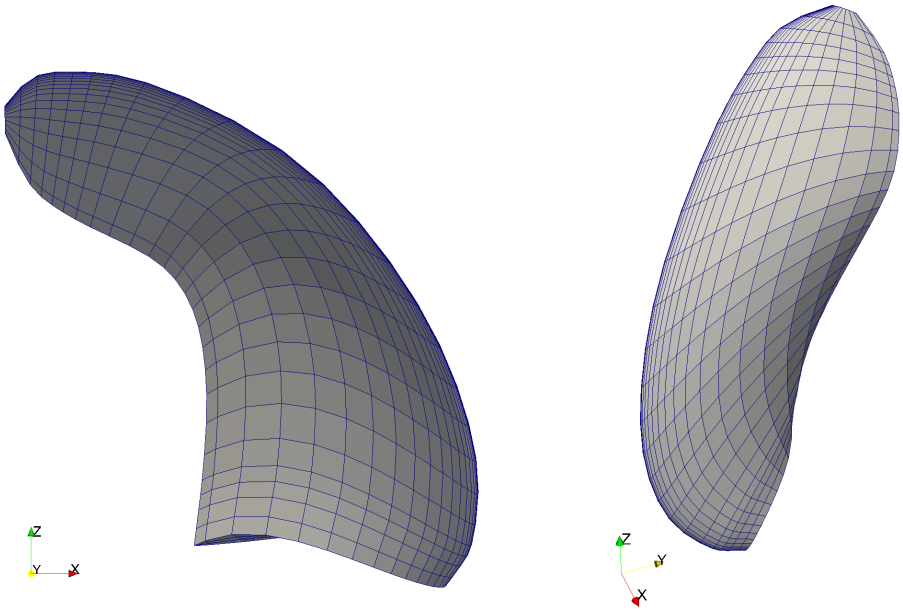


Figure 8.18: Propeller geometry PM 867 resulting from the optimization of the stiff propeller.

In the optimization of the stiff propeller, additional input parameters are considered, such as the pitch, camber, and tilt near the tip. Details regarding these parameters in HYKOPS can be found in Neitzel et al. [135, p. 5]. In the scope of this optimization, 2000 individuals were computed.

The propeller geometry resulting from this optimization based on uncoupled hydrodynamic simulations is called "PM 867" [135, p. 8] and is visualized in Fig. 8.18. Propeller geometry PM 867 serves as a basis for the first optimization of the flexible propeller with fully coupled FSI simulations in the next step. A complete comparison of the performance of this propeller compared to the initial KCS propeller will be provided in the next section, together with the optimized anisotropic flexible propellers.

Optimization of the flexible anisotropic propeller

The simulation parameters for the optimization of the flexible anisotropic propeller are modified slightly in order to reduce the simulation time for every individual of the optimization. The simulation covers the entire propeller with 5 blades and with the ship wake inflow condition, as depicted in Fig. 8.7.

Several hundred individuals are simulated, which makes this reduction in simulation time essential. Hence, the number of time steps is reduced compared to the simulation parameters presented in Tab. 8.4 – from 900 time steps to 540 time steps. This also requires that the wake deformation is disabled to reduce the run-in period of the simulation. The evaluation is done over the last 180 time steps (or the last full rotation). The advance ratio J is set to 0.896. The simulation parameters are listed in Tab. 8.13.

Table 8.13: Parameters for FSI simulation used for the optimization of the flexible anisotropic ship propeller.

parameter	value
rotation rate	$1.745\frac{1}{\mathrm{s}}$
angle increment $\Delta\alpha_f$	$2°$
time-step size Δt	$\frac{1}{180 \cdot 1.745}\,\mathrm{s} \approx 0.003184\,\mathrm{s}$
time of full rotational speed t_{rot}	$90\Delta t \approx 0.29\,\mathrm{s}$
traction ramp start t_{st}	$135\Delta t \approx 0.43\,\mathrm{s}$
traction ramp end t_e	$180\Delta t \approx 0.57\,\mathrm{s}$
total simulation time t_e	$540\Delta t \approx 2.72\,\mathrm{s}$

For the optimization of the flexible propeller, the pitch $\mathcal{P}_{0.6}$ of the propeller geometry PM 867 is taken as one of the input parameters. Since only the pitch is varied in the optimization, the original KCS propeller geometry is not in the parameter range of this optimization of the flexible propeller. In the first stage of the optimization, the pitch is varied once again between -6° and 6°. Otherwise, the propeller geometry is equal to the PM 867 propeller. In addition, the fiber angle is varied between 0° and 180°, and the stiffness scaling factor is varied between 0.1 and 1 in this stage. With this setup, the first optimization of the flexible propeller is conducted. The parameters listed in Tab. 8.12 are used with the fitness function Eq. 8.6.

Up to 12 individuals are simulated simultaneously. After Dakota completes the simulation of the 24 individuals in a population, the optimization procedure generates a new set of individuals as explained in Sec. 8.2.1. In total, the simulation of 396 individuals took approximately 5 weeks on the HPC of TUHH in the first stage of the optimization. The results for the 10 best individuals are listed in Tab. 8.14 together with the results for the stiff KCS and the PM 867 propeller. In this table, the input parameters (pitch angle $\mathcal{P}_{0.6}$, stiffness scaling factor s_E, and fiber angle β) are listed on the left side and the output parameters (see Eq. (8.6)) are listed on the right side.

Table 8.14: Results of optimization stage 1 of the anisotropic flexible propeller together with the results for the stiff KCS and PM 867 propeller

ID	$\mathcal{P}_{0.6}$	s_E	β	f_{fit}	T [kN]	η	k_{cav} [%]	T_σ [kN]	$k_{cav,\sigma}$ [%]
KCS	-	-	-	1.88	2028	0.758	0.712	48	0.356
PM 867	-	-	-	1.61	2028	0.714	0.377	20	0.041
339 (FSI 1)	-0.20	0.638	113.3°	1.594	2031	0.734	0.184	59.6	0.034
366	-0.53	0.575	145.1°	1.599	2030	0.734	0.230	42.7	0.066
376	0.46	0.689	102.0°	1.672	2059	0.731	0.168	72.5	0.033
388	-0.81	0.620	134.8°	1.675	2054	0.734	0.286	40.6	0.074
112	-1.38	0.705	130.3°	1.706	2022	0.735	0.376	50.0	0.091
145	0.47	0.666	100.2°	1.712	1993	0.733	0.210	78.6	0.036
162	0.47	0.670	100.2°	1.715	2002	0.733	0.209	89.1	0.040
147	-1.38	0.670	130.3°	1.722	1996	0.737	0.359	41.1	0.083
210	0.35	0.596	107.8°	1.724	1974	0.733	0.209	67.2	0.031
341	-0.65	0.762	113.0°	1.726	2082	0.733	0.232	55.3	0.062

It can be seen that only 2 individuals (339 and 366) perform better than the initial stiff geometry PM 867. This relatively poor performance can be attributed to numerical issues, and an improvement will be presented for stage 2 of the optimization. Nonetheless, the efficiency of the anisotropic flexible propellers is around 2 % better, and the cavitation area is reduced by more than half for some individuals. The relatively bad performance stems from the deviations in the target thrust and from an increased thrust variation. The deviation in the target thrust could simply be solved by computing more individuals, allowing for a fine-tuning of the factors influencing the thrust. Nevertheless, individuals 339 and 366 already meet a thrust that is certainly within the limits of the accuracy of the simulation. The increased thrust variation is caused by numerical problems concerning the singular panels, as mentioned in Sec. 7.2.

When looking at the input parameters, the parameter range shows some clear tendencies towards ranges for all of the parameters. The pitch variation $\mathcal{P}_{0.6}$ is for the best-performing individuals between -1.5° and 0.5, and the fiber angle β is between 100° and 150°. The stiffness scaling factor s_E lies between 0.5 and 0.8. The results for the stiffness scaling factor are higher than expected from the results presented in Sec. 8.2.4, as it was notable that a stronger bend-twist effect was possible with a lower stiffness scaling factor. The problem with simulations using a lower stiffness scaling factor is that many of them fail due to singular panels with unrealistically high-pressure values. The modified tip rake geometry increases this problem in comparison with the previously used KCS geometry. When the simulation fails, the fitness function is set to a predefined penalty value of 110.

Due to this penalty, the optimizer tends to avoid simulations with a lower stiffness scaling factor.

To reduce this problem, the filter introduced in Sec. 7.2 is improved by excluding extreme pressure values that deviate strongly from the neighboring panels. Furthermore, the penalty for failed simulations is decreased from 110 to 10 in stage 2.

The parameter set for stage 2 of the optimization is narrowed to a range between -3.5° and 2° for the pitch variation, between 0.1 and 0.7 for the stiffness scaling factor, and between 90° and 180° for the fiber orientation. In this stage, 424 individuals are computed on the HPC of TUHH. The results for the 10 best individuals are listed in Tab. 8.15.

Table 8.15: Results of optimization stage 2 of the anisotropic flexible propeller together with the results for the stiff KCS and PM 867 propeller

ID	$\mathcal{P}_{0.6}$	s_E	β	f_{fit}	T [kN]	η	k_{cav} [%]	T_σ [kN]	$k_{cav,\sigma}$ [%]
KCS	-	-	-	1.88	2028	0.758	0.712	48	0.356
PM 867	-	-	-	1.61	2028	0.714	0.377	20	0.041
FSI 1	-0.20	0.638	113.3°	1.594	2031	0.734	0.184	59.6	0.034
249 (FSI 2)	0.67	0.306	167.8°	1.531	2025	0.726	0.139	26.4	0.051
319	0.72	0.303	170.0°	1.541	2024	0.725	0.136	30.9	0.048
197	0.23	0.376	163.9°	1.542	2026	0.728	0.157	29.5	0.053
279	0.42	0.347	162.4°	1.542	2032	0.727	0.145	31.6	0.051
67	0.26	0.367	160.5°	1.545	2021	0.728	0.148	29.0	0.051
106	0.22	0.393	157.9°	1.551	2034	0.728	0.150	32.5	0.052
224	0.86	0.280	180.0°	1.554	2021	0.723	0.151	28.6	0.053
223	-0.03	0.447	154.6°	1.556	2040	0.729	0.169	24.4	0.053
328	0.81	0.297	164.7°	1.556	2033	0.725	0.132	40.2	0.045
200	0.26	0.376	160.5°	1.560	2033	0.728	0.154	36.4	0.053

The modification of the pressure filter leads, in combination with the lower penalty for failed simulations to stiffness scaling factors between 0.28 and 0.45, while the stiffness scaling factor was almost twice as high in the previous stage. With these improvements, an even smaller cavitation area is possible – with an additional reduction of 25 % for the best individuals of stage 2 compared to the FSI 1 propeller. This comes at the expense of slightly lower efficiency. The efficiency of the best individuals in stage 2 is still significantly better than that of the stiff PM 867 propeller.

With the FSI simulation, it is also possible to analyze the stress inside the propeller for new designs such as the FSI 2 propeller during the full loading cycle. A visualization of the FSI 2 propeller under loading is shown in Fig. 8.19. When FSI simulations are used, the stress or strain could also be used as a secondary condition in the optimization to obtain a failure-safe design. With equivalent stresses (see Eq. 2.69) of more than 100 MPa, the FSI 2 propeller already exhibits a high load under standard operating conditions.

In order to compare the performance of flexible propellers that were optimized based on the PM 867 geometry against flexible propellers optimized based on the KCS propeller geometry, another optimization is performed. The basic geometry for this is not the tip rake geometry used so far in the optimization, but the initial geometry of the KCS propeller visible in Fig. 8.1 and 8.2. The geometry of the KCS propeller remains unchanged in this optimization, and only the fiber angle and the stiffness scaling factor are optimized. The fiber angle is varied between 0° and 180°, and the stiffness scaling factor is varied between

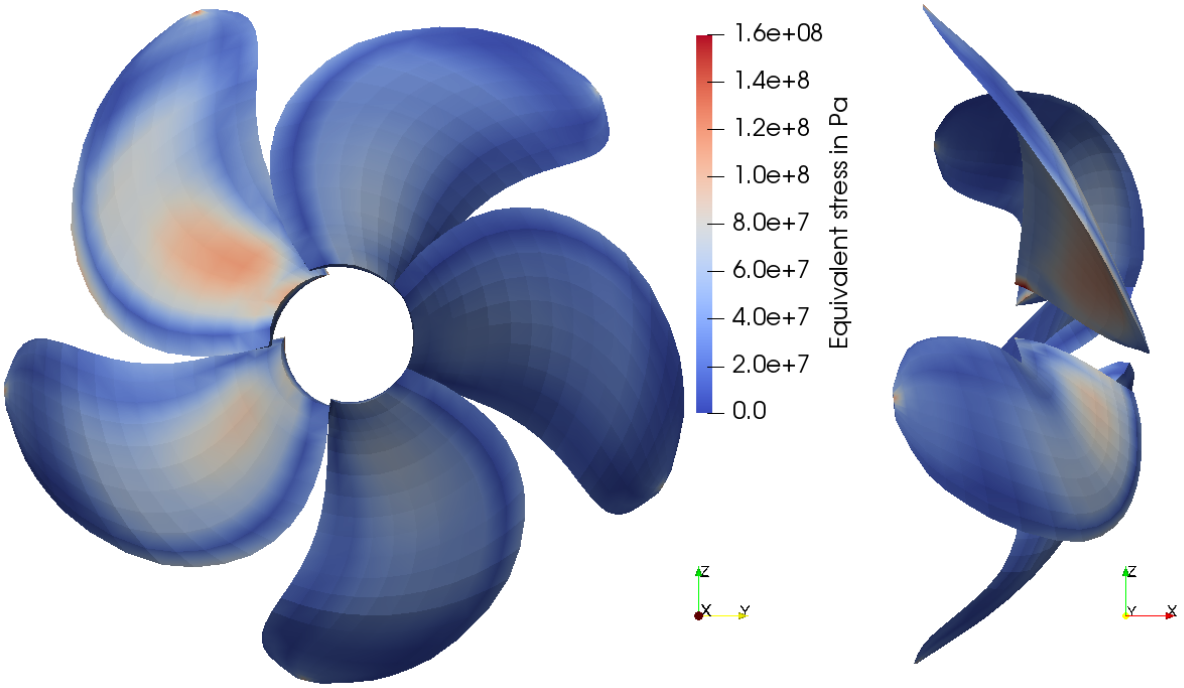


Figure 8.19: Equivalent stress of FSI 2 propeller.

0.1 and 1.4. A total number of 157 individuals are simulated for this optimization. The lower number of individuals should not be a problem since only two input parameters are optimized.

The results for the 10 best individuals are listed in Tab. 8.16 together with the previously introduced propellers. The best individual from this optimization is labeled FSI KCS.

Table 8.16: Results of the optimization of the anisotropic flexible KCS propeller

ID	$\mathcal{P}_{0.6}$	s_E	β	f_{fit}	T [kN]	η	k_{cav} [%]	T_σ [kN]	$k_{cav,\sigma}$ [%]
KCS	-	-	-	1.88	2028	0.758	0.712	48	0.356
PM 867	-	-	-	1.61	2028	0.714	0.377	20	0.041
FSI 1	-0.20	0.638	113.3°	1.594	2031	0.734	0.184	59.6	0.034
FSI 2	0.67	0.306	167.8°	1.531	2025	0.726	0.139	26.4	0.051
138 (FSI KCS)		0.535	112.6°	1.454	2035	0.755	0.140	10.8	0.048
146		0.535	121.5°	1.454	2023	0.755	0.153	11.5	0.043
87		0.535	121.1°	1.457	2025	0.755	0.154	12.5	0.050
121		0.542	121.5°	1.457	2027	0.755	0.159	13.1	0.052
82		0.492	112.6°	1.472	2003	0.756	0.118	12.1	0.031
126		0.528	102.6°	1.475	2021	0.755	0.151	10.5	0.079
145		0.525	101.0°	1.480	2023	0.754	0.159	12.2	0.082
114		0.585	126.0°	1.481	2023	0.755	0.181	16.1	0.048
133		0.494	121.2°	1.483	2000	0.756	0.132	11.7	0.034
107		0.505	106.0°	1.485	2001	0.755	0.123	13.8	0.038

Surprisingly, the FSI KCS propeller even performs a little bit better than the FSI 2 propeller for most of the performance parameters. The efficiency of the FSI KCS propeller

is around 3% higher than the FSI 2 propeller. The other performance parameters are quite similar to the FSI 2 propeller.

The better performance of the FSI KCS propeller could to a certain extent be caused by higher numerical oscillations of the optimized tip rake shape of the FSI 2 propeller. The shape of the FSI 2 propeller leads to more elements with sharp angles, and the tip is exposed to higher deformations.

To reduce this problem and allow for a better comparison of the different propellers, a moving mean filter is added to exclude high-frequency oscillation in the computation of the thrust variation in post-processing. The high-frequency oscillations due to singular panels are only apparent in the FSI simulation and cause fluctuations of around ±3 % of the thrust (for the FSI 2 propeller). Since the FSI propellers are already optimized to have a low thrust variation, the fluctuations dominate the contribution to the thrust variation in the result function if they are unfiltered.

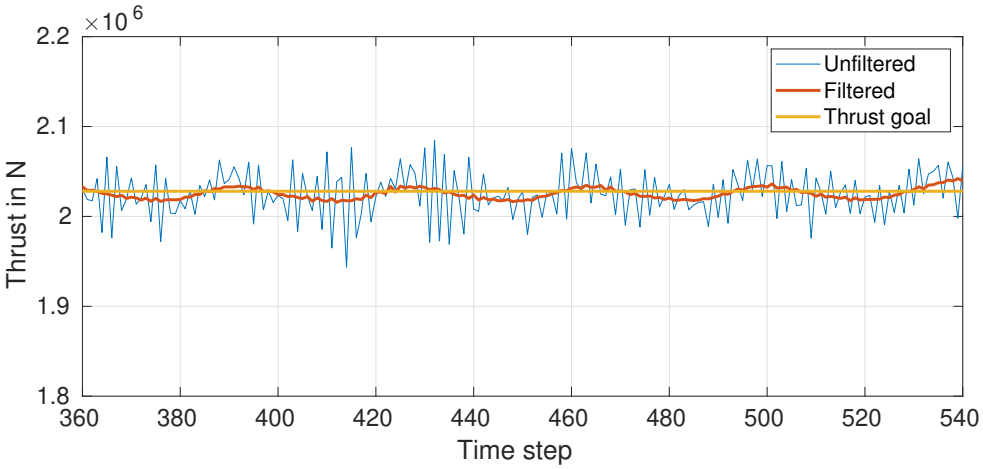


Figure 8.20: Filtering of thrust variation shown on FSI 2 propeller.

Fig. 8.20 illustrates the filtering of the thrust for the FSI 2 propeller. The moving mean filter averages the thrust over a window of 16 time steps and is, hence, suitable to reduce the high-frequency oscillations caused by singular panels. After the filtering, the natural frequency of the propeller FSI 2 is clearly visible.

In Fig. 8.21, the result functions of all propellers introduced so far are shown in a bar plot. The data used for this figure can also be found in the appendix in Tab. A.1. The different colors indicate the contribution of the performance parameters to the fitness function. For the FSI propellers, the filtered thrust variation is introduced in addition to the other parameters. The filtered thrust variation is the part of the thrust variation that is reduced by the moving mean filter. Hence, the filtered thrust variation represents the part of the thrust variation caused by numerical errors.

The higher thrust variation of the FSI 1 and FSI 2 propellers is reduced by the moving mean filter. While the filtered thrust variation of the FSI KCS propeller is 5.4 kN, the thrust variation of the FSI 2 propeller is 6.1 kN. Therefore, the thrust variations of the FSI 2 and the FSI KCS propeller are closer to each other. The thrust variation of the FSI 1 propeller is (with 16.6 kN) more than twice as high, but still lower than the stiff propellers. Since the filtering was not applied to the stiff propellers, the comparability of the thrust variation between the stiff and flexible propellers is limited.

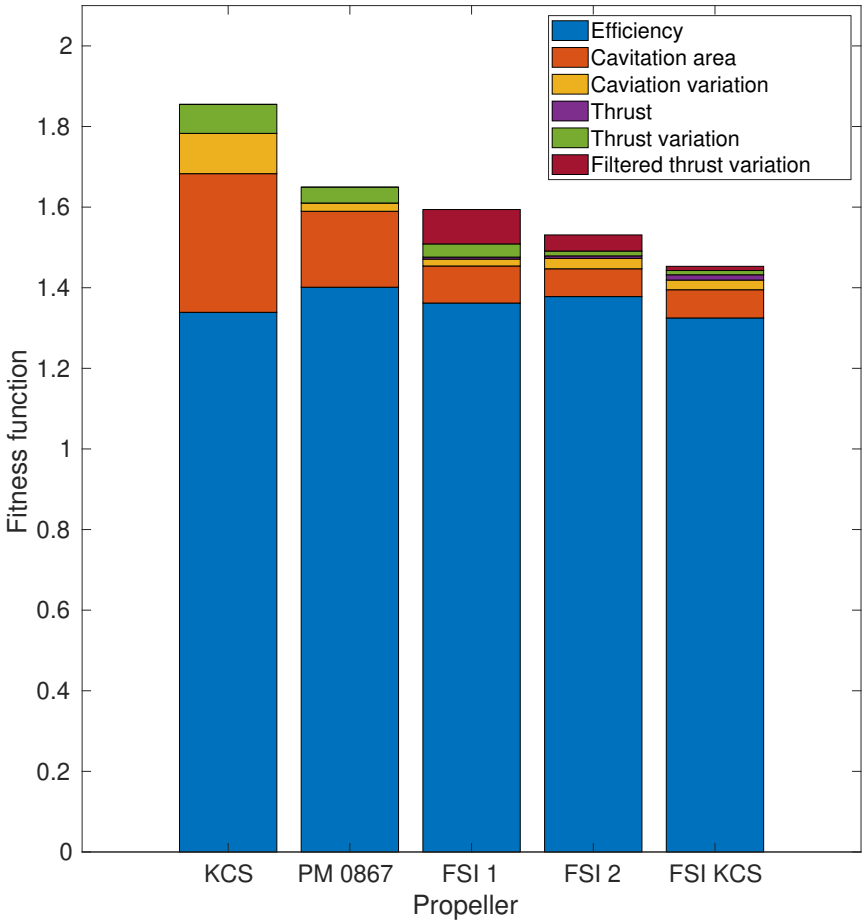


Figure 8.21: Comparison of the result functions.

The FSI KCS propeller and the FSI 2 propeller show quite similar results for cavitation, cavitation variation, and thrust variation. Despite the good cavitation and thrust properties, the efficiency of the FSI KCS propeller is almost as high as the efficiency of the initial stiff KCS propeller, which leads to a better overall performance. Since the cavitation effects in the tip vortex are not fully modeled here, the tip rake propellers could still have a higher performance compared with the KCS geometry due to reduced tip vortex cavitation.

The numerical fluctuations could cause some doubt about the accuracy of the results. The optimization procedure presented here penalizes all fluctuations with a higher fitness function. Therefore, the procedure also optimizes in favor of designs with lower numerical fluctuations, which should increase the accuracy of the result for the optimized propellers. This could mean, on the other hand, that better propellers in the range of the input parameter exist – causing stronger numerical errors. These propellers might be neglected due to a numerically increased high thrust variation. Propellers with a lower stiffness scaling factor would probably be most affected by this. Despite this, the optimized propellers already show a significantly better performance than the initial KCS propeller.

Optimizations in Dakota with multiple fiber directions in alternating layers do not lead to an overall better performance for the fitness function presented in the beginning. Still, these setups might be beneficial to increase the robustness of the propeller against failure.

In conclusion, the FSI effects seen here can be quite beneficial to improve the performance of the propeller. The focus of the presented optimization was to reduce cavitation, cavitation

variation, and thrust variation, in order to reduce noise emissions and undesired structural vibrations. In total, approximately 1000 FSI simulations were performed for the different optimizations. A high reduction of more than 60% in the cavitation area compared to the PM 867 propeller was obtained with the FSI 2 and the FSI KCS design. Compared to the initial stiff KCS propeller, a reduction of approximately 80% in the cavitation area was achieved. Also, a significantly lower thrust and cavitation variation compared to the initial KCS propeller was obtained. Despite this, an improvement in the efficiency of up to 2% is visible (compared to the stiff PM 867 propeller) with the load-dependent twist due to the bend-twist effect. If the focus of the fitness function were shifted more toward efficiency, a significant improvement in efficiency could be possible as well. An analysis of the noise emissions of the FSI 2 propeller compared to the KCS propeller can be found in Gonzalés et al. [116].

The material optimization of the KCS Propeller geometry with an unchanged pitch showed that the flexible anisotropic blade seems to be sufficient to reduce the cavitation to a similar degree as the FSI 2 propeller. Despite this, the initial efficiency of the stiff KCS propeller was maintained. The conclusion from this could be that a combination of the tip rake geometry with the bend-twist effect does lead to a further improvement (in the reduction of cavitation) compared to a design that only uses the bend-twist effect. Here, the flexibility and the bend-twist effect seem to be sufficient to reduce cavitation – and the reduction is achieved without a loss of efficiency.

A possible explanation for the remaining cavitation area is that the reduction of the very last part of the cavitation does, in comparison with the loss of efficiency that comes with it, not lead to a lower result function. If this is the case, the very last part of cavitation could be erased by a higher penalty for the small remaining cavitation area.

Overall, the coupled FSI simulations enable a significant design improvement. The accurate capturing of complex effects such as the bend-twist of flexible anisotropic propellers under realistic flow conditions would currently not be possible with other methods. With the increasing requirements for propeller designers and the new materials used in construction, the importance of considering FSI effects will likely increase further in the future.

9 Summary and outlook

This work demonstrated the benefit of partitioned fully-coupled FSI simulations applied in the development of complex maritime applications. To this end, the thesis starts with an introduction to the theoretical basis of FSI: structural mechanics, fluid mechanics, and the coupling of both. In this, a focus has been on the theoretical aspects that are necessary for later applications, such as static condensation and anisotropic material formulation in the structural mechanics chapter and propeller parameters in the fluid mechanics chapter.

In the main part of this thesis, four different application cases were presented to demonstrate the benefit of FSI simulations for maritime applications. Two of the application cases stem from the field of renewable energy production. Increasing demand for renewable energy production and limited space on land makes offshore renewable energy production increasingly important. However, offshore renewable energy production poses several new challenges, as shown in the FWT example.

As with most of the other simulations in this work, the FWT was simulated with *comana* – in a coupled simulation with the BEM solver *pan*MARE and the structural solver ANSYS. The benefit of this combination is a good trade-off between solution accuracy and computational time. For FSI simulations, a reduction in computational time is essential since the partitioned coupling approach requires multiple evaluations of the structural and the fluid field in every time step. To this end, static condensation was employed for the FWT. The FWT has been thoroughly analyzed in two different scenarios: in an extreme wave scenario and in regular operating conditions. The extreme wave scenario demonstrated the significant capability of the approach to identify improvements in the design, such as the necessary reinforcement of the platform. This is enabled by the accurate solution of the complete interaction between the fluid and structural field, which allows a direct analysis of the stress and deformation inside the structure in every time step. The analysis of the operating conditions of the FWT showed effects such as an undesired rotation of the FWT due to an oscillatory movement of the wind turbine blades interacting with the varying wind forces. With simulation approaches that do not consider FSI, this effect could not be captured.

The WEC simulation was the only application case presented here that was simulated with the FVM instead of the BEM on the fluid side of the simulation. For a smaller example like the WEC that does not include rotational movement, the FVM is a good alternative, also for application-related examples. Here, the FVM offers the benefit of accurately capturing added mass effects for different wave frequencies.

The two last cases are strongly related to each other, as both applications are turbo machine rotors that are exposed to a complex water flow. The submersible mixer is used in this work as a validation to prove the applicability and accuracy of the FSI approach for complex application-related examples. A challenge in this FSI setup, which was also apparent in the propeller cases, was singular panels in the BEM solver resulting from the strongly distorted panels. Managing these mesh deformations is a challenge that is inherent to the FSI approach since the deformations of the structure are bound to lead

to deformed meshes. In an uncoupled approach, the panels of the BEM solver keep the same shape over the whole simulation. In the presented FSI simulations, the deviations caused by the singular panels are controlled sufficiently well by different filters that exclude unrealistic extreme pressure values from the solution. Not only in the BEM but also in the simulation of the WEC with the FVM, a strong mesh deformation was apparent for larger wave heights, and a modification to the spar of the submerged body was used to reduce the mesh deformation. Despite these challenges, the extensive validation of the submersible mixer simulation showed that global variables such as thrust and torque coefficients, as well as local variables such as strains, can be predicted accurately for realistic flows on flexible rotors with a complex multilayered structure. The verification of the simulation with refined structural meshes showed that the spatial discretization error is sufficiently small. Validation and verification of the approach for application-related problems are essential for FSI simulations – given the complexity of the approach, which provides ample possibilities for errors.

The last and most important application case in this work is the ship propeller. The basis for this example is the flexible KCS propeller, which is investigated in detail in the first part of the chapter on ship propellers. The investigation delivered detailed insight into the impact of the flexibility on performance parameters of the propeller, such as thrust, torque, and efficiency. Furthermore, the deformation of the isotropic flexible KCS propeller was analyzed. To increase the solution accuracy, a corotational approach was developed that only transfers displacements to the fluid solver that are caused by deformation. In this way, it was possible to get rid of an interpolation error that was already observed in the FWT simulation. In addition to this, the influence of the coupling tolerance was analyzed to obtain a good trade-off between solution accuracy and computational time. A comparison between the different advanced convergence acceleration methods led to further improvements in the coupling, highlighting the impact of the convergence acceleration method on the computational effort and the stability of the simulation. This comparison was performed under realistic ship wake inflow conditions that led to varying loads on the propeller blades over each rotation.

The second part of the chapter on ship propellers aimed at optimizing the flexible anisotropic KCS propeller with an evolutionary optimization algorithm. For the modeling of the anisotropic KCS propeller, a transversely isotropic model of a CFRP material was defined in which the stiffness and fiber direction could be varied. The chosen material model was designed in such a way that it could take advantage of the bend-twist coupling, leading to efficiency improvements for the propeller under realistic inflow conditions. An improved stiff propeller with a modified tip rake geometry resulting from a geometry optimization was introduced. It showed a significantly reduced cavitation area, cavitation variation, and thrust variation. The improved stiff propeller was further optimized based on the previously described material model and modifications of the propeller pitch. The aim of the optimization was a further reduction in cavitation and a higher efficiency while maintaining the thrust goal. Also, cavitation and thrust variation were further minimized. Finally, the optimization of the material was also applied to the original KCS propeller geometry. The improvements in solution accuracy and computation time described in the previous section allowed for an optimization based on approximately 1000 individual FSI simulations of the full propeller. With fully-coupled FSI simulations, it is possible to not only evaluate the performance parameters, but also to conduct a highly accurate and

detailed stress analysis of the influence of the complex and varying loading on the ship propellers.

As a result, a propeller design was obtained that had a 2% higher efficiency and a 60% lower cavitation area than the optimized stiff propeller. Furthermore, the thrust was maintained, and both cavitation and thrust variation were significantly reduced. In the optimization of the anisotropic flexible propeller based on the KCS geometry, a similar improvement in cavitation area, cavitation variation, and thrust variation was obtained, while the efficiency and thrust remained almost unchanged.

Altogether, the results showcase the huge benefits that FSI simulations offer in the design of maritime structures. For an existing design of a flexible structure, FSI simulations can be used to accurately predict the behavior of the structure in complex and highly dynamic loading states, as shown on the FWT example. Based on the results, it is possible to propose and investigate potential improvements in the design. This can save significant costs that would usually arise in the design and testing of large-scale prototypes.

When it comes to improving existing designs, the FSI approach can be employed to adapt the dynamic deformation of the structure to a desired outcome, as shown for the ship propeller. Developing an anisotropic flexible structure that deforms in an optimal way under loading would be very challenging and expensive with other techniques. The partitioned approach enables the use of highly specialized and advanced codes with minimal modifications. As a consequence, it is possible to use, e.g., the static condensation and anisotropic material model formulation directly in ANSYS. In *pan*MARE, the modeling of the propeller wake with panels is possible in the same way as in the rigid body simulation of the propeller. The disadvantage of the presented approach is the high computational cost (due to the repeated solution of both subfields).

An important focus of this work was the verification and validation of the FSI simulation results for real-world applications from the maritime industry. The proof that the developed methods produce reliable results also for complex problems is essential for all simulation techniques.

In the future, it would be desirable to achieve a further reduction in computational time – as the considerable computational cost and the long simulation time (in the range of days and weeks) still hamper the industrial use of large-scale FSI simulations for real-world applications. With potentially increasing computational power in the future, this might become less of a problem.

To enable a more accurate simulation of the fluid, large simulations of maritime applications with the FVM are promising, as they are capable of capturing effects such as drag, turbulence, and vortex shedding more accurately. However, due to the requirement that the entire volume has to be meshed instead of only the boundary, as in the BEM, the computational cost is still prohibitively high for most realistic large-scale application cases if a sufficiently high mesh resolution is used. Even when using RANS equations in the FVM, the mesh should be able to resolve the boundary layer of the flow, which currently leads to enormous computational cost in combination with FSI if used for large-scale problems. Furthermore, problems with high mesh deformations in regions where a strong relative motion between the different structural parts occurs must be solved for highly dynamic simulations. Remeshing strategies could serve to handle high mesh deformations, even though they increase the computational cost.

A possibility to reduce the computational effort for FSI simulations with the FVM on the fluid side could be a multi-level approach following the partitioned approach: In the

region where drag, turbulence, and vortex shedding are dominant, the FVM could be used in the FSI simulation. In the other regions of the domain, the BEM could be used and coupled on the interface with the FVM region. In doing so, the computational cost could be reduced significantly.

In view of the application cases, further development of the presented designs is desirable. If a prototype or full-scale model of the presented FWT (or a similar one) is to be built, the presented simulation setup should be further validated based on the envisaged measurements on the FWT. Similarly, further validations of the WEC could be performed based on measurements performed on a prototype.

The presented ship propeller optimization should be extended to a more realistic cavitation and noise model to directly optimize the sound emissions instead of the presented indirect optimization by cavitation area and variation reduction. If the detailed production procedure for the CFRP ship propeller is defined in cooperation with a producer, the optimization of the propeller can be adapted to this procedure. Subsequently, the input parameters of the optimization could be defined in such a way that they are in accord with a realistic production procedure. Following another optimization based on a model that considers this production procedure, an optimized propeller could be built and investigated experimentally.

A Appendix

A.1 Ship propeller – simulation results

Table A.1: Results of the optimization of the KCS propeller - data for Fig. 8.21

ID	f_{fit}	T [kN]	η	k_{cav} [%]	T_σ [kN]	T_σ [kN, fil]	$k_{cav,\sigma}$ [%]
KCS	1.88	2028	0.758	0.712	48	-	0.356
PM 867	1.61	2028	0.714	0.377	20	-	0.041
FSI 1	1.594	2031	0.734	0.184	59.6	16.6	0.034
FSI 2	1.531	2025	0.726	0.139	26.4	6.1	0.051
FSI KCS	1.454	2035	0.755	0.140	10.8	5.4	0.048

Table A.2: Results for varying scaling factor and fiber angle of anisotropic ship propeller.

scaling factor	fiber angle	thrust coeff.	torque coeff.	efficiency	cav. area [%]
stiff	stiff	0.132	0.022	0.77	<0.01
0.1	0°	0.109	0.018	0.79	<0.01
0.2	0°	0.116	0.019	0.78	<0.01
0.5	0°	0.123	0.020	0.77	<0.01
1	0°	0.127	0.021	0.77	<0.01
0.1	10°	0.118	0.019	0.80	0.03
0.2	10°	0.120	0.020	0.78	<0.01
0.5	10°	0.125	0.021	0.77	<0.01
1	10°	0.128	0.021	0.77	<0.01
0.1	20°	0.135	0.022	0.80	0.07
0.2	20°	0.129	0.021	0.78	0.02
0.5	20°	0.129	0.021	0.77	<0.01
1	20°	0.130	0.021	0.77	<0.01
0.2	40°	0.201	0.035	0.73	0.37
0.5	40°	0.149	0.025	0.77	0.07
1	40°	0.138	0.023	0.77	<0.01
0.2	60°	0.217	0.049	0.56	2.12
0.5	60°	0.168	0.028	0.75	0.14
1	60°	0.147	0.024	0.77	0.06
0.1	80°	0.055	0.010	0.69	0.42
0.2	80°	0.116	0.021	0.70	0.47
0.5	80°	0.144	0.024	0.76	0.04
1	80°	0.140	0.023	0.77	<0.01
0.1	100°	0.051	0.010	0.68	0.23
0.2	100°	0.090	0.015	0.75	0.16
0.5	100°	0.129	0.021	0.77	<0.01
1	100°	0.134	0.022	0.77	<0.01
0.1	120°	0.066	0.011	0.73	0.20
0.2	120°	0.102	0.017	0.75	0.14
0.5	120°	0.129	0.021	0.77	<0.01
1	120°	0.133	0.022	0.77	<0.01
0.1	140°	0.076	0.013	0.75	0.16
0.2	140°	0.104	0.017	0.77	0.08
0.5	140°	0.122	0.020	0.77	<0.01
1	140°	0.127	0.021	0.77	<0.01
0.1	160°	0.093	0.015	0.78	0.06
0.2	160°	0.109	0.018	0.78	<0.01
0.5	160°	0.121	0.020	0.77	<0.01
1	160°	0.126	0.021	0.77	<0.01
0.1	170°	0.102	0.016	0.79	0.02
0.2	170°	0.112	0.018	0.78	<0.01
0.5	170°	0.122	0.020	0.77	<0.01
1	170°	0.126	0.021	0.77	<0.01

Bibliography

[1] N. Panwar, S. Kaushik, S. Kothari, Role of renewable energy sources in environmental protection: A review, Renewable and Sustainable Energy Reviews 15 (3) (2011) 1513–1524. doi:10.1016/j.rser.2010.11.037.

[2] D. Y. Leung, Y. Yang, Wind energy development and its environmental impact: A review, Renewable and Sustainable Energy Reviews 16 (1) (2012) 1031–1039. doi:10.1016/j.rser.2011.09.024.

[3] H. K. Jacobsen, P. Hevia-Koch, C. Wolter, Nearshore versus offshore: comparative cost and competitive advantages, International Association for Energy Economics (2016).

[4] Class NK, Nakashima Propeller, ClassNK and Nakashima Propeller Announce World's First Installation of a CFRP Propeller on a Merchant Vessel, https://www.classnk.or.jp/hp/en/hp_news.aspx?id=1043&type=press_release&layout=1, press release, accessed on 5th July 2023 (August 2014).

[5] Z. Liu, Y. L. Young, Utilization of deformation coupling in self-twisting composite propellers, in: Proceedings of 16th International Conference on Composite Materials, Kyoto, Japan, 2007, pp. 8–13.

[6] Y. Bazilevs, K. Takizawa, T. Tezduyar, Computational fluid-structure interaction: methods and applications, Wiley Series in Computational Mechanics, John Wiley & Sons, 2013. doi:10.1002/9781118483565.

[7] H. Bungartz, M. Schäfer (Eds.), Fluid-Structure Interaction, Modelling, Simulation and Optimisation, Vol. 53 of Lecture Notes in Computational Science and Engineering, Springer, 2006.

[8] T. Richter, Fluid-structure interactions: models, analysis and finite elements, Vol. 118, Springer, 2017. doi:10.1007/978-3-319-63970-3.

[9] W. Joppich, M. Kürschner, MpCCI — a tool for the simulation of coupled applications, Concurrency and computation: Practice and Experience 18 (2) (2006) 183–192. doi:10.1002/cpe.913.

[10] M. Heil, A. L. Hazel, J. Boyle, Solvers for large-displacement fluid–structure interaction problems: segregated versus monolithic approaches, Computational Mechanics 43 (1) (2008) 91–101. doi:10.1007/s00466-008-0270-6.

[11] H.-J. Bungartz, F. Lindner, B. Gatzhammer, M. Mehl, K. Scheufele, A. Shukaev, B. Uekermann, preCICE – a fully parallel library for multi-physics surface coupling, Computers & Fluids 141 (2016) 250–258, Advances in Fluid-Structure Interaction. doi:10.1016/j.compfluid.2016.04.003.

[12] S. Chimakurthi, S. Reuss, M. Tooley, S. Scampoli, ANSYS workbench system coupling: a state-of-the-art computational framework for analyzing multiphysics problems, Engineering with Computers 34 (04 2018). doi:10.1007/s00366-017-0548-4.

[13] J. Degroote, K.-J. Bathe, J. Vierendeels, Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction, Computers & Structures 87 (2009) 793–801. doi:10.1016/j.compstruc.2008.11.013.

[14] R. Haelterman, A. Bogaers, K. Scheufele, B. Uekermann, M. Mehl, Improving the performance of the partitioned QN-ILS procedure for fluid-structure interaction problems: Filtering, Computational Mechanics 171 (Supplement C) (2016) 9–17. doi:10.1016/j.compstruc.2016.04.001.

[15] J. Yan, A. Korobenko, X. Deng, Y. Bazilevs, Computational free-surface fluid–structure interaction with application to floating offshore wind turbines, Computers & Fluids 141 (2016) 155–174. doi:10.1016/j.compfluid.2016.03.008.

[16] M.-C. Hsu, Y. Bazilevs, Fluid–structure interaction modeling of wind turbines: simulating the full machine, Computational Mechanics 50 (2012) 821–833. doi:10.1007/s00466-012-0772-0.

[17] Y. Bazilevs, M.-C. Hsu, M. Scott, Isogeometric fluid–structure interaction analysis with emphasis on non-matching discretizations, and with application to wind turbines, Computer Methods in Applied Mechanics and Engineering 249-252 (2012) 28–41, higher Order Finite Element and Isogeometric Methods. doi:10.1016/j.cma.2012.03.028.

[18] Y. Zhai, H. Zhao, X. Li, W. Shi, Hydrodynamic responses of a barge-type floating offshore wind turbine integrated with an aquaculture cage, Journal of Marine Science and Engineering 10 (7) (2022). doi:10.3390/jmse10070854.

[19] J. Kumar, F.-H. Wurm, Bi-directional fluid–structure interaction for large deformation of layered composite propeller blades, Journal of Fluids and Structures 57 (2015) 32–48. doi:10.1016/j.jfluidstructs.2015.04.007.

[20] A. Lass, Simulation der Rotordynamik gehäuseloser Strömungsmaschinensysteme mit flexiblen Rotorblättern im Zeitbereich - Entwicklung eines Modellierungsansatzes und experimentelle Validierung anhand eines Tauchmotorrührwerkes, Ph.D. thesis, Universität Rostock (2019). doi:10.18453/rosdok_id00002967.

[21] Y. L. Young, Hydroelastic behavior of flexible composite propellers in wake inflow, in: Proceedings of 16th International Conference on Composite Materials, Kyoto, Japan, 2007, pp. 1–6.

[22] S. Rohde, P. Ifju, B. Sankar, D. Jenkins, Experimental testing of bend-twist coupled composite shafts, Experimental Mechanics 55 (2015) 1613–1625. doi:10.1007/s11340-015-0050-0.

[23] N. Mulcahy, B. Prusty, C. Gardiner, Flexible composite hydrofoils and propeller blades, in: International Maritime Conference, 2010, pp. 27–29.

[24] M. Bauer, M. Abdel-Maksoud, A three-dimensional panel method for the simulation of sheet cavitation in marine propeller flows, in: Proceedings in Applied Mathematics and Mechanics, Vol. 45, 2012, p. 1179–1184. doi:10.3182/20120215-3-AT-3016.00209.

[25] S. Netzband, C. Schulz, M. Abdel-Maksoud, A fully coupled simulation method for floating offshore wind turbine dynamics using a boundary element method in the time domain, in: 10th International Workshop on Ship and Marine Hydrodynamics IWSH 2017, 2017, pp. 1–15.

[26] M. König, L. Radtke, A. Düster, A flexible C++ framework for the partitioned solution of strongly coupled multifield problems, Computers & Mathematics with Applications 72 (2016) 1764–1789. doi:10.1016/j.camwa.2016.07.031.

[27] P. Wriggers, Nonlinear finite element methods, Springer-Verlag, 2008. doi:10.1007/978-3-540-71001-1.

[28] B. Szabó, I. Babuška, Finite element analysis, John Wiley & Sons, 1991.

[29] K.-J. Bathe, Finite element procedures, 2nd Edition, Prentice Hall, 2014.

[30] O. Zienkiewicz, R. Taylor, The finite element method – solid mechanics, 5th Edition, Vol. 2, Butterworth-Heinemann, 2000.

[31] J. Bonet, R. Wood, Nonlinear continuum mechanics for finite element analysis, Cambridge University Press, New York, 1997. doi:10.1017/cbo9780511755446.

[32] S. C. Cowin, Continuum mechanics of anisotropic materials, Springer, New York, NY, 2013. doi:10.1007/978-1-4614-5025-2.

[33] P. Vannucci, General anisotropic elasticity, in: Anisotropic Elasticity, Springer, 2018, pp. 19–73. doi:10.1007/978-981-10-5439-6.

[34] H. Schürmann, Konstruieren mit Faser-Kunststoff-Verbunden, 2nd Edition, VDI-Buch, Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2007. doi:10.1007/978-3-540-72190-1.

[35] Y. Li, J. Barbic, Stable orthotropic materials, in: V. Koltun, E. Sifakis (Eds.), Eurographics/ ACM SIGGRAPH Symposium on Computer Animation, The Eurographics Association, 2014, pp. 41–46. doi:10.2312/sca.20141121.

[36] J. E. Mardsen, T. J. R. Hughes, Mathematical foundations of elasticity, Prentice-Hall, 1993.

[37] P. Drabek, G. Holubova, Elements of partial differential equations, De Gruyter Textbook, de Gruyter, Berlin u.a., 2014. doi:10.1515/9783110316674.

[38] J. Stoer, R. Bulirsch, Introduction to numerical analysis, Springer, 2002. doi:10.1007/978-0-387-21738-3.

[39] J. Bonet, A. J. Gil, R. D. Wood, Nonlinear solid mechanics for finite element analysis: dynamics, Cambridge University Press, 2021. doi:10.1017/9781316336083.

[40] N. Newmark, A method of computation for structural dynamics, Journal of Engineering Mechanics (ASCE) 85 (1959) 67–94. doi:10.1061/jmcea3.0000098.

[41] J. Chung, G. Hulbert, A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method, Journal of Applied Mechanics 60 (1993) 371–375. doi:10.1115/1.2900803.

[42] D. Kuhl, M. A. Crisfield, Energy-conserving and decaying algorithms in nonlinear structural dynamics, International Journal for Numerical Methods in Engineering 45 (1999) 569–599. doi:10.1002/(sici)1097-0207(19990620)45:5<569::aid-nme595>3.0.co;2-a.

[43] S. Erlicher, L. Bonaventura, O. S. Bursi, The analysis of the generalized-α method for non-linear dynamic problems, Computational Mechanics 28 (2) (2002) 83–104. doi:10.1007/s00466-001-0273-z.

[44] T. J. Ypma, Historical development of the Newton–Raphson method, SIAM Review 37 (4) (1995) 531–551. doi:10.1137/1037125.

[45] E. Isaacson, H. B. Keller, Analysis of numerical methods, Dover publications, inc., 1994.

[46] U. Fischer (Ed.), Tabellenbuch Metall: mit Formelsammlung, 45th Edition, Europa-Fachbuchreihe für Metallberufe, Verl. Europa-Lehrmittel, Haan-Gruiten, 2011.

[47] ANSYS, ANSYS theory reference, release 19.2 (2018).

[48] J. Craig, R. Roy, M. C. Bampton, Coupling of substructures for dynamic analyses, AIAA Journal 6 (7) (1968) 1313–1319. doi:10.2514/3.4741.

[49] R. J. Guyan, Reduction of stiffness and mass matrices, AIAA Journal 3 (2) (1965) 380–380. doi:10.2514/3.2874.

[50] J. Ferziger, M. Peric, R. L. Street, Numerische Strömungsmechanik, 2nd Edition, Springer eBook Collection, Springer Vieweg, Berlin, Heidelberg, 2020. doi:10.1007/978-3-662-46544-8.

[51] J. Katz, A. Plotkin, Low speed aerodynamics, 2nd Edition, Cambridge aerospace series, Cambridge University Press, Cambridge, 2001. doi:10.1017/CBO9780511810329.

[52] H. Goldstein, C. Poole, J. Safko, Classical mechanics, American Association of Physics Teachers, 2002.

[53] A. R. Paschedag, CFD in der Verfahrenstechnik: allgemeine Grundlagen und mehrphasige Anwendungen, Wiley-VCH, Weinheim, 2004. doi:10.1002/3527603859.

[54] B. P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, Computer Methods in Applied Mechanics and Engineering 19 (1) (1979) 59–98. doi:10.1016/0045-7825(79)90034-3.

[55] S. F. Hoerner, Fluid-dynamic drag - practical information on aerodynamic drag and hydrodynamic resistance, Hoerner Fluid Dynamics, 1965. doi:10.1017/S0001924000034187.

[56] H. Herwig, B. Schmandt, Strömungsmechanik: Physik - mathematische Modelle - thermodynamische Aspekte, 3rd Edition, Lehrbuch, Springer Vieweg, Berlin Heidelberg, 2015. doi:10.1007/978-3-662-45069-7.

[57] C. Runge, Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten, Zeitschrift für Mathematik und Physik 46 (1901) 224–243.

[58] B. Gatzhammer, Efficient and flexible partitioned simulation of fluid-structure interactions, Ph.D. thesis, Technische Universität München (2014).

[59] A. de Boer, A. H. van Zuijlen, H. Bijl, Comparison of conservative and consistent approaches for the coupling of non-matching meshes, Computer Methods in Applied Mechanics and Engineering (2008). doi:10.1016/j.cma.2008.05.001.

[60] C. Ericson, Real-time collision detection, Elsevier, 2005.

[61] D. Shepard, A two-dimensional interpolation function for irregularly-spaced data, in: Proceedings of the 1968 23rd ACM national conference, 1968, pp. 517–524. doi:10.1145/800186.810616.

[62] M. König, Partitioned Solution Strategies for Strongly-Coupled Fluid-Structure Interaction Problems in Maritime Applications, Ph.D. thesis, Technische Universität Hamburg (2018). doi:10.51202/9783186351180.

[63] R. Krause, E. Rank, A fast algorithm for point-location in a finite element mesh, in: Computing, Vol. 57, 1996, pp. 49–63. doi:10.1007/bf02238357.

[64] L. Radtke, A partitioned solution approach for fluid-structure interaction problems in the arterial system, Ph.D. thesis, Technische Universität Hamburg (2020). doi:10.15480/882.3638.

[65] A. MacLeod, Acceleration of vector sequences by multi-dimensional Δ^2 methods, Communications in Applied Numerical Methods 1 (1986) 3–20. doi:10.1002/cnm.1630020409.

[66] A. C. Aitken, IV.—Studies in practical mathematics. V. on the iterative solution of a system of linear equations, Proceedings of the Royal Society of Edinburgh. Section A. Mathematical and Physical Sciences 63 (1950) 52–60. doi:10.1017/S008045410000697X.

[67] B. Irons, R. C. Tuck, A version of the aitken accelerator for computer implementation, International Journal for Numerical Methods in Engineering 1 (1969) 275–277.

[68] J. E. Dennis, R. B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, Society for Industrial and Applied Mathematics, 1996. doi:10.1137/1.9781611971200.

[69] C. G. Broyden, A class of methods for solving nonlinear simultaneous equations, Mathematics of Computation 19 (92) (1965) 577–593. doi:10.1090/s0025-5718-1965-0198670-6.

[70] C. Broyden, On the discovery of the "good Broyden" method, Mathematical programming 87 (2) (2000) 209–213. doi:10.1007/s101070050111.

[71] R. Haelterman, J. Degroote, D. Heule, J. Vierendeels, The quasi-Newton least squares method: A new and fast secant method analyzed for linear systems, SIAM Journal on Numerical Analysis 47 (2009) 2347. doi:10.1137/070710469.

[72] L. Radtke, A. Larena-Avellaneda, E. Debus, A. Düster, Convergence acceleration for partitioned simulations of the fluid-structure interaction in arteries, Computational Mechanics 57 (6) (2016) 901–920. doi:10.1007/s00466-016-1268-0.

[73] A. Düster, S. Kollmannsberger, AdhoC^4 – user's guide, Lehrstuhl für Computation in Engineering, TU München, Numerische Strukturanalyse mit Anwendungen in der Schiffstechnik, TU Hamburg-Harburg (2010).

[74] ANSYS, ANSYS academic research mechanical, release 19.2 (2018).

[75] H. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Computers in Physics 12 (6) (1998) 620–631. doi:10.1063/1.168744.

[76] S. Netzband, C. W. Schulz, U. Göttsche, D. F. González, M. Abdel-Maksoud, A panel method for floating offshore wind turbine simulations with fully integrated aero- and hydrodynamic modelling in time domain, Ship Technology Research 65 (3) (2018) 123–136. doi:10.1080/09377255.2018.1475710.

[77] L. Radtke, T. Lampe, M. Abdel-Maksoud, A. Düster, Partitioned simulation of multi-field problems – efficient and robust coupling of fluids and structures, Proceedings in Applied Mathematics and Mechanics 18 (2018) e201800424. doi:10.1002/pamm.201800424.

[78] B. Uekermann, H. J. Bungartz, B. Gatzhammer, M. Mehl, A parallel, black-box coupling algorithm for fluid-structure interaction, in: Proceedings of the 5th International Conference on Computational Methods for Coupled Problems in Science and Engineering, Ibiza, Spain, 2013, pp. 1–12.

[79] M. Mehl, B. Uekermann, H. Bijl, D. Blom, B. Gatzhammer, A. van Zuijlen, Parallel coupling numerics for partitioned fluid-structure interaction simulations, Computers & Mathematics with Applications 71 (4) (2016) 869–891. doi:10.1016/j.camwa.2015.12.025.

[80] J. Cruse, M. Abdel-Maksoud, A. Düster, A. Bockstedte, G. Haake, S. Siegfriedsen, Design of floating offshore wind turbine (FOWT) "SelfAligner", in: J. Park, D. Whang (Eds.), EKC 2019 Conference Proceedings. Science, Technology, and Humanity: Advancement and Sustainability, Springer, Singapore, 2021, pp. 55–67. doi:10.1007/978-981-15-8350-6_5.

[81] A. Düster, M. König, B. Wiegard, Fluid-Struktur-Interaktion und Optimierung einer schwimmenden Plattform für Offshore-Windenergieanlagen (FSIOpt), Abschlussbericht des Teilvorhabens innerhalb des BMWi-Verbundprojektes: Hydrodynamische und strukturmechanische Optimierung eines Halbtauchers für Offshore-Windenergieanlagen (HyStOH), Institut für Konstruktion und Festigkeit von Schiffen, Technische Universität Hamburg (2019). doi:10.2314/KXP:1686083203.

[82] B. Wiegard, L. Radtke, M. König, M. Abdel-Maksoud, A. Düster, Simulation of the fluid-structure interaction of a floating wind turbine, Ships and Offshore Structures 14 (2019) 207–218. doi:10.1080/17445302.2019.1565295.

[83] J. Lund, B. Wiegard, L. Radtke, M. König, A. Düster, Partitioned coupling of fluid-structure interaction for the simulation of floating wind turbines, in: Proceedings of 8th GACM Colloquium on Computational Mechanics for Young Scientists from Academia and Industry, Kassel, Germany, 2019, pp. 335–338.

[84] B. Wiegard, M. König, J. Lund, L. Radtke, S. Netzband, M. Abdel-Maksoud, A. Düster, Fluid-structure interaction and stress analysis of a floating wind turbine, Marine Structures 78 (2021) 102970. doi:10.1016/j.marstruc.2021.102970.

[85] S. Netzband, C. W. Schulz, M. Abdel-Maksoud, Self-aligning behaviour of a passively yawing floating offshore wind turbine, Ship Technology Research 67 (1) (2020) 15–25. doi:10.1080/09377255.2018.1555986.

[86] A. Arapogianni, A.-B. Genachte, R. Manzanas, J. Vergara, C. Castell, A. Rodriguez Tsouroukdissian, J. Korbijn, N. Bolleman, F. Huera-Huarte, F. Schuon, A. Ugart, J. Sandberg, V. Laleu, J. Maciel, A. Tunbjer, R. Roth, P. Gueriviere, P. Coulombeau, S. Jedrec, H. Grubel, Deep water. the next step for offshore wind energy, Tech. rep., European Wind Energy Association (2013).

[87] M. Hall, A. Goupee, Validation of a lumped-mass mooring line model with DeepCwind semisubmersible model test data, Ocean Engineering 104 (2015) 590–603. doi:10.1016/j.oceaneng.2015.05.035.

[88] JBO, DESPOW - "Design Entwicklung einer schwimmenden Plattform für eine selbstausrichtende Offshore Windenergieanlage": DESPOW-Abschlussbericht JBO (Stahlbaulicher Teil im HyStOH-Verbund) , Abschlussbericht des Teilvorhabens innerhalb des BMWi-Verbundprojektes: Hydrodynamische und strukturmechanische Optimierung eines Halbtauchers für Offshore-Windenergieanlagen (HyStOH), JÖRSS – BLUNCK – ORDEMANN GmbH (2020). doi:10.2314/KXP:1736881388.

[89] J. Simo, F. Armero, Geometrically non-linear enhanced strain mixed methods and the method of incompatible modes, International Journal for Numerical Methods in Engineering 33 (1992) 1413–1449. doi:10.1002/nme.1620330705.

[90] ANSYS, ANSYS element reference, release 19.2 (2018).

[91] S. Rezaei, A. Rahimi, J. Parvizian, S. Mansourzadeh, A. Düster, Dimensional optimization of a two-body wave energy converter using response surface methodology, Ocean Engineering 261 (2022) 112186. doi:10.1016/j.oceaneng.2022.112186.

[92] A. Rahimi, S. Rezaei, J. Parvizian, S. Mansourzadeh, J. Lund, R. Hssini, A. Düster, Numerical and experimental study of the hydrodynamic coefficients and power absorption of a two-body point absorber wave energy converter, Renewable Energy 201 (2022) 181–193. doi:10.1016/j.renene.2022.10.103.

[93] R. Hssini, Optimization of wave energy converter parameters using partitioned fluid-structure interaction simulations, Master thesis, Hamburg University of Technology (2021).

[94] B. Drew, A. Plummer, M. Sahinkaya, A review of wave energy converter technology, Proceedings of The Institution of Mechanical Engineers Part A-journal of Power and Energy 223 (2009) 887–902. doi:10.1243/09576509JPE782.

[95] K. Edwards, M. Mekhiche, et al., Ocean power technologies powerbuoy®: system-level design, development and validation methodology, in: Proceedings of the 2nd Marine Energy Technology Symposium, Seattle, USA, 2014, pp. 1–9.

[96] J. Weber, F. Mouwen, A. Parish, D. Robertson, Wavebob—research & development network and tools in the context of systems engineering, in: Proc. Eighth European Wave and Tidal Energy Conference, Uppsala, Sweden, Vol. 8, 2009, pp. 416–420.

[97] H. Weller, C. Greenshields, C. de Rouvray, OpenFOAM, release 2.3.1, http://www.openfoam.org, accessed on 5th July 2023 (2014).

[98] S. B. Pope, S. B. Pope, Turbulent flows, Cambridge university press, 2000. doi:10.1017/cbo9780511840531.

[99] W. Jones, B. Launder, The prediction of laminarization with a two-equation model of turbulence, International Journal of Heat and Mass Transfer 15 (2) (1972) 301–314. doi:10.1016/0017-9310(72)90076-2.

[100] B. Launder, D. Spalding, The numerical computation of turbulent flows, Computer Methods in Applied Mechanics and Engineering 3 (2) (1974) 269–289. doi:10.1016/0045-7825(74)90029-2.

[101] C. Hirt, B. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, Journal of Computational Physics 39 (1) (1981) 201–225. doi:10.1016/0021-9991(81)90145-5.

[102] H. Jasak, Dynamic mesh handling in OpenFOAM, in: 47th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition, 2009, p. 341. doi:10.2514/6.2009-341.

[103] H. Weller, C. Greenshields, C. de Rouvray, OpenFOAM – user guide, OpenFOAM Foundation (2014).

[104] L. E. Borgman, J. E. Chappelear, The use of the Stokes-Struik approximation for waves of finite height, in: 6th Conference on Coastal Engineering, 1958, pp. 252–280. doi:10.9753/icce.v6.16.

[105] P. Higuera, J. Lara, I. Losada, Realistic wave generation and active wave absorption for Navier-Stokes models: Application to OpenFOAM, Coastal Engineering 71 (2013) 102–118. doi:10.1016/j.coastaleng.2012.07.002.

[106] J. Falnes, Ocean Waves and Oscillating Systems: Linear Interactions Including Wave-Energy Extraction, Applied Mechanics Reviews 56 (1) (2003). doi:10.1115/1.1523355.

[107] J. Lund, D. Ferreira González, J. C. Neitzel-Petersen, L. Radtke, M. Abdel-Maksoud, A. Düster, Validation of a partitioned fluid-structure interaction simulation for turbo machine rotors, Ships and Offshore Structures 0 (0) (2022) 1–12. doi:10.1080/17445302.2022.2069389.

[108] P.-L. George, F. Hecht, É. Saltel, Fully automatic mesh generator for 3d domains of any shape, IMPACT of Computing in Science and Engineering 2 (3) (1990) 187–218. doi:10.1016/0899-8248(90)90012-Y.

[109] S. Keil, Dehnungsmessstreifen, 2nd Edition, Springer Vieweg, Wiesbaden, 2017. doi:10.1007/978-3-658-13612-3.

[110] SIMMAN - Workshop on verification and validation of ship manoeuvring simulation methods, Model data KCS propeller, http://www.simman2008.dk/KCS/container.html, website, accessed on 5th July 2023 (2008).

[111] J. Carlton, Chapter 9 - cavitation, in: Marine Propellers and Propulsion (Fourth Edition), fourth edition Edition, Butterworth-Heinemann, 2019, pp. 217–260. doi:10.1016/B978-0-08-100366-4.00009-2.

[112] T. Hino, K. G. A. Kenkyūjo, The Proceedings of CFD Workshop, Tokyo, 2005, National Maritime Research Institute, 2005.

[113] L. Schumaker, Spline Functions: Basic Theory, 3rd Edition, Cambridge Mathematical Library, Cambridge University Press, 2007. doi:10.1017/CBO9780511618994.

[114] U. Göttsche, Entwicklung einer numerischen Methode zur Vorhersage der hydroakustischen Schallabstrahlung von Schiffspropellern, Ph.D. thesis, Technische Universität Hamburg (2020). doi:10.15480/882.2953.

[115] J. Otzen, P. Esquivel, C. Simonsen, F. Stern, CFD and EFD Prediction of Added Powering of KCS in Regular Head Waves, in: 32nd Symposium on Naval Hydrodynamics, 2018, pp. 5–10.

[116] D. F. González, J. Lund, R. Bévand, A. Düster, M. Abdel-Maksoud, Optimization of ship propellers under consideration of the acoustic emission based on partitioned fluid-structure interaction simulations, submitted for publication (2024).

[117] L. Savio, L. Sileo, S. Kyrre Ås, A comparison of physical and numerical modeling of homogenous isotropic propeller blades, Journal of Marine Science and Engineering 8 (1) (2020) 21. doi:10.3390/jmse8010021.

[118] J. Carlton, Chapter 18 - propeller materials, in: J. Carlton (Ed.), Marine Propellers and Propulsion (Fourth Edition), fourth edition Edition, Butterworth-Heinemann, 2019, pp. 425–436. doi:10.1016/B978-0-08-100366-4.00018-3.

[119] S. Engelsmann, V. Spalding, S. Peters, Kunststoffe in Architektur und Konstruktion, Birkhäuser, Berlin, Boston, 2010. doi:10.1515/9783034611848.

[120] R. J. Pemberton, J. Graham-Jones, J. Summerscales, R. Pemberton, Marine Composites: Design and Performance, Elsevier, 2018.

[121] J. Lund, L. Radtke, A. Düster, D. Ferreira González, M. Abdel-Maksoud, Advanced methods for partitioned fluid-structure interaction simulations applied to ship propellers, in: Proceedings of the ASME 2022 41st International Conference on Ocean, Offshore and Arctic Engineering, Hamburg, Germany, 2022, pp. 1–9. doi:10.15480/882.4337.

[122] J. Carlton, Chapter 10 - propeller noise, in: J. Carlton (Ed.), Marine Propellers and Propulsion (Fourth Edition), fourth edition Edition, Butterworth-Heinemann, 2019, pp. 261–280. doi:10.1016/B978-0-08-100366-4.00010-9.

[123] J. Nocedal, S. Wright, Numerical optimization, Springer, 2006. doi:10.1007/978-0-387-40065-5.

[124] R. Storn, K. Price, Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359. doi:10.1023/A:1008202821328.

[125] C. Darwin, On the origin of species, D. Appleton and Company, 1859. doi:10.1093/owc/9780199219223.001.0001.

[126] A. Slowik, H. Kwasnicka, Evolutionary algorithms and their applications to engineering problems, Neural Computing and Applications 32 (2020) 12363–12379. doi:10.1007/s00521-020-04832-8.

[127] A. Pétrowski, S. Ben-Hamida, Evolutionary algorithms, Metaheuristics set, ISTE, London, 2017. doi:10.1002/9781119136378.

[128] B. Adams, W. Bohnhoff, K. Dalbey, M. Ebeida, J. Eddy, M. Eldred, R. Hooper, P. Hough, K. Hu, J. Jakeman, M. Khalil, K. Maupin, J. Monschke, E. Ridgway, A. Rushdi, T. Seidl, A. Stephens, L. Swiler, A. Tran, J. Winokur, Dakota, A Multilevel Parallel Object-Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis: Version 6.12 Developers Manual, Sandia National Laboratories, Albuquerque, NM (May 2022).
URL http://dakota.sandia.gov/documentation.html

[129] E. Praefke, T. Stoye, C. Abt, A generalized description of hydrodynamic parts based on aerodynamic profile sections, in: Fifth International Symposium on Marine Propulsors, Espoo, Finland, 2017, pp. 1–6.

[130] T. Stoye, F. Borasch, T. Lange, Final report of the research project HYKOPS "development of the framework and integration into a design environment", Tech. rep., Flensburger Schiffbau-Gesellschaft mbH & Co. KG; (2019). doi:10.2314/KXP:1743168721.

[131] S. P. Timoshenko, Theory of bending, torsion and buckling of thin-walled members of open cross section, Journal of the Franklin Institute 239 (4) (1945) 249–268. doi:10.1016/0016-0032(45)90161-X.

[132] C. H. Yoo, S. C. Lee, Chapter 5 - torsion in structures, in: Stability of Structures, Butterworth-Heinemann, Boston, 2011, pp. 245–301. doi:10.1016/B978-0-12-385122-2.10005-3.

[133] A. Hildebrandt, P. Sharma, A. Düster, S. Diebels, Experimental and numerical investigation of the deformation behaviour of cables and thin beam-like structures under multi-axial loading, Mathematics and Mechanics of Solids 27 (10) (2022) 2314–2337. doi:10.1177/10812865221114299.

[134] D. R. Lide, CRC handbook of chemistry and physics, Vol. 85, CRC press, 2004.

[135] J. C. Neitzel-Petersen, D. F. González, R. Gosda, M. Abdel-Maksoud, A two-stage optimization approach for propellers with unconventional blade shape in a wake field using BEM, in: Seventh International Symposium on Marine Propulsors, Wuxi, China, 2022, pp. 1–10.

[136] J. Dang, Improving cavitation performance with new blade sections for marine propellers, International shipbuilding progress 51 (4) (2004) 353–376.

[137] A. Amini, M. Reclari, T. Sano, M. Iino, M. Farhat, Suppressing tip vortex cavitation by winglets, Experiments in Fluids 60 (2019) 1–15. doi:10.1007/s00348-019-2809-z.