An advanced fluid-structure interaction software package for industrial problems in nuclear reactor applications

Chouchoulis, J.

Award date:
2018

Technische Universiteit Eindhoven University of Technology

Industrial and Applied Mathematics
Mathematics and Computer Science

An advanced Fluid-Structure Interaction software package for industrial problems in nuclear reactor applications

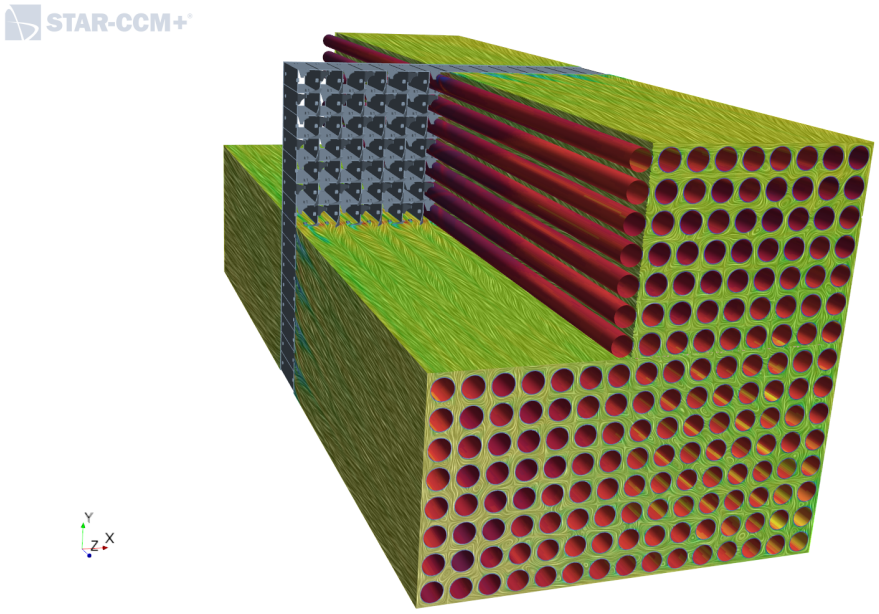
MSc Thesis

JEREMY CHOUCHOULIS

Supervisors: prof. dr. ir. Barry Koren Mathematics and Computer Science, CASA, TU/e
dr. ir. Afaque Shams Computational Physics for Solutions, NRG

Petten, November 2018

NRG



A display of the flow behaviour around a simplified fuel rod assembly produced at NRG by means of the STAR-CCM+ software.

Technische Universiteit Eindhoven University of Technology

An advanced Fluid-Structure Interaction software package for industrial problems in nuclear reactor applications

MSc Thesis

JEREMY CHOUCHOULIS

Supervisors: prof. dr. ir. Barry Koren Mathematics and Computer Science, CASA, TU/e
dr. ir. Afaque Shams Computational Physics for Solutions, NRG

Petten, November 2018

Abstract

Modernized civilization demands more energy supply than ever before. The nuclear industry aims to contribute largely in achieving this goal by investing a decent budget in research and development of the newer, more efficient generation of reactors. These nuclear reactors feature larger and more slender components, more intense thermal energies, higher velocities and densities of the coolant fluids resulting in more severe vibrations of the structural components. As a consequence the components that are developed will be more prone to Flow-Induced Vibrations (FIV). These vibrations can lead to unfavorable conditions such as corrosion cracking, fatigue problems, resulting in unsafe working conditions and substantial stand-still costs due to accidental failures. The diagnose of modal characteristics of specific nuclear reactor components, such as the fuel rod assembly is thus of major importance.

Qualified researchers from all over the world are being employed to ensure safe energy production, including mathematicians. At Nuclear Research and Consultancy Group (NRG), an in-house Fluid-Structure interaction (FSI) software package, called NRG-FSIFOAM, has been developed based on other open-source software packages to perform valuable investigation of FIV by means of the-state-of-the-art algorithms existing to date. Within NRG-FSIFOAM, a numerical method to predict pressure fluctuations in turbulent flow is implemented in order to study Turbulence-Induced Vibrations (TIV), which is called the pressure fluctuation model (PFM). PFM, being an extension to the U-RANS model, reduces computational costs greatly compared to high-fidelity solvers such as DNS and LES.

In this Master thesis project, NRG-FSIFOAM is validated by simulating FIV of bare and wire-wrapped rods in turbulent water and lead-bismuth eutectic flow. It is observed that compared with commercial software (STAR-CCM+) NRG-FSIFOAM generates good results by means of wall-function models and in general over-predicts the wall shear stress tensor in wall-resolved simulations with the $k-\omega$ SST model. Modal characteristics from PFM simulations are in good agreement with results from the classical U-RANS model in NRG-FSIFOAM. Secondly, new developments in the FSI-coupling are investigated to improve overall performance of the NRG-FSIFOAM solver. To this purpose a novel extrapolation is designed, manually accommodating for the variability in the interface-displacement in FSI problems. The previous bare rod and wire-wrapped cases suggest future potential, having retrieved results as efficient as the generally proposed extrapolation method. Additionally the Manifold Mapping is integrated in NRG-FSIFOAM as an alternative efficient FSI coupling method.

Acknowledgements

This report is the work of many individuals through much support of them during the course of my final year Graduation Thesis project. In summary, there are more reasons why completion of my Graduation project would have not been realised, compared to it being realised. Therefore, I'd like to express my honest gratitude to everybody that played a role by giving a vote of thanks.

In the first place, I want to thank Nuclear Research and Consultancy Group (NRG) for giving me access to the required tools and information to complete the task. It was very nice being in an international environment of professionals from different fields and learning how these come together in nuclear research as a whole. Secondly, I want to express my sincere gratitude towards my supervisor at the TU/e, prof. dr. ir. Barry Koren, giving me valuable lessons when it comes to doing research through years of his own experience. I'm very grateful for the input, as well technical as strategy-wise, and will put it to good use in my future projects. Also, I sincerely would like to thank my supervisors at NRG, dr. ir. Afaque Shams and dr. ir. Dante De Santis for their guidance every week in our many meetings. Setting out goals and pursuing them step by step together was a nice learning experience for me.

Next, I'd like to thank other colleagues that supported me in many ways. Thank you, dr. ir. Andrea de Santis, dr. ir. Edo Frederix and dr. ir. Blaz Mikuz, I sincerely enjoyed the many talks we had during the course of my stay in the office, both informational or pleasantry. Moreover, thanks everyone else from the team for keeping good spirit, especially during lunch time.

Finally, I express my very profound gratitude to my family for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. Thanks Anna Bosma for all the *blah* that were mentioned when finalizing the report.

Alkmaar, November 2018

Jeremy Chouchoulis

Contents

Chapter 1

Introduction

Nuclear energy provides the highest energy density out of all the other sources currently on earth. In other words, the effective energy extracted per unit of mass or volume is the most for nuclear fuel. To wrap your mind around this; a chunk of uranium having the size of an egg is sufficient to produce energy for a complete life time of a human being. Being that efficient, employing nuclear fuel is a very neat way to generate power. On the contrast, that same "egg" could release energy equivalent to the amount of energy produced by the atomic bomb dropped on Hiroshima. Having such enormous potential, nuclear material can be extremely hazardous [16]. Consequently, it is of major importance that nuclear energy sources and their derivatives are properly dealt with.

Qualified researchers from all over the world are being employed to ensure safe energy production, including mathematicians. Due to the drastic increase in computer power over the last decade, application of mathematics and physics has been translated into algorithms by teaming up with computer scientists, making it possible to provide computer simulation results capable of predicting behaviour of nuclear components, and thus providing substantial information for further development of newer reactor types.

Large amounts of fossil fuels have been used over the last decades, diminishing air quality and increasing temperatures all of the world. Engineers and scientists from all over the world annually pursuit cleaner and more powerful energy sources to reduce the burden of fossil fuels. Several new technologies, such as the ITER project building a fusion reactor in Cadarache (southern France), are currently undergoing a lot of development. In order to comply with the ever increasing demand for power in the mean time, nuclear energy will play a major role in the future, it is the only proven low-carbon power source to meet modern economy electricity requirements [3].

For nuclear applications to play an increasing role in supplying energy needs, depending heavily on water-cooled and moderated reactors is outdated. An increase of coolant flow, applying a different coolant liquid or even reconsidering different structural components are innovations being considered. Each adjustment causes different fluid and structural behaviour, in which Flow-Induced Vibrations (FIV) can become more cumbersome. Studying FIV, and even more general Fluid-Structure Interaction (FSI), is thus of major importance in nuclear engineering, in order to detect such a phenomenon before physical application. Several different structural components are submerged in exceptional fluids (highly pressurized water or liquid metals), creating a system working in

extreme conditions compared to daily life. Small mistakes hence can have grave consequences. Due to the severity of the matter, safe operation of nuclear reactors is a prime goal to secure.

Trying to achieve such goals through experimental testing is very expensive, if even possible. Moreover analytical theories are often limited to simplified settings, and thus resulting conclusions can not be used as valuable information for realistic nuclear reactors. In the current digital age, computational simulation therefore provides great opportunity to work towards safe nuclear reactor operation in a cost-friendly manner. In order for such simulation methods to be a serious influencer in nuclear industry, it is important that results can be generated within a certain time frame. Direct numerical simulations (DNS, solving the exact equations) of fluid behaviour can produce very accurate results, these simulations however can easily take up months of computational efforts. This is unfavourable, as there are reactors constructed, developed in a duration of ten years before construction started. Hence, if future ambitions include design of reactor components, or even full cores, efficient simulation is to key to victory.

1.1 Research objective

At Nuclear Research and Consultancy Group (NRG), an in-house Fluid-Structure interaction (FSI) software package, called NRG-FSIFOAM, has been developed based on other open-source software packages. In this way, state-of-the-art algorithms from all aspects of FSI can be implemented and combined in order to work towards the most efficient simulations to date. Not only must NRG-FSIFOAM be efficient, but the results produced must also be reliable. Therefore a first set task is to

verify NRG-FSIFOAM's coupling and pressure fluctuation model for its accuracy compared with different (commercial) software.

Moreover, as NRG-FSIFOAM is based on open-source codes, it is very flexible to develop new models and improve all aspects of FSI separately from one another. As FSI is a very broad subject, seeking improvement in all parts would be to big of a task for the scope of this project. Consequently focus is put in to

improving the efficiency of the fluid-structure interaction coupling method, by maintaining at the same time the stability and the accuracy of the method,

under the form of a second research objective. The strongly-coupled algorithms, such as IQN-ILS, that NRG-FSIFOAM utilizes, take up several iterations per time step, meaning that a big chunk of the total duration goes to coupling. Small improvements for a single time-step could thus results in detrimental improvements for a complete simulation.

1.2 Thesis outline

Chapter 2 starts of with an introduction of fluid-structure interaction in nuclear reactor history. Much detail is put in to the way flow-induced vibrations of nuclear components

can manifest. Thereafter, chapter 3 describes the governing mathematical equations of the problem at hand. Knowing this, chapter 4 follows with a summary of the numerical procedures needed to solve these mathematical models. Only the theory of coupling methods is shifted to chapter 5, as this will play a larger part in the research of the thesis. Additionally in this same chapter, improving theories are investigated. To complete the theoretical section of the reports, chapter 6 highlights efficient procedures to simulate fluid behaviour that are being put to use in this work. The first results are given in chapter 7, that compares vibrational characteristics of a bare and wire-wrapped rod obtained with NRG-FSIFOAM and STAR-CCM+. Chapter 8 then gives the results obtained for the second research objective. Conclusions are at last made of all results and in chapter 10 a list of topics is written down for future research.

Chapter 2

FSI in nuclear applications

Flow-induced vibrations (FIV), and even more general Fluid-Structure Interaction (FSI), are of major importance in the context of nuclear reactors. Many structural defects due to fretting wear, fatigue problems and stress corrosion cracking originate from FIV. The earliest reports of incidents concerning heat exchanger tubes appeared in the 1950s [46]. In order to perform well set up research, it is important to look in the past. The peculiar behaviour of specific components in events from the past provide a base of knowledge for scientific investigation. Ahead of describing such events, a general overview of the internal workings of nuclear reactors and their differences is given.

2.1 Nuclear reactors

Getting to understand basic concepts of nuclear energy via the use of reactors, makes one realize how far human kind has come. Reforming atoms of materials from within was beyond imaginable two or three human generations ago. This section shortly introduces the reader to the formation of nuclear history, what the future will hold and how nuclear energy came to existence.

2.1.1 Types of nuclear reactors

The origins of nuclear fission (and reactors) begin shortly before the onset of WWII. At that time, attention was heavily put in exploitation of nuclear materials for its destructive capabilities. Not long after, nuclear propulsion of submarines became primary application for the military, shifting focus towards production of compact reactors. In order to fit in submarines, nuclear engineers experimented with alternative coolants and moderators to reduce size. As a consequence, the earliest models of small light water reactors (LWRs), pressurized water reactors (PWRs), liquid metal-cooled fast reactors (LMFRs), lead-cooled fast reactors (LFRs) and sodium-cooled fast reactors (SFRs) came into existence, of which the LWRs were most common to be used.

Since then, over the last half century, nuclear reactor technology has developed a lot. This development arose in major design changes, and a fair amount of future design concepts already exist. For this reason, classification of nuclear reactors is done by generations (see figure (2.1)), Generation I reactors being early commercial prototypes

developed in the 1950s.

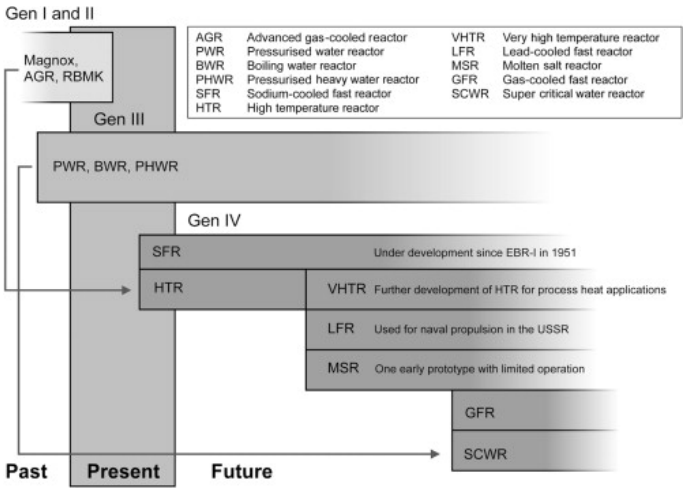


Figure 2.1: *Timeline for development of different fission reactor technologies [16].*

Most of the nuclear reactors being operational to date are Generation II reactors from 1960s onwards, giving the first commercial PWRs and BWRs. A short diagram is given in pie charts (2.2), representing the distribution of reactors and their net energy capacity by type.

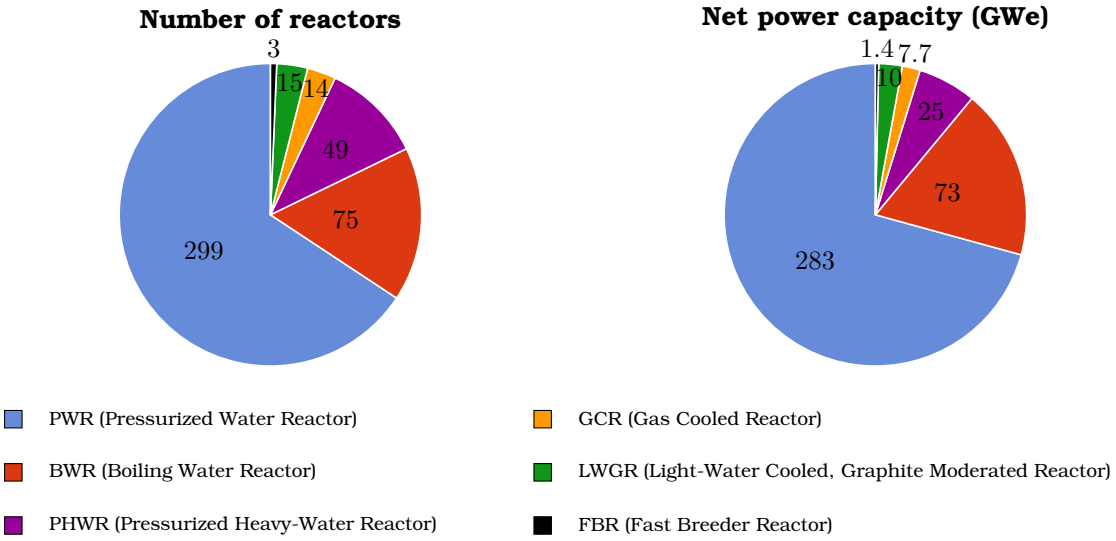


Figure 2.2: *Number of reactors and net capacity given by the IAEA [19], data updated at 1st September of 2018. GWe = capacity in thousands of megawatts (gross).*

From 1990 on, advanced versions of PWRs, BWRs and pressurized heavy water reactors (PHWRs) were commissioned, designated by Generation III. Generation IV reactors are meant to address many of the social, safety and economical concerns to further spread nuclear technology. Under these are six; very high temperature reactors (VHTRs), gas-cooled fast reactors (GFRs), molten salt reactors (MSRs), supercritical water-cooled reactors (SCWRs), the SFRs and the LFRs. Generally, these new concepts rely on running on way higher temperatures compared to the water-cooled reactors currently in use. The technology behind these concepts is however not yet ready for commercial usage and is

still undergoing a serious development program to provide smartly engineered materials and components capable of handling the extreme conditions. Aside prototype models, Generation IV reactors will not be operational before the mid-2020s [16, 1].

2.1.2 Workings of a pressurized water reactor

Many different kinds of reactors exist and giving a concise description of all of them is too vast for the purposes of this work. However, most different designs rely on basic concepts which will be explained below through the example of a pressurized water reactor (PWR). For this, the reader is suggested to follow figure (2.3), starting from the fuel elements on.

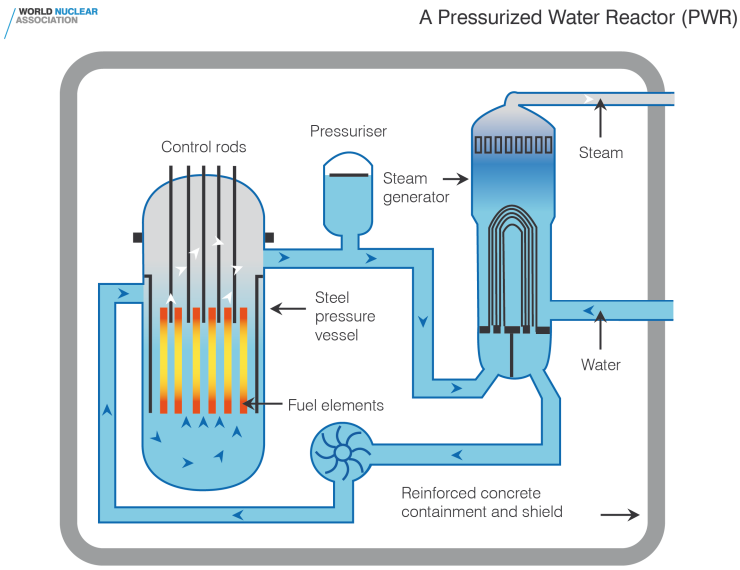


Figure 2.3: *A diagram of the basic working of a pressurized water reactor [1].*

Generating nuclear energy commences in the reactor core, i.e. in the nuclear fuel assembly containing fuel elements. The fuel is specifically selected to consist of heavy elements, such as uranium, making it energetically favourable to split nuclei. This process is more well known as 'nuclear fission'. When nuclei split, its nuclear fragments are in very excited states and emit neutrons, other subatomic particles, and photons. These neutrons on their turn get absorbed by other nuclear fragments, permitting other nuclear fission reactions to take place, allowing the reaction to become self-sustaining. In this 'chain-reaction', a very large amount of energy releases, and hence an enormous amount of heat gets built up rapidly. In an atomic bomb, the goal is to generate such an extremely fast reaction, producing the prompt, tremendously energetic explosions. Nuclear reactors are designed in such a way, to keep output on a constant level, by maintaining this chain-reaction ([29] p. 1). One way this is achieved, is by surrounding the heated fuel in coolant material, that often also acts as a speed-limiter for neutrons (moderator). This is preferred, as neutrons are absorbed more frequently when moving slower. In a PWR, light water (normal water) is utilized for these purposes. However, temperature gets up way too fast at normal pressure. Hence, a pressurizer (see figure (2.3)) increases pressure up to 150 times atmospheric pressure, allowing water to reach 325°C [1]. Having such high

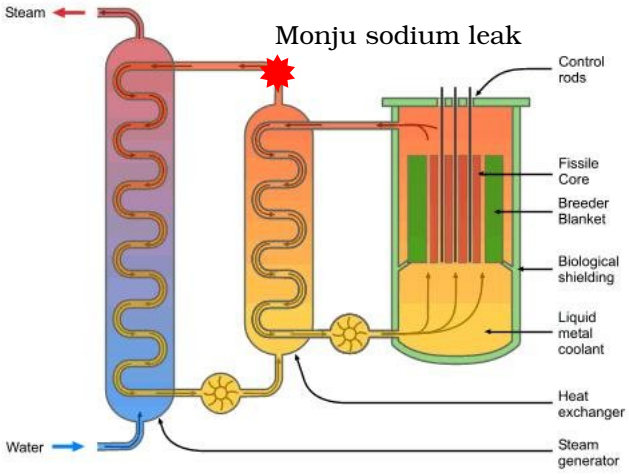


Figure 2.4: *A diagram showing the basic workings of a "loop" design fast breeder reactor [2]. Moreover, the position of the Monju sodium leak is shown [33].*

temperature, this water on its turn is used to generate steam. Via an inlet, water flows to the U-tubes which induces steam to be formed. To conclude, this steam, as in fossil fuel energy facilities, forces turbines to spin, which is often the last barrier to producing energy. Water in the U-tubes will then flow back towards the reactor core, which will heat it up again, after having transferred quite a lot of heat to build up steam. This completes the nuclear reactor cycle.

Differences between reactors constitute of having more than one cycle, different coolants, different reactor cores and steam generator designs, using different nuclear fuels and even more enhancements are being designed currently. For example, liquid-metal fast breeder reactors have been built using two loops and sodium, lead or a lead-bismuth alloy as primary coolant, illustrated in figure (2.4).

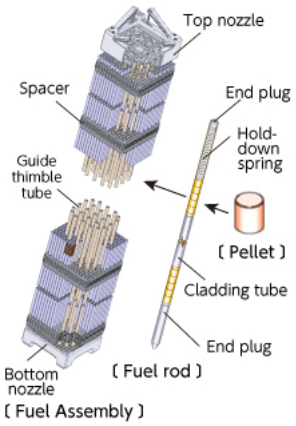


Figure 2.5: *A sketch of a fuel assembly and nuclear fuel pellets inserted in the core of a reactor [32].*

The use of metal coolants is favourable, since there are low vapour pressures, allowing to operate the reactor at nearly atmospheric pressures. Moreover, thermal conductivity is very high, allowing the core to cool down effectively through natural convection, and thus leaving out the necessity of having a moderator. As is often the case on earth, everything comes with a trade-off. Nuclear fuel assemblies (see figure 2.5)) consist of longer slender tubes. The tubes, called pins or rods, measure approximately 1 centimetre in diameter and roughly 3 to 4 metres in length as mentioned on page 4 in Britannicas introduction to nuclear reactors [29]. Liquid-metals having very high densities for fluids, are more likely to induce fretting wear, corrosion and a lot of flow-induced accidents as will be detailed below. Every reactor design has to be investigated extensively, as minor changes are easy to think of conceptually, but consequences are of very large extent.

2.2 Flow-Induced Vibrations

The many different structural components in a nuclear reactor, together with all the possible fluid flows around it give varying possibilities in the eventual behaviour of the system. In order to effectively perform any computational study, it is important to have a classification with which one can set up a model for the FSI. The term Flow-Induced Vibrations (FIV) is used for this purpose, characterizing all phenomena related to the FSI behaviour. Two major FIV events are described to support the classification.

2.2.1 Classification

In figure (2.3) it can be seen that different excitation mechanism prevail in different regions within the reactor. The fuel rods are mostly immersed in an axial flow, whereas in the steam generator both axial and cross flow play a role. A boiling water reactor (BWR) can have even different states, i.e. gas, liquid or multiphase. Pettigrew et al. [36] have given a classification (figure 2.6), based on four natural excitation mechanisms; (1) fluid-elastic instability, (2) periodic vortex shedding, (3) turbulence induced vibrations (TIV) and (4) acoustic resonance. In this diagram the relative importance of the above mentioned flows w.r.t. to the four excitation mechanisms is given.

Flow situation	Fluidelastic instability	Periodic shedding	Turbulence excitation	Acoustic resonance
Axial flow				
Internal				
Liquid	*	—	**	***
Gas	*	—	*	***
Two-phase	*	—	**	*
External				
Liquid	**	—	**	***
Gas	*	—	*	***
Two-phase	*	—	**	*
Cross flow				
Single cylinders				
Liquid	—	***	**	*
Gas	—	**	*	*
Two-Phase	—	*	**	—
Tube Bundle				
Liquid	***	**	**	*
Gas	***	*	*	***
Two-phase	***	*	**	—

***Most important.
**Should be considered.
*Less likely.
—, Does not apply.

Figure 2.6: *Vibration excitation mechanisms and their relative importance [36].*

A short description of the four vibration excitation mechanisms which can be present in nuclear reactors is given below:

1. Fluid-elastic instabilities arise directly form the fluid-induced dynamic forces onto the structure. If these forces transfer more energy than is damped out by the structure to restore its position, excessive vibration arises. The velocity of the fluid at which this occurs is called the critical velocity. Mostly for axial flow, this is not a problem, as the Elastic modulus (E) for tubes and fuel rod used in a nuclear reactor

are generally large enough to stay under critical velocities. Cross flow around tube bundles however are very much affected on the other hand.

2. Vortices are periodically shed downstream of structures in cross flow. Because of the shedding, periodic pressure variations are induced onto the structure. If the vortices are created at a similar frequency as the natural frequency of the structure, resonance may occur. This can be a problem, if the amplitude is large enough to keep the generation of vertices intact.

3. Turbulence can be a source of vibration in two different ways. Locally, the structure can be put into vibration as the flow moves around it, which is call near-field excitation. It's counterpart, far-field excitation, occurs if an upstream component, such as inlet nozzles, elbows and other piping elements produce turbulence. This turbulence generates random pressure fluctuations around the surface of components forcing them to vibrate. TIV is the principle source of vibration in axial flow.

4. Acoustic resonance can play a role both in axial and cross flow. In cross flow, acoustic resonance occurs mostly for gas flow around a tube bundle if the periodic vortex shedding frequency coincides with the natural frequency of the acoustic cavity formed by the structures surrounding the tube bundle. Intense acoustic noise is produced, which can lead to large vibration amplitudes and severe damage as a consequence.

Generally, a structural component is evaluated for its response towards all four mechanisms, which are usually different. Thereafter, the responses are superimposed to the predict the final behaviour in relation to the flow, which is demonstrated in figure (2.7). FIV usually has the greatest potential. However, different mechanism contribute to the onset of another one, as for example is explained with acoustic resonance.

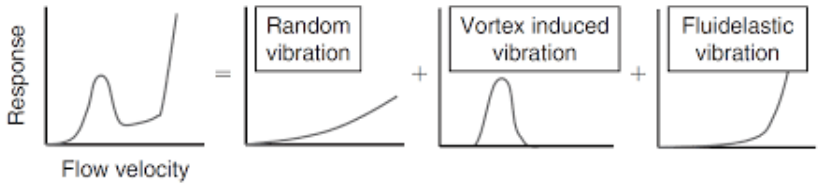


Figure 2.7: *Superimposition of response vibration to different FIV mechanisms [22].*

2.2.2 FIV accidents in the past

In the past, there have been plenty of occurrences of FSI vibration in nuclear reactor components resulting into an unscheduled shutdown due to mechanical damage. To get a better grasp of the previous concepts in the FIV classification, two specific accidents are shortly detailed for the application at hand.

Monju thermometer well — symmetric vortex shedding

In Japan at the Tsuruga Nuclear Power Plant, the Fast Breeder Reactor (FBR) *Monju* leaked sodium causing a major fire and forcing shutdown. A mechanical defect of a temperature sensor installed in the secondary loop near the outlet of the internal heat exchanger allowed several hundred kilograms of sodium to leak out onto the floor below the pipe (see figures (2.4) and (2.8)). Thermocouple wires are usually enclosed by a sheath,

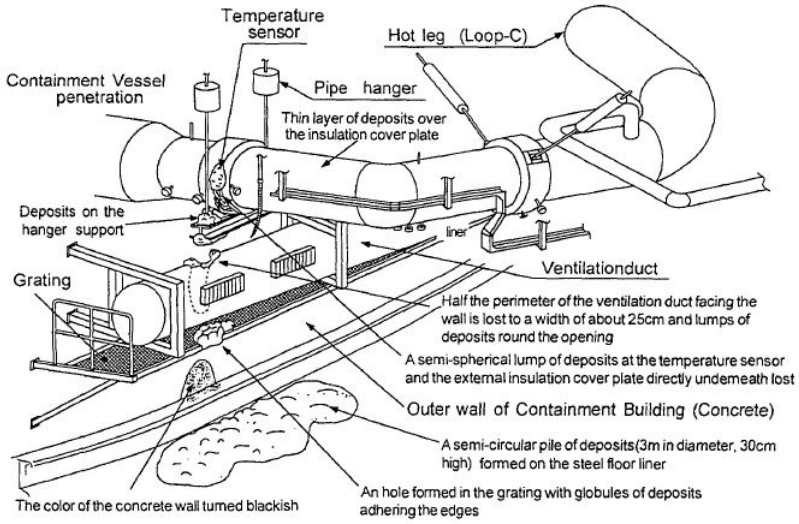


Figure 2.8: *Sketch of the post-accidental state of the Monju reactor displaying the leakage of sodium [33].*

which by itself is housed in a slim well tube, being approximately 15cm in length. Due to being horizontally inserted into the center of the pipe, the well tube is susceptible for flow-induced vibrations. A taskforce organized by the Science and Technology Agency (STA) took X-ray photographs of the incident location in order to estimate the extent of remaining sodium compounds around the temperature sensor and to assess structural condition. Results revealed the tip of the well was missing and the united thermocouple wires in the sheath were bent in the stream-wise direction, which is shown in figure (2.9).

In order to perform more detailed investigation, part of the pipe wall together with the temperature sensor was cut out and transported to a research institute. Microscopic and metallographic examinations together with flow-induced vibration analysis confirmed the direct cause of the failure was high cycle fatigue with crack initiation and slow propagation with ductile rupture in conclusion. Vibration was not propagated by Von Karman vortex shedding but by a symmetric vortex shedding.

Due to this accident, new wells for thermometers were designed to specifically compensate against flow induced vibration according to a revised design guide prepared by JNC (Japan Nuclear Cycle development Institute, currently known as the Japan Atomic Energy Agency (JAEA)). Main changes were made by decreasing length and increasing diameter size. Several different countermeasures in conjunction with the new design regulations of the thermometer were undertaken at the facility. [33, 42].

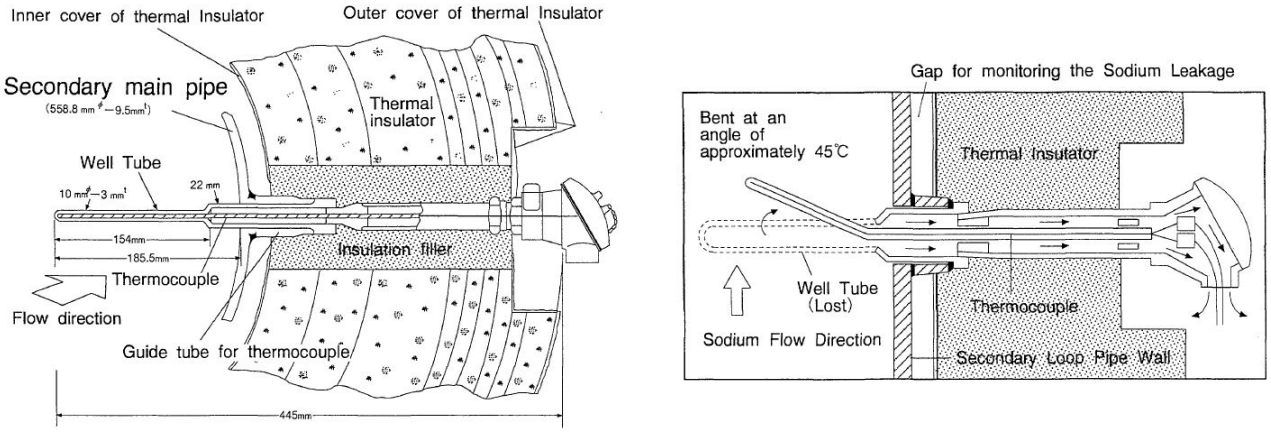


Figure 2.9: *On the left the thermocouple well of the secondary circuit is visualized before the Monju accident. On the right the path the sodium followed is shown together with the bending of the thermometer [33].*

Mihama steam generator tube failure – fluid-elastic vibration

Four years earlier, also in Japan, the *Mihama Nuclear Power Plant Unit 2* was automatically shut down due to leakage in one of its steam generators. In the upper U-bend part, it was found that one tube was ruptured at the No. 6 tube supporting plate, leaking 55 tons of primary cooling water into the secondary cooling loop [24, 42].

After an intensive investigation, it was concluded the source was the incorrect insertion (not deep enough) of Anti-Vibration Bars (AVBs) (see figures (2.10) and (2.11)), 20 years before the event. Due to this poor composition, fluid velocity was accelerated more than it should, leading to out of plane fluid-elastic vibration of the U-tube. Even more so, the critical velocity to excite the natural frequency was decreased due to lack of support by the AVBs, making it easier for the actual event to roll out as it happened.

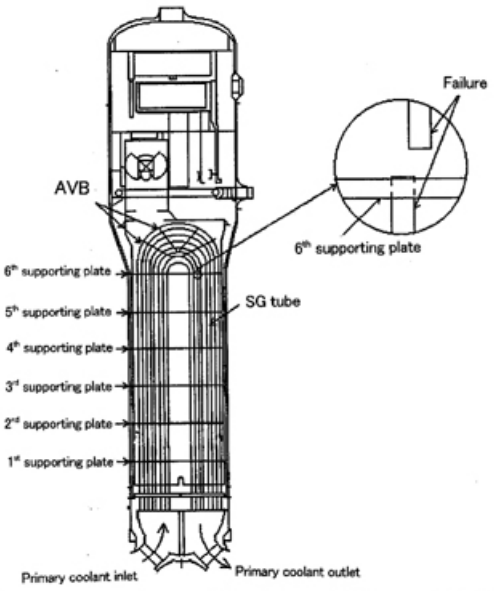


Figure 2.10: *Mihama steam generator and the location of failure of the steam generator tube [24].*

A full chronology of the event is given in [25], describing the operators handling manoeuvres step by step. Next to tube rupture of the steam generator, some pressurizer relief valves were unavailable and the main steam isolation valve did not manage to fully seal via the automatic closure system.

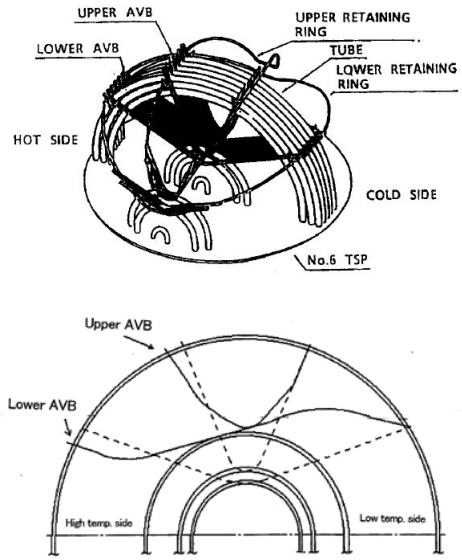


Figure 2.11: *The upper figure shows a concept of the steam generator Anti-Vibration Bars. The other figure displays the positioning of the AVBs during accident, in which the dotted lines represent normal positioning [24, 25].*

This steam generator tube rupture event was the first of its kind in Japan, gathering a lot of public attention, also from nuclear safety specialists. Consequently, the regulatory authority in Japan ordered a thorough investigation of the event and undertaking the appropriate measures in order to prevent similar accidents in the future. Among these, it was requested to fully examine fabrication records and eddy current testing records of AVBs in all steam generators in Japan. Next to record checks, several improvements and restrictions were made.

The above described events caused a long stoppage of reactor operation, putting a heavy financial burden on the nuclear companies through post-accidental investigations and not being able to sell the actual product of the nuclear reactor, whether it be energy or medical products. In the future, the goal is to use computational simulations of FSI components in nuclear reactors to prevent extensive financial cost and to win time in design and safety investigations, consequently, leading to a better nuclear industry for everyone.

Chapter 3

Governing equations in FSI

Knowing the importance of Fluid-Structure Interaction (FSI) in the field of nuclear applications, it is important to understand the mathematical behaviour of fluids and solids in order to computationally research particular aspects in nuclear industry. The world we are living in however is quite complex. Having non-Newtonian fluids (e.g. Oobleck) and very elastic materials (e.g. Graphene, used for flexible touch screens), multi-physics phenomena of many different materials can be described. Attention will therefore be focused on fluids and solids related to the nuclear industry, describing only the governing equations of interest for the scope of this work. Moreover, the mathematical equations to keep the solid and fluid in equilibrium are also given.

3.1 Frames of reference

Before the actual equations are formulated, it is important to understand from which point of view it is beneficial to observe the physical phenomena at hand. Despite the actual behaviour being independent of the *frame of reference*, different formulations of solid and fluid behaviour are more convenient to understand and consequently solve in certain situations. A simple example is a person walking in a train; dependent on the application, sometimes it is more convenient using the train as reference, but it could also be necessary to know the persons speed w.r.t. a waiting bystander to pass the rail. Typically, in continuum mechanics the Lagrangian and the Eulerian approach are considered. For FSI however, in order to solve fluid dynamics, often a method in between, called the Arbitrary Langrangian-Eulerian (ALE) is preferred. These three are explained below.

In the *Lagrangian description* physical properties of the considered body are described in terms of the material or referential coordinates $\boldsymbol{X}$ and time t. The reference configuration is the initial configuration of the material body at time $t = 0$s. To clarify, an observer on the moving body follows the change in properties of all domain points w.r.t. to its initial position. Computationally this is achieved by creating a grid that follows the body in its motion by having all grid points fixed as a specific material points. Therefore the Lagrangian description does not have to deal with convective terms. This is especially convenient for structural mechanics, as any displacement $\mathbf{u}$ of a specific point on the body is directly followed and compared with its initial state.

The extreme alternative is to observe the moving continuum from a fixed spatial location, independent of the moving material, this is called the *Eulerian description*. Sometimes it is more interesting to study physical properties of a material body at a specific pre-set point in space. Practically this means that in order to measure change of any physical property f over time on the (Eulerian) coordinate $\mathbf{x}$, both the change of the property at that specific coordinate and the change due to convection of the material must be added. This is expressed by the material derivative

$$\frac{\mathrm{d}f}{\mathrm{d}t} \coloneqq \left.\frac{\partial f}{\partial t}\right|_{\boldsymbol{X}} = \left.\frac{\partial f}{\partial t}\right|_{\mathbf{x}} + \mathbf{v}\cdot\nabla f\,. \tag{3.1}$$

Remark 1. *The functions f on the left hand side (depending on $\boldsymbol{X}$) and the right hand side (depending on $\mathbf{x}$) are two different functions. As both describe the same property, no difference is made in notation.*

One can simply imagine the convenience of the Eulerian frame by thinking of water in rough sea; it is more helpful understanding the macro-behaviour of the sea by looking at specific locations, compared to tracking particles of the sea water. For this reason, fluid dynamics are mostly solved in the Eulerian description.

The bottleneck in FSI is that a solid is pushing a fluid and vice-versa, meaning that the fluid domain is changing. In concrete, the general Eulerian description deals with many difficulties when having moving boundaries, and is thus not an option. Several solutions exist (see for example subsection 2.2 in [11]), each with their own advantages. The most well known however is the *Arbitrary Langrangian-Eulerian* (ALE) approach, that does move the frame of reference, however not at the velocity of the material itself, but an arbitrary velocity $\hat{\mathbf{v}}$. To this end, a third domain is needed as referential configuration, using coordinates $\boldsymbol{\mathcal{X}}$ (figure 3.1). A similar formula as the definition for the material derivative (3.1) can be derived (see Donea et al. [14]) in ALE formulation. This is given as

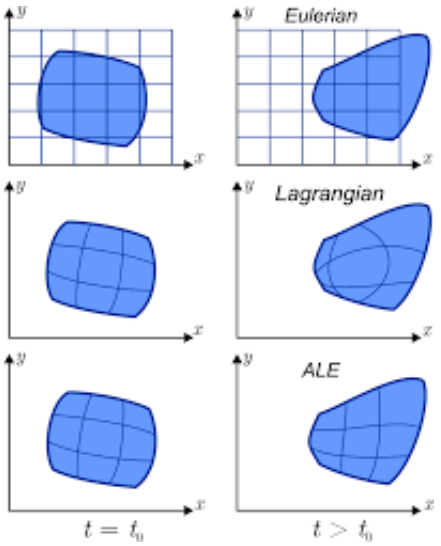


Figure 3.1: *Short illustration of the working of Arbitrary Lagrangian-Eulerian formulation on a fluid domain [7].*

$$\left.\frac{\partial f}{\partial t}\right|_{\boldsymbol{X}} = \left.\frac{\partial f}{\partial t}\right|_{\boldsymbol{\mathcal{X}}} + \mathbf{c}\cdot\nabla f\,, \tag{3.2}$$

in which $\mathbf{c} \coloneqq \mathbf{v} - \hat{\mathbf{v}}$ is the relative velocity between the material and the mesh, and ∇f is the gradient in Eulerian coordinates. Intuitively this means that a portion of convection is already taken into account by the moving frame of reference. Hence this is compensated by subtracting the mesh velocity $\hat{\mathbf{v}}$ in the convection term.

Specifically in FSI, it is chosen to move the interface according to the Lagrangian

description, which avoids overlap between the fluid and solid domains. The interior of the fluid domain is then extended in accordance with the interface using appropriate techniques to avoid excessive distortion in the fluid domain.

3.2 Governing solid equations

Structure deformation is given by the *Cauchy momentum equation* in Lagrangian form by

$$\rho \frac{\mathrm{d}^2 \mathbf{u}}{\mathrm{d}t^2} - \nabla \cdot \boldsymbol{\sigma}_s = \rho \mathbf{g}_s \,, \tag{3.3}$$

$\mathbf{u}$ being the displacement, $\boldsymbol{\sigma}_s$ the Cauchy stress tensor and $\mathbf{g}_s$ the body forces per unit volume on the structure.

For our purposes, linear elastic behaviour is considered. This means stress $\boldsymbol{\sigma}_s$ and strain $\boldsymbol{\epsilon}_s$ are linearly dependent according to Hooke's constitutive law

$$\boldsymbol{\sigma}_s = \mathsf{C} : \boldsymbol{\epsilon}_s \,, \tag{3.4}$$

C being the fourth-order stiffness tensor. To complete the system of differential equations, a strain-displacement relation must be given. This is given by the *Green-Lagrange* strain tensor

$$\boldsymbol{\epsilon}_s = \frac{1}{2}\left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T + (\nabla \mathbf{u})^T \nabla \mathbf{u}\right] \,, \tag{3.5}$$

valid for large displacements. This way, non-linearity is introduced to the system, which makes things more difficult. Hence, eq. (3.5) is often linearised, giving

$$\boldsymbol{\epsilon}_s = \frac{1}{2}\left[\nabla \mathbf{u} + (\nabla \mathbf{u})^T\right] \,, \tag{3.6}$$

valid for small (linear) displacements. Making a simplification such as this is justified if small time steps are used when solving for the solid behaviour. For larger time steps there is a loss in accuracy.

3.3 Governing fluid equations

The flow of Newtonian fluids is considered, described by the incompressible Navier-Stokes equations. These are given by the continuity equation and the momentum equation

$$\begin{cases} \nabla \cdot \mathbf{v} = 0 \,, & \text{(3.7a)} \\ \left.\dfrac{\partial \mathbf{v}}{\partial t}\right|_{\mathbf{x}} + (\mathbf{v} \cdot \nabla)\,\mathbf{v} - \nu \nabla^2 \mathbf{v} = -\dfrac{1}{\rho_0}\nabla p + \mathbf{g}_f \,, & \text{(3.7b)} \end{cases}$$

on $\Omega_f(t)$, with $\mathbf{v}$ the velocity field, p the pressure, $\mathbf{g}_f$ the body forces per unit volume acting on the fluid and $\nu = \mu/\rho_0$ the kinematic viscosity (μ the dynamic viscosity) of the considered fluid. Incompressibility is assumed, as mostly water or liquid-metal flow is considered, thus having Mach numbers (the ratio of the speed of the flow to the speed of

sound) lower than 0.3. Hence, the density is a constant $\rho_0 \coloneqq \rho$. The above equations are in the Eulerian description. Using eq. (3.2), one can write the Navier-Stokes equations in ALE formulation as follows

$$\begin{cases} \nabla \cdot \mathbf{v} = 0\,, & \text{(3.8a)} \\ \left.\dfrac{\partial \mathbf{v}}{\partial t}\right|_{\boldsymbol{\mathcal{X}}} + (\mathbf{c} \cdot \nabla)\,\mathbf{v} - \nu \nabla^2 \mathbf{v} = -\dfrac{1}{\rho_0}\nabla p + \mathbf{g}_f\,. & \text{(3.8b)} \end{cases}$$

Newtonian fluids are defined by having viscous stresses linearly dependent on local strain rates at every point. The expression of the deviatoric (viscous) stress for isotropic Newtonian fluids is given by *Stokes' stress constitutive equation*,

$$\boldsymbol{\tau} = \mu \left(\nabla \mathbf{v} + \nabla \mathbf{v}^T\right)\,. \tag{3.9}$$

In total, the *Cauchy stress tensor* is obtained from the sum of pressure terms and viscous terms,

$$\boldsymbol{\sigma}_f = -p\mathbf{I} + \mu\boldsymbol{\tau}\,, \tag{3.10}$$

in which $\mathbf{I}$ denotes the unit tensor.

3.4 Coupling between solid and fluid

Having two different physical continua to model, it is important to gear one to another correctly. In FSI, a fluid exerts additional forces on a solid and simultaneously, solid movement forces the fluid on a different track. This behaviour is communicated to each other through the boundary conditions on the fluid-structure interface $\Gamma_i(t)$. As highlighted before, these boundary conditions determine well-posedness for both the fluid as the solid problems on their own. Setting realistic coupling conditions hence is of much importance for keeping FSI in equilibrium. A no-slip fluid-structure interface adheres to the equality of velocity (kinematic condition)

$$\mathbf{v} = \frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t}\,, \tag{3.11}$$

and the equality of traction (dynamic condition)

$$\boldsymbol{\sigma}_f \cdot \mathbf{n}_f = -\boldsymbol{\sigma}_s \cdot \mathbf{n}_s\,, \tag{3.12}$$

where $\mathbf{n}_f$ and $\mathbf{n}_s$ are unit normal vectors pointing outwards from $\Omega_f(t)$ and $\Omega_s(t)$. Because of the interaction, both domains will reform. Boundary conditions on $\Gamma_i(t)$ will have to be reset each time step, which will be done by calculating the data needed for the next time step, and transferring it from fluid to solid or vice-versa.

3.5 Boundary conditions

Dynamical equations are always accompanied with proper boundary conditions for the particular domain in consideration to have a well-posed problem. Next to the coupling interface, the other boundaries also have to be imposed with specific variable conditions.

In structure mechanics, a fixed, pinned or free boundary condition is mostly assumed on $\Gamma_s(t)$, and not taking them all free. This assures that the structure does not undergo rigid body translations and rotations. A basic illustration of the boundary conditions for a beam is given in figure 3.2.

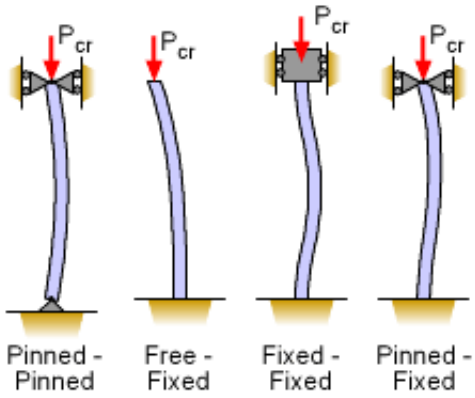


Figure 3.2: *Different types of boundary conditions frequently being used in solid mechanics [17].*

Imposing conditions on $\Gamma_f(t)$, the boundary of the fluid domain, is usually done by having a source velocity profile together with different conditions for the pressure to create a realistic situation. A reasoning for this is given through the different types of partial differential equations. For simplistic problems, such as the *parabolic heat equation* $\partial_t \mathbf{v} - \nabla^2 \mathbf{v} = \mathbf{g}$ or the *hyperbolic* (1st order) *transport equation* $\partial_t \mathbf{v} - \boldsymbol{a} \cdot \nabla \mathbf{v} = \mathbf{g}$, choosing correct boundary conditions is well known due their classification. More involved partial differential equations such as the Navier-Stokes equations (3.8a) and (3.8b) however do not adhere to such classification strictly, but should be seen as leaning towards parabolic or hyperbolic behaviour. To illustrate:

- large ν describes a very viscous fluid. An example is a stirred eddy that dissipates quickly. One can recognize the character of the *heat equation* dominantly over the transport behaviour. In this case all boundaries are constraint by a condition to form a well-posed problem;

- small ν describes a flow with much transport compared to viscous dissipation, like flow of water in a tube. Here, the character of the *transport equation* is dominantly present. A pure hyperbolic problem does not need a boundary condition at the outflow of a generic fluid domain.

As viscosity and thus turbulence will always play a role, even when ν being small, boundary conditions are imposed at all parts of the fluid domain. More specifically, at the

outflow, $\partial_{\mathbf{n}}\mathbf{v} = 0$ ($\mathbf{n}$ the unit normal at the outflow of $\Omega_f(t)$) and $p = 0$ are chosen for all considered simulations. Usually, software for fluid solvers request more boundary conditions for different variables not mentioned here. This for example occurs by means of the U-RANS method for solving fluid equations that describes new variables and additional equations, explained in chapter 6.

Chapter 4

Numerical procedures

There are several different manners with which one can solve Partial Differential Equations (PDEs) such as given in sections 3.2 and 3.3. Usually, three different methods are distinguished, the Finite Difference Method (FDM), the Finite Volume Method (FVM) and the Finite Element Method (FEM). All of them are valid approaches, however the different choices are made as a specific method can be deemed as more convenient for a particular PDE. Multiple aspects however come up while solving FSI problems that are not present solving either fluid or solid problems apart from each other. Under these are the mesh movement of the fluid domain, as explained in the previous section, and mapping of data between solid and fluid domains. Once a specific choice has been made in all aspects, a coupling method has to be identified to solve the FSI problem. These approaches are described in this chapter except for coupling approaches which are given more into depth in the next chapter.

4.1 Finite element method

The structural equations (3.3), (3.4) and (3.6) are preferred to be solved by a Finite Element Method (FEM), due to structural equations usually having a specific symmetry. This is symmetry is denoted by the PDE operator being *self-adjoint*. Consider the vector space U, and the linear operator $L\colon U \to U$ (L represents the PDE as operator). Then L is self-adjoint if and only if

$$\int_\Omega uLv \, \mathrm{d}\Omega = \int_\Omega vLu \, \mathrm{d}\Omega, \qquad \forall u, v \in U\,. \tag{4.1}$$

This symmetry condition is very convenient as FEMs use the same basis functions for solving for the solution as for the complete PDEs (by means of a test function).

Galerkin's method is put to use in order to derive a weak formulation of the eqs. (3.3), (3.4) and (3.6). According to a chosen discretization of the solid domain, the displacement $\mathbf{u}$ is written as a sum $\mathbf{u}(\mathbf{x}, t) = \sum_{i=1}^{n} \mathbf{u}^i(t)\boldsymbol{\phi}^i(\mathbf{x})$ of specifically chosen basis function vectors $\{\boldsymbol{\phi}^i |\, i = 1, \ldots, n\}$ for the solution vector space U on the domain. The basis elements are chosen in such a manner that they "belong" to nodes/elements of the grid, resulting in the $\boldsymbol{\phi}^i$ being equal to zero in the remainder of the grid. Thus if $\mathbf{x}_i$ is a node of the grid,

then for each coordinate direction j, there exist a basis function vector, such that for a general grid node $\mathbf{x}_g$

$$\left(\boldsymbol{\phi}^i\right)_j(\mathbf{x}_g) = \begin{cases} 1, & \text{if } \mathbf{x}_g = \mathbf{x}_i \\ 0 & \text{if } \mathbf{x}_g \neq \mathbf{x}_i \end{cases}. \tag{4.2}$$

Linear basis functions were used during the course of this work. This results in the sparse semi-discrete system

$$M\boldsymbol{u}''(t) + K\boldsymbol{u}(t) = \boldsymbol{f}(t)\,, \tag{4.3}$$

in which M is the global mass matrix, K is the global stiffness matrix, $f(t)$ is the global load vector and $\boldsymbol{u} \coloneqq \left[\mathbf{u}^1(t), \ldots, \mathbf{u}^n(t)\right]$. In order to solve this equation, it is rewritten in two first order equations and then further discretized by means of the theta-method.

The C++ program library DEAL-II (Bangerth et al. [4]) is integrated in NRG-FSIFOAM for solving structural dynamics, which is targeted at the computational solution of partial differential equations using adaptive finite elements.

4.2 Finite volume method

Fluid equations, such as the Navier-Stokes equations (3.7a) and (3.7b) are not self-adjoint because of the convection term $(\mathbf{v}\cdot\nabla)\,\mathbf{v}$. This would mean that a FEM would impose a symmetry on the discrete version that is not actually there in the continuous PDEs. The Navier-Stokes equations do however also satisfy a convenient property, namely it is a composition of conservation laws (conservation of mass and momentum). The FVM involves the division of the computational domain into volumes and the imposition of conservation laws on these volumes. Afterwards, Gauß' theorem $\int_\Omega \nabla\cdot\boldsymbol{\phi}\,\mathrm{d}\Omega = \int_\Gamma \boldsymbol{\phi}\cdot\mathbf{n}\,\mathrm{d}\Gamma$ is applied to convert the problem to boundary integrals, $\boldsymbol{\phi}$ being the fluxes, which are evaluated at the boundaries. CFD calculations are therefore regarded more convenient to be solved by the FVM.

A more into depth discussion of the numerical approaches that are used in discretizing the convection term, diffusion term, source term, the reader is referred to take a look in Sharma's [41] and ter Hofstede's [43] Master theses reports. Both of these works discuss research and development of the in-house software package NRG-FSIFOAM, meaning that the discretization techniques explained are in-line with the methods in this project. The code used for this thesis project is a modified version of the OpenFOAM-Extend project (foam-extend 3.2, [34])

4.3 Mesh deformation

Fluid-Structure Interaction systems as an entity are given by the equations in sections 3.2 to 3.4. Multiple aspects however come up while solving FSI problems that are not present solving either fluid or solid problems apart from each other. Under such, as explained in the previous chapter, belongs the fluid mesh movement.

A brute-force approach is to completely regenerate the fluid mesh after each time step. This comes with major drawbacks as expected from brute-forcing. The Navier-Stokes equations are conservation laws, and these are not preserved with a complete regeneration. This is also not convenient for industrial applications, often having very intricate meshes, which would lead to expensive procedures ran by user-interaction procedures. Hence, preference goes to having an efficient and automatic method that relocates the mesh nodes, as the mesh must be updated frequently.

Node relocation is possible in the Arbitrary Lagrangian-Eulerian (ALE) formulation of the flow equations (section 3.1). The idea is mixing both approaches, leading to a velocity of the grid $\mathbf{w}_{\Omega_f}$ not equal to the fluid itself, but rather arbitrary. Only at the fluid-structure interface, velocity is strictly set because of boundary condition (3.11), which avoids overlap with the solid domain. The grid velocity at the interface is then spread out (function S) over the complete fluid domain,

$$\mathbf{w}_{\Omega_f} = S(\mathbf{w}_{\Gamma_i})\,. \tag{4.4}$$

Several different strategies exist to achieve this, having two basic underlying concepts.

A first group of techniques requests knowledge of the connectivity between grid points. Having this information, the grid itself is hypothetically considered as a physical body, implying that several known methods for movement can be applied. For example, the edges between nodes can be seen as linear springs, for which forces are modelled by Hooke's law. Moreover, one can also consider torsional springs, or a mixed model of them. Next to this spring analogy it is also possible to look at grid as a linear elastic body, the same way the structure is solved. Moreover, there is also the possibility to model grid velocity by diffusion, according the Laplace equation $\nabla \cdot (D\nabla \mathbf{w}) = 0$, D being the diffusivity. Biharmonic operators (fourth order derivatives) are also considered, as it has the characteristics of Laplacian smoothing, but offers grid cell control at the boundaries.

The second group does not take advantage of this property and solely relies on the position of the nodes. Interpolation by means of Radial Basis Functions (RBFs) is a well-established manner to perform mesh movement, described by de Boer et al. in [10], which are being put to use in this work. Although the use of RBFs for mesh movement on the one hand and mapping data between fluid meshes and solid meshes on the other hand is different, the basics are described in the next chapter.

4.4 Mapping methods

Due to having several different grids in FSI, often one for a fluid and one for a structure, complications arise letting both data from the fluid and structure interact. Considering partitioned solvers (explained in next section), different size requirements for the fluid solution and solid solution lead to grids being non-conformal or even non-matching with gaps and overlapping (figure 4.1). Also having a moving solid, the fluid grid needs to be deformed correctly, as mentioned in the previous section. Moreover, to speed up FSI calculations, surrogate model algorithms have been developed, taking into account

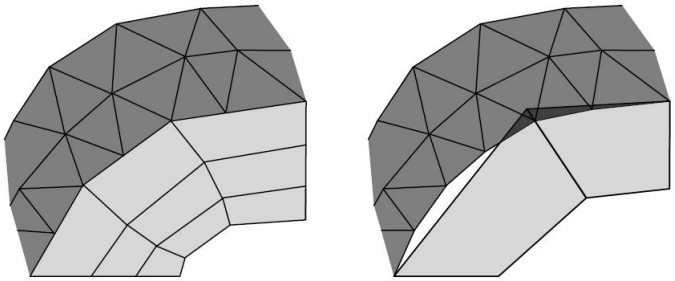


Figure 4.1: *Different types of boundary conditions frequently being used in solid mechanics.*

several levels of grids for the fluid and the solid and thus complicating the situation even more. In all of the mentioned cases, using appropriate mapping methods is essential. For very basic cases, one could rely on the nearest-neighbour "interpolation", taking over the data values from the nearest point on the other domain. In order to obtain more accuracy, interpolation can not be avoided however, resulting in more computations. The nearest-projection for example projects the nearest point on the requested grid, and interpolates the value at the projected point. Generally, Radial Basis Functions (RBFs) are put to use for mappings as they have been proven to be accurate and robust, and hence they are also mostly opted for when performing mapping operations.

Let's consider interpolation of fluid tractions $\boldsymbol{\sigma}_f$ from the fluid domain interface $\Gamma_{i,f}$ to the solid domain interface $\Gamma_{i,s}$, or the other way around the displacement $\mathbf{u}$ from $\Gamma_{i,s}$ to $\Gamma_{i,f}$. For generality, the parameter value is denoted by $\mathbf{z}$. The basic principle behind RBF interpolation is to construct an interpolant s (global or local) on the nodes $\mathbf{x}_{f,k} \in \Gamma_{i,f}$ ($k = 1, \ldots, n_f$), that is being evaluated at the nodes $\mathbf{x}_{s,l} \in \Gamma_{i,s}$ ($l = 1, \ldots, n_s$), based on radially symmetric basis functions $\phi\colon \mathbb{R} \to \mathbb{R}$

$$\phi\colon \|\mathbf{x}\| \mapsto \phi(\|\mathbf{x}\|), \qquad \mathbf{x} \in \mathbb{R}^3\,, \tag{4.5}$$

$\|.\|$ the Euclidean norm. Many different RBFs are being used, local or global, having different properties and consequently lead to different results [39]. It has been shown that the global Thin Plate Spline $\phi(\|\mathbf{x}\|) = \|\mathbf{x}\|^2 \log(\|\mathbf{x}\|)$ and the C^2 continuous basis function with compact support $\phi(\|\mathbf{x}\|) = (1 - \|\mathbf{x}\|)_+^4(4\|\mathbf{x}\| + 1)$ (the plus denotes the first term is zero if $1 - \|\mathbf{x}\| < 0$) perform very well for mesh movement for example. The number of points on the fluid-structure interface is often still very large. Instead of a single interpolant, and thus a large linear system, local RBFs with compact support are thus opted for. Multiple smaller linear system then have to be solved, of which the derivation is given below. The interpolant is then given as $s\colon \mathbb{R}^3 \to \mathbb{R}$

$$s(\mathbf{x}) \coloneqq \sum_{k=1}^{n_f} \alpha_k \phi_r(\|\mathbf{x} - \mathbf{x}_{\mathbf{f},\mathbf{k}}\|) + p(\mathbf{x})\,, \tag{4.6}$$

$\phi_r(\|\mathbf{x}\|) \coloneqq \phi(\|\mathbf{x}/r\|)$ and the radius is chosen as the maximum distance to any of the points $\mathbf{x}_{s,l}$, i.e. $r = \max\limits_{l=1,\ldots,n_s} \|\mathbf{x}_{s,l} - \mathbf{x}_{f,k}\|$. The function p is a global linear function

$p(\mathbf{x}) = \beta_0 + \beta_1 \mathrm{x}_1 + \beta_2 \mathrm{x}_2 + \beta_3 \mathrm{x}_3$, where $\mathbf{x} = [\mathrm{x}_1 \;\; \mathrm{x}_2 \;\; \mathrm{x}_3]^T$. Including p determines a unique interpolant, but request more equations to be solved as the values β_i ($i = 0, 1, 2, 3$) have to calculated. The main principle of interpolation is to build s in such a way to exactly satisfy the values $\mathbf{z}_{f,k}$ in the nodes $\mathbf{x}_{f,k} \in \Gamma_{i,f}$ ($k = 1, \ldots, n_f$) around which it is constructed, so

$$s(\mathbf{x}_{f,k}) = \mathbf{z}_{f,k}, \qquad k = 1, \ldots, n_f \,. \tag{4.7}$$

However, as mentioned, these equations on their own give an under-determined system. Regularity of the system is obtained by forcing that

$$\sum_{k=1}^{n_f} \alpha_k q(\mathbf{x}_{f,k}) = 0 \,, \tag{4.8}$$

for every polynomial $q \colon \mathbb{R}^3 \to \mathbb{R}$ of degree lower or equal to p. By this method, rigid body translations will captured exactly if $\mathbf{z}$ represent displacement for example. After solving both eqs. (4.7) and (4.8), for the values α_k ($k = 1, \ldots, n_f$) and β_i ($i = 0, 1, 2, 3$), the values $\mathbf{z}_{s,l}$ ($l = 1, \ldots, n_s$) are obtained through

$$s(\mathbf{x}_{s,l}) = \sum_{k=1}^{n_f} \alpha_k \phi_r(\|\mathbf{x}_{s,l} - \mathbf{x}_{\mathbf{f},\mathbf{k}}\|) + p(\mathbf{x}) \,. \tag{4.9}$$

All of the above evaluations can be more efficiently reformulated through matrix notation. Benjamin Uekermann introduces this together with parallelization of the RBF-interpolation in his dissertation [45], which is implemented in the preCICE library [8]. One can take advantage of the notation by not having to recalculate particular matrices.

Chapter 5

Coupling algorithms for FSI

Having the mathematical approaches at hand, it is important to consider how to solve both fluid equations and solid equations together, i.e. how to perform the coupling. Two different approaches exist. The *monolithic* approach writes all fluid and solid equations together as one bigger system of PDEs. The FSI problem is thus considered as one system, and any interacting variables are immediately taken into consideration. The remaining approach, solving the fluid and equations separately, is called the *partitioned* approach. These methods rely on a coupling scheme that maps the correct data at the interface between the solvers. A lot of computational efforts can be spend on coupling, enforcing well aligned fluid and solid solutions. The monolithic approach has the advantage that any coupling procedures are immediately taken into account in the system. However, it request a new code implementation for each case. Moreover, as explained above, different numerical schemes are preferable for different physics. A partitioned solver can handle this, as highly optimized codes from the last decades can be used on both the fluid and solid problem. This makes the partitioned approach modular and and thus the resulting software is easy to maintain and reuse. As the partitioned approach is used for all the simulations in this thesis, a mathematical basis is described below.

An extensive review article has been written on different partitioned simulation techniques by Degroote [11]. Most of the information described in this chapter is taken over from this paper, this includes basic algorithms and the IQN-ILS algorithm. Afterwards, two aspects of the IQN-ILS algorithm are inspected to investigate acceleration of unsteady FSI simulations. This is detailed through a surrogate-based method (Manifold Mapping), described by Blom et al. [6] in an FSI context and investigation of varying extrapolation methods. Al of the investigated coupling methods are integrated in NRG-FSIFOAM through preCICE (Precice Code Interaction Coupling Environment, [8]), which is a highly flexible open source coupling library for multi-physics simulations, that combines the structure solver library DEAL-II [4] and fluid solver library OpenFOAM [34].

5.1 Explicit versus implicit coupling

First, fluid and solid equations are discretized in space and time with the preferred methods (FVM with OpenFOAM for the fluid and FEM with DEAL-II for the solid). The following

coupled (fully) discrete equations are obtained

$$\begin{cases} \mathbb{F}(\boldsymbol{v}, \boldsymbol{d}(\boldsymbol{u})) = & \mathbf{0}\,, \\ \mathbb{S}(\boldsymbol{u}, \boldsymbol{s}(\boldsymbol{v})) = & \mathbf{0}\,, \end{cases} \tag{5.1}$$

in which $\boldsymbol{v}$ (discrete, thus different from $\mathbf{v}$) groups the flow variables velocity, pressure and others in Ω_f. The structure variables displacement, stress, etc. are gathered in $\boldsymbol{u}$ ((discrete, thus different from $\mathbf{u}$) w.r.t. Ω_s. On the fluid-structure interface Γ_i, the displacement w.r.t the initial geometry is given by $\boldsymbol{d}$ and the traction on the interface by $\boldsymbol{s}$. A Dirichlet-Neumann (DN) decomposition of the coupled problem is assumed, being the most common applied decomposition in Fluid-Structure Interaction (FSI) problems. The DN decomposition solves the flow equations for a specific velocity (displacement $\boldsymbol{d}$) of the fluid structure interface (Dirichlet boundary condition), in which the displacement is derived as a function of the solid parameters $\boldsymbol{d} = \boldsymbol{d}(\boldsymbol{u})$. The structure equations are then solved for a given traction distribution $\boldsymbol{s}$ on the interface (Neumann boundary condition), which is a function of the flow degrees of freedom, $\boldsymbol{s} = \boldsymbol{s}(\boldsymbol{v})$. All variables are at the new time level t^{n+1}. For clarity, constant vectors such as the solution vectors determined on previous time levels (t^n, t^{n-1},...) are hidden.

FSI problems are mainly focused on the interface as explained by the DN decomposition. The flow solver for example calculates $\boldsymbol{v}$ satisfying $\mathbb{F}(\boldsymbol{v}, \boldsymbol{d}(\boldsymbol{u})) = \mathbf{0}$ for a given *interface*-displacement $\boldsymbol{d}$ and subsequently extracts the *interface*-traction $\boldsymbol{s}$ from the newly calculated $\boldsymbol{v}$. The same holds for the structure solver. Hence it is convenient to write the solvers, calculating for variables satisfying the equations in (5.1), in an input-output fashion, leading for the flow solver to

$$\boldsymbol{s} = \boldsymbol{\mathcal{F}}(\boldsymbol{d})\,, \tag{5.2}$$

and for the structural solver to

$$\boldsymbol{d} = \boldsymbol{\mathcal{S}}(\boldsymbol{s})\,. \tag{5.3}$$

Due to the above notation, it is possible to look at FSI problems in a modular way; one has to solve fluid behaviour (5.2), solid movement (5.3) and gear those to one another via some coupling method. This separative manner of working facilitates an easier implementation in computer applications for general FSI problems. In order to achieve this, general coupling algorithms have to be derived, applicable for various fluid solvers $\boldsymbol{\mathcal{F}}$ and solid solvers $\boldsymbol{\mathcal{S}}$. As a consequence, the fluid solver $\boldsymbol{\mathcal{F}}$ and the solid solver $\boldsymbol{\mathcal{S}}$ are often seen as black-box solvers, limited to the fluid-structure interface. Next, these black-box solvers have to work with one another via some coupling method. Several methods exist, and it is reliant of the interaction strength in FSI which one is more appropriate. Therefore, a further division is made between *loosely-coupled* systems and *strongly-coupled* systems.

5.1.1 Explicit coupling

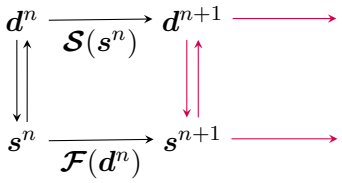
FSI problems are called loosely-coupled (or weakly-coupled) when the interaction is dominated by one participant, either the fluid or the solver. To clarify, imagine a steel lamppost

standing in a cool breeze. The natural frequency of this lamppost is hardly affected by the low forces of the wind. This is due the density of the solid ρ_s being much higher than the density of the fluid ρ_f. One speaks of a small 'added-mass', i.e., the mass of fluid which is accelerated by the structure.

Because of the domination from one side, it is sufficient to map data only once at each time-step. In other words, the solvers calculate new information solely based on the values at the previous time-step without iterations. Hence these methods are being called *explicit methods*.

ALGORITHM 5.1: Parallel explicit coupling

Solve $\boldsymbol{s}^{n+1} = \boldsymbol{\mathcal{F}}(\boldsymbol{d}^n)$
Solve $\boldsymbol{d}^{n+1} = \boldsymbol{\mathcal{S}}(\boldsymbol{s}^n)$
Advance to time-step $n+1$



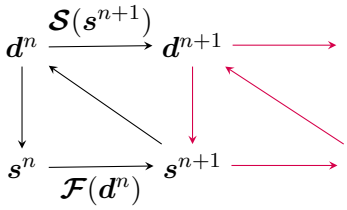
Two main strategies exist in this aspect. The first scheme solves both equations (5.2) and (5.3) parallel and at the end of each step exchanges data between the solvers, see algorithm 5.1.

Remark 2. *In all the diagrams next to the algorithms, each node represents the data calculated up until a certain time-step n by the solver $\boldsymbol{\mathcal{S}}$ or $\boldsymbol{\mathcal{F}}$, using the values send by the arrow. All the nodes thus represent the status of the structure (above) and the fluid (below).*

Secondly it is possible to perform a serial scheme, solving first for the dominant physics of the FSI system, then mapping data to the second solver to solve this afterwards. A basic algorithm is given in algorithm 5.2.

ALGORITHM 5.2: Serial explicit coupling

Solve $\boldsymbol{s}^{n+1} = \boldsymbol{\mathcal{F}}(\boldsymbol{d}^n)$
Solve $\boldsymbol{d}^{n+1} = \boldsymbol{\mathcal{S}}(\boldsymbol{s}^{n+1})$
Advance to time-step $n+1$



Such explicit methods are easy to implement and very efficient as they have nearly no loss in calculation time in the coupling. However, one has to keep in mind that this is only valid for loosely-coupled systems. The equilibrium equations (3.11) and (3.12) are not exactly satisfied, which means restrictions are put on the time-step to have stable coupling. If added-mass is large in the system, this is not satisfactory, meaning that within each time-step multiple iterations are needed. Algorithms for loosely-coupled systems have been shown to suffer from poor convergence and instabilities with strong interaction [5, 9].

5.1.2 Implicit coupling

One speaks of *implicit coupling* if several sub-iterations are performed within a coupling scheme in order advance a time-step. Especially for an FSI system in which both fluid

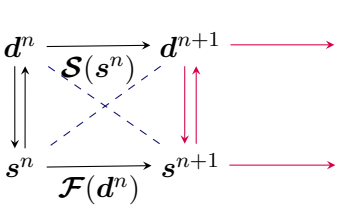
and solid are heavily influenced by one another, it is important that both equations (5.2) and (5.3) are satisfied well enough. Such FSI systems have a fluid density ρ_f and solid density ρ_s comparable with each other, which means large added-mass. Having more iterations to achieve a certain convergence measure makes implicit coupling more stable, which implies that large time-steps can be used. Therefore, strongly-coupled cases thrive by applying these schemes.

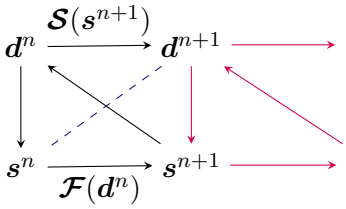
ALGORITHM 5.3: Jacobi Scheme

$k = 0$
$\tilde{\boldsymbol{s}}^0 = \boldsymbol{\mathcal{F}}(\boldsymbol{d}^0)$ and $\tilde{\boldsymbol{d}}^0 = \boldsymbol{\mathcal{S}}(\boldsymbol{s}^0)$
$\boldsymbol{r}^0 = \tilde{\boldsymbol{d}}^0 - \boldsymbol{d}^0$
while $\|\boldsymbol{r}^k\|_2 > \epsilon_0$ **do**
  $\boldsymbol{d}^{k+1} = \tilde{\boldsymbol{d}}^k$ and $\boldsymbol{s}^{k+1} = \tilde{\boldsymbol{s}}^k$
  $\tilde{\boldsymbol{s}}^{k+1} = \boldsymbol{\mathcal{F}}(\boldsymbol{d}^{k+1})$ and $\tilde{\boldsymbol{d}}^{k+1} = \boldsymbol{\mathcal{S}}(\boldsymbol{s}^{k+1})$
  $\boldsymbol{r}^{k+1} = \tilde{\boldsymbol{d}}^{k+1} - \boldsymbol{d}^{k+1}$
  $k = k + 1$
end

ALGORITHM 5.4: Gauß-Seidel scheme

$k = 0$
$\tilde{\boldsymbol{d}}^0 = \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}^0)$
$\boldsymbol{r}^0 = \tilde{\boldsymbol{d}}^0 - \boldsymbol{d}^0$
while $\|\boldsymbol{r}^k\|_2 > \epsilon_0$ **do**
  $\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \boldsymbol{r}^k = \tilde{\boldsymbol{d}}^k$
  $\tilde{\boldsymbol{s}}^{k+1} = \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}^{k+1})$
  $\boldsymbol{r}^{k+1} = \tilde{\boldsymbol{d}}^{k+1} - \boldsymbol{d}^{k+1}$
  $k = k + 1$
end





The most basic schemes are the equivalents of algorithms 5.1 and 5.2. They are called the Jacobi (parallel) iterative scheme (algorithm 5.3) and the Gauß-Seidel (serial) iterative scheme (algorithm 5.4). Due to usage of new data, Gauß-Seidel iterations are generally assumed to perform better. In order to stabilize convergence more for the Gauß-Seidel scheme, often an under-relaxation factor can ω can be used

$$\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \omega \boldsymbol{r}^k \,. \tag{5.4}$$

This way, very large jumps in between steps are avoided, by tuning down the convergence residuals. Convergence rates are are also decreased however. In order to keep convergence rates optimal, a dynamically varying relaxation factor ω^k can be opted for. Aitkens under-relaxation scheme uses information based on the previous time-step to minimize coupling iterations, it calculates ω^k as follows

$$\omega^k = \omega^{k-1} \frac{\left(\boldsymbol{r}^{k-1}\right)^T \left(\boldsymbol{r}^k - \boldsymbol{r}^{k-1}\right)}{\left(\boldsymbol{r}^k - \boldsymbol{r}^{k-1}\right)^T \left(\boldsymbol{r}^k - \boldsymbol{r}^{k-1}\right)} \,. \tag{5.5}$$

Although under-relaxation factors can be of great use, there have been many cases in which bad convergence, or even no convergence has been observed for the Jacobi or the Gauß-Seidel scheme. Especially for incompressible fluids these schemes tend to show stability issues. To get more accuracy, derivatives $\boldsymbol{\mathcal{F}}'$ and $\boldsymbol{\mathcal{S}}'$ have to be used to achieve a better than first order coupling. However, assuming $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{S}}$ are black-box solvers, such

solver-dependent characteristics can not be obtained, meaning approximations have to be sought.

5.2 IQN-ILS

The **I**nterface **Q**uasi-**N**ewton algorithm with an approximation for the **I**nverse of the Jacobian from a **L**east-**S**quares model (IQN-ILS) calculates approximations of the Jacobians by gradually building them based on residual data.

5.2.1 The algorithm

A first step, is to rewrite the FSI problem as a non-linear system of equations dependent on the interface displacement

$$\mathcal{R}(\boldsymbol{d}) \coloneqq \mathcal{S} \circ \mathcal{F}(\boldsymbol{d}) - \boldsymbol{d} = \boldsymbol{0}\,. \tag{5.6}$$

Equation (5.6) can be solved by means of the Newton-Raphson method.

> **Remark 3.** *In what follows, k denotes the Newton-Raphson iteration at time step $n+1$. The indication of this time step $n+1$ is mostly omitted to avoid hard to read formulas, so $\boldsymbol{d}^k \coloneqq \boldsymbol{d}^{n+1,k}$.*

$$\begin{cases} \mathcal{R}'^k \Delta \boldsymbol{d}^k = -\boldsymbol{r}^k\,, & \text{(5.7a)} \\ \boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \Delta \boldsymbol{d}^k\,, & \text{(5.7b)} \end{cases}$$

where the residuals at iteration k are given by

$$\boldsymbol{r}^k = \mathcal{R}(\boldsymbol{d}^k) = \mathcal{S} \circ \mathcal{F}(\boldsymbol{d}^k) - \boldsymbol{d}^k = \tilde{\boldsymbol{d}}^k - \boldsymbol{d}^k\,. \tag{5.8}$$

The tilde is put on *structure solver output* because the *input* for the *fluid solver* is different and thus making it easier to differentiate $\tilde{\boldsymbol{d}}^k$ and $\boldsymbol{d}^k$ further ahead. Furthermore $\mathcal{R}'^k \coloneqq \mathcal{R}'(\boldsymbol{d}^k)$ is written, corresponding to the Jacobian matrix of $\mathcal{R}$ evaluated at $\boldsymbol{d}^k$.
Before starting the Newton-Raphson method however, usually an initial guess is calculated by means of an extrapolation method, such as the second-order accurate formula,

$$\boldsymbol{d}^{n+1,0} = \frac{5}{2}\boldsymbol{d}^n - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2}\,. \tag{5.9}$$

Extrapolation formulas considering less previous time-steps are used for the first two time steps.

As the Newton-Raphson method is an iterative method, convergence is achieved if $\|\boldsymbol{r}^k\|_2 \leq \epsilon_0$, in which ϵ_0 represents the convergence tolerance. Nonetheless, a straightforward method is not yet obtained; $\mathcal{F}$ and $\mathcal{S}$ are black-box solvers and thus the Jacobian of $\mathcal{R}$ can not be exactly calculated in equation (5.7a). Hence an approximation $\widehat{\mathcal{R}'}$ of the Jacobian $\mathcal{R}'$ is sought for, which means the method is Quasi-Newton. More correctly, as will be shown later (due to being more advantageous in terms of memory and speed) the

inverse of the Jacobian will be approximated, leading to

$$\boldsymbol{d}^{k+1} = \boldsymbol{d}^k + \widehat{\left(\boldsymbol{\mathcal{R}}'^k\right)}^{-1}(-\boldsymbol{r}^k)\,. \tag{5.10}$$

As is often the case, it is not necessary to calculate $\widehat{\left(\boldsymbol{\mathcal{R}}'^k\right)}^{-1}$ explicitly, but we may carry out the calculation of the matrix-vector product $\widehat{\left(\boldsymbol{\mathcal{R}}'^k\right)}^{-1}(-\boldsymbol{r}^k)$ immediately. The strategy in doing this is by building up information in each Quasi-Newton iteration. More specifically each coupling iteration k, the structure solver output $\tilde{\boldsymbol{d}}^k$ and the residual $\boldsymbol{r}^k$ are stored, and subtracted with their corresponding value at the previous time step. This gives the following sets that grow with each coupling iteration

$$\boldsymbol{r}^k, \boldsymbol{r}^{k-1}, \ldots, \boldsymbol{r}^1, \boldsymbol{r}^0 \xrightarrow{\Delta\boldsymbol{r}^{k-1} := \boldsymbol{r}^k - \boldsymbol{r}^{k-1}} \Delta\boldsymbol{r}^{k-1}, \Delta\boldsymbol{r}^{k-2}, \ldots, \Delta\boldsymbol{r}^1, \Delta\boldsymbol{r}^0\,, \tag{5.11}$$

$$\tilde{\boldsymbol{d}}^k, \tilde{\boldsymbol{d}}^{k-1}, \ldots, \tilde{\boldsymbol{d}}^1, \tilde{\boldsymbol{d}}^0 \xrightarrow[\Delta\tilde{\boldsymbol{d}}^{k-1} := \tilde{\boldsymbol{d}}^k - \tilde{\boldsymbol{d}}^{k-1}]{} \Delta\tilde{\boldsymbol{d}}^{k-1}, \Delta\tilde{\boldsymbol{d}}^{k-2}, \ldots, \Delta\tilde{\boldsymbol{d}}^1, \Delta\tilde{\boldsymbol{d}}^0\,. \tag{5.12}$$

Each of the newly formed sets are put together to form the matrices

$$\underline{\mathbf{V}}^k = \begin{bmatrix} \Delta\boldsymbol{r}^{k-1} & \Delta\boldsymbol{r}^{k-2} & \ldots & \Delta\boldsymbol{r}^1 & \Delta\boldsymbol{r}^0 \end{bmatrix}, \tag{5.13}$$

and

$$\underline{\mathbf{W}}^k = \begin{bmatrix} \Delta\tilde{\boldsymbol{d}}^{k-1} & \Delta\tilde{\boldsymbol{d}}^{k-2} & \ldots & \Delta\tilde{\boldsymbol{d}}^1 & \Delta\tilde{\boldsymbol{d}}^0 \end{bmatrix}. \tag{5.14}$$

Consecutive time steps often show very similar behaviour. Hence previous time-steps can be used; $\underline{\mathbf{V}}^k$ and $\underline{\mathbf{W}}^k$ are combined with q earlier corresponding matrices (if they exist), giving

$$\mathbf{V}^k = \begin{bmatrix} \underline{\mathbf{V}}^k & \mathbf{V}^n & \ldots & \mathbf{V}^{n-q+2} & \mathbf{V}^{n-q+1} \end{bmatrix}, \tag{5.15}$$

and

$$\mathbf{W}^k = \begin{bmatrix} \underline{\mathbf{W}}^k & \mathbf{W}^n & \ldots & \mathbf{W}^{n-q+2} & \mathbf{W}^{n-q+1} \end{bmatrix}. \tag{5.16}$$

As mentioned by Degroote [11], coupling iterations are remarkably accelerated by using previous time steps.

Denote the amount of rows and columns of $\mathbf{V}^k$ and $\mathbf{W}^k$ by u and v. Usually the $v \leq u$, but it can occur due to usage of q previous time steps that v has to be limited to u by removing columns from the right.

The vector $-\boldsymbol{r}^k$ in equation (5.7a) can be seen as the difference between the desired residual $\mathbf{0}$ and the one obtained at the current coupling iteration $\boldsymbol{r}^k$, which will be defined as $\Delta\boldsymbol{r} := \mathbf{0} - \boldsymbol{r}^k$. This vector is approximated as a linear combination in the $\Delta\boldsymbol{r}^i$ $(i = 0, \ldots, k-1)$:

$$\Delta\boldsymbol{r} \approx \mathbf{V}^k\boldsymbol{c}^k\,, \tag{5.17}$$

where $\boldsymbol{c}^k \in \mathbb{R}^{v\times 1}$ represent the coefficients in the decomposition. Due to generally having $v \leq u$, an overdetermined system of equations has to be solved. This can be done by the

least-squares solution using the normal equations

$$\boldsymbol{c}^k = \left(\mathbf{V}^{k^T}\mathbf{V}^k\right)^{-1}\mathbf{V}^{k^T}\Delta\boldsymbol{r}\,. \tag{5.18}$$

Although this formula seems fairly straightforward, it is not used due to increasing instability for increasing values of v, the amount of columns of $\mathbf{V}^k$. Since usage of previous time steps means more columns, this encourages instability even more. Henceforth a QR-decomposition method is used on the basis of Householder transformations

$$\mathbf{V}^k = \mathbf{Q}^k\mathbf{R}^k\,, \tag{5.19}$$

in which $\mathbf{Q}^k \in \mathbb{R}^{u\times v}$ is orthogonal and $\mathbf{R}^k \in \mathbb{R}^{v\times v}$ is upper triangular. Every coupling iteration k, this decomposition has to be recalculated. The computational burden however is much lower generally compared to those of $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{S}}$. Next, the coefficient vector $\boldsymbol{c}^k$ is computed from the triangular system

$$\mathbf{R}^k\boldsymbol{c}^k = \mathbf{Q}^{k^T}\Delta\boldsymbol{r}\,, \tag{5.20}$$

using back substitution.

> **Remark 4.** *Solving the system with back substitution can only be performed if none of the diagonal elements of $\mathbf{R}^k$ are (almost) zero. Therefore corresponding rows in $\mathbf{V}^k$ and $\mathbf{W}^k$ have to be removed. Afterwards the QR-decomposition is executed again. This procedure is repeated until no (nearly zero) elements are left.*

In an analogous manner to equation (5.17), the $\Delta\tilde{\boldsymbol{d}}$ corresponding to $\Delta\boldsymbol{r}$ is calculated,

$$\Delta\tilde{\boldsymbol{d}} = \mathbf{W}^k\boldsymbol{c}^k\,. \tag{5.21}$$

Using the fact that $\boldsymbol{r}^k = \tilde{\boldsymbol{d}}^k - \boldsymbol{d}^k$, it follows that

$$\Delta\boldsymbol{r} = \Delta\tilde{\boldsymbol{d}} - \Delta\boldsymbol{d}\,. \tag{5.22}$$

Thus, substituting (5.21) into the previous formula gives

$$\Delta\boldsymbol{d} = \mathbf{W}^k\boldsymbol{c}^k - \Delta\boldsymbol{r}\,. \tag{5.23}$$

As $\boldsymbol{c}^k$ is calculated from $\Delta\boldsymbol{r}$, (5.23) can be seen as a way to approximate the matrix-vector product consisting of the approximation of the inverse of the Jacobian and $\Delta\boldsymbol{r}$, in summary

$$\Delta\boldsymbol{d} = \widehat{\left(\boldsymbol{\mathcal{R}}'^k\right)^{-1}}\Delta\boldsymbol{r} = \mathbf{W}^k\boldsymbol{c}^k + \boldsymbol{r}^k\,. \tag{5.24}$$

Since the method is especially designed for having $v \leq u$ columns in $\mathbf{V}^k$, the amount of storage in calculating this matrix-vector product is $u \times v$ instead of u^2, hence speeding up the coupling significantly. The IQN-ILS method is described in Algorithm 5.5 on p.41.

5.2.2 Investigation of adaptations

As part of the research objectives, if possible, an upgrade of the IQN-ILS method is sought to reduce the amount of coupling iterations as much as possible. The reasoning behind this is that solvers $\mathcal{F}$ (mostly this one) and $\mathcal{S}$ are the most expensive in one iteration, and thus even having one iteration less on average would already be very beneficial.

Extensive investigation has been reported in [11] (Degroote) in order to draw out the maximum potential of the algorithm. A short summary of the investigated parts include

- use of a fixed reference ($i = 0, \ldots, k-1$):

$$\Delta \boldsymbol{r}^i \coloneqq \boldsymbol{r}^{i+1} - \boldsymbol{r}^0 \,,$$
$$\Delta \boldsymbol{r}^i \coloneqq \boldsymbol{r}^i - \boldsymbol{r}^k \,,$$

  instead of using subsequent coupling iterations $\Delta \boldsymbol{r}^{k-1} \coloneqq \boldsymbol{r}^k - \boldsymbol{r}^{k-1}$. Numerical experiments show convergence of the Newton-Raphson coupling is similar. Moreover, subsequent iterations allow for an easier comparison with Aitken relaxation;

- updating $\mathbf{V}^k$ and $\mathbf{W}^k$ by adding new columns from the right. Numerical experiments from the past have shown faster convergence by adding from the left, hence this is not done;

- calculating $\widehat{\left(\mathcal{R}'^k\right)}^{-1}$ explicitly instead of directly as a matrix-vector product. Depending on the software, users sometimes have the option to specify this option. In general it is preferred to not do this if the specific matrix is not requested for post-processing are other practices, as more calculations are needed in total;

- calculating $\mathcal{R}'^k$ with a finite difference approximation or solving (5.7a) by means of a matrix-free Krylov solver. These methods request multiple evaluation of the black-box solver $\mathcal{R}$ within each Nexton iteration, compared with IQN-ILS needing only a single evaluation. Consequently these methods are very expensive in practice;

- secant methods, rank-one update methods, rank-two update methods, Broyden's method, Pearson's method and other mathematical-based methods are not preferred as $\mathcal{R}'^k$ most of the times must be symmetric positive definite, IQN-ILS is faster than Broyden or any other column-update methods and IQN-ILS can be reformulated as rank-one update quasi-Newton method;

- using different decomposition than QR like SVD does not show significant improvement in efficiency.

Having this list, a lot of directions to seek further improvement are being removed. The following suggestions have not been mentioned yet however, and can potentially provide this improvement.

Generally the initial guess is now given by $\boldsymbol{d}^{n+1,0} = \frac{5}{2}\boldsymbol{d}^{n} - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2}$, or a lower/higher order accurate extrapolation. As FSI is relatively specific topic, a tailor-made extrapolation for the FSI problem at hand might provide closer guesses each time-step during a complete simulation, resulting in less amount of coupling iterations. Such an idea is investigated in section 5.3.

Secondly, the use of a second order Newton iteration method (s.a. Euler-Chebyshev method),

$$\boldsymbol{r}^{k} + \boldsymbol{\mathcal{R}}'^{k}\Delta\boldsymbol{d}^{k} + \left(\Delta\boldsymbol{d}^{k}\right)^{T}\boldsymbol{\mathcal{R}}''^{k}\Delta\boldsymbol{d}^{k}\,,$$

in general provide very accurate solutions, as more information of the system is used to gear fluid and solid behaviour to one another. Third order tensors like $\boldsymbol{\mathcal{R}}''^{k}$ are very expensive to calculate however. Moreover, due to the solvers $\boldsymbol{\mathcal{F}}$ and $\boldsymbol{\mathcal{S}}$ being black-box solvers, it would take many computations in order to find an approximation of $\boldsymbol{\mathcal{R}}''^{k}$. Hence further ideas are not be explored in this direction.

ALGORITHM 5.5: The interface quasi-Newton algorithm with an approximation for the inverse of the Jacobian from a least-squares model (IQN-ILS)

$k = 0$
$\boldsymbol{d}^{n+1,0} = \frac{5}{2}\boldsymbol{d}^{n} - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2} \quad (\boldsymbol{d}^{0} := \boldsymbol{d}^{n+1,0})$
$\tilde{\boldsymbol{d}}^{0} = \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}^{0})$
$\boldsymbol{r}^{0} = \tilde{\boldsymbol{d}}^{0} - \boldsymbol{d}^{0}$
while $\|\boldsymbol{r}^{k}\|_2 > \epsilon_0$ **do**
  if $k = 0$ *and* ($q = 0$ *or* $n = 0$) **then**
    $\boldsymbol{d}^{k+1} = \boldsymbol{d}^{k} + \omega\boldsymbol{r}^{k}$
  else
    construct $\mathbf{V}^{k}$ and $\mathbf{W}^{k}$ as in eqs. (5.13), (5.14), (5.15) and (5.15)
    calculate QR-decomposition $\mathbf{V}^{k} = \mathbf{Q}^{k}\mathbf{R}^{k}$
    solve $\mathbf{R}^{k}\boldsymbol{c}^{k} = -\mathbf{Q}^{k^{T}}\boldsymbol{r}^{k}$
    $\boldsymbol{d}^{k+1} = \boldsymbol{d}^{k} + \mathbf{W}^{k}\boldsymbol{c}^{k} + \boldsymbol{r}^{k}$
  end
  $\tilde{\boldsymbol{d}}^{k+1} = \boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}^{k+1})$
  $\boldsymbol{r}^{k+1} = \tilde{\boldsymbol{d}}^{k+1} - \boldsymbol{d}^{k+1}$
  $k = k + 1$
end

5.3 Extrapolation methods

First a derivation is given of the second-order accurate extrapolation method (5.9). The interface-displacement vector $\boldsymbol{d}$ is a solution vector obtained at each time-step n from system (5.1). However, looking at this vector as continuous function $\boldsymbol{d}(t)$ in time, consecutive Taylor approximations can be applied with a perturbation of respectively Δt, $2\Delta t$

and $3\Delta t$, giving

$$-5\boldsymbol{d}^{n} = -5\boldsymbol{d}^{n+1} + 5(\boldsymbol{d}')^{n+1}\Delta t - \frac{5}{2}(\boldsymbol{d}'')^{n+1}\Delta t^2 - \mathcal{O}(\Delta t^3)\,, \tag{5.25}$$

$$4\boldsymbol{d}^{n-1} = 4\boldsymbol{d}^{n+1} - 8(\boldsymbol{d}')^{n+1}\Delta t + 8(\boldsymbol{d}'')^{n+1}\Delta t^2 - \mathcal{O}(\Delta t^3)\,, \tag{5.26}$$

$$-\boldsymbol{d}^{n-2} = -\boldsymbol{d}^{n+1} + 3(\boldsymbol{d}')^{n+1}\Delta t - \frac{9}{2}(\boldsymbol{d}'')^{n+1}\Delta t^2 - \mathcal{O}(\Delta t^3)\,, \tag{5.27}$$

in which $(\boldsymbol{d}')^{n+1}$ denotes the interface velocity (for convenience not written with the letter 'v'; $\mathbf{v}$ describes the flow velocity in the governing equations of the fluid in section 3.3 and $\boldsymbol{v}$ in (5.1) denotes the flow variables in the discrete FSI problem) and $(\boldsymbol{d}'')^{n+1}$ denotes the interface acceleration. Henceforth adding the equations (5.25), (5.26) and (5.27) and extracting $\boldsymbol{d}^{n+1}$ gives

$$\boldsymbol{d}_1^{n+1} = \frac{5}{2}\boldsymbol{d}^{n} - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2} + \mathcal{O}(\Delta t^2)\,, \tag{5.28}$$

which is a second-order accurate time method. As mostly all FSI problems will be solved on small time scales, it is assumed that $(\boldsymbol{d}'')^{n+1} \approx 0$, this way the initial guess is obtained.

Another way of doing extrapolation is by using the unique interpolation function (for example with the method of Lagrange) resulting from previously obtained data. This gives the following formula

$$\boldsymbol{d}_2^{n+1} = 3\boldsymbol{d}^{n} - 3\boldsymbol{d}^{n-1} + \boldsymbol{d}^{n-2}\,. \tag{5.29}$$

This formula can also be derived from a Taylor approach, similar to eq. (5.9). The derivation is given in Appendix B. The first extrapolation $\boldsymbol{d}_1^{n+1}$ bases its guess on a more gradual weighting of the previous time-steps $\boldsymbol{d}^{n}, \boldsymbol{d}^{n-1}$ and $\boldsymbol{d}^{n-2}$, seemingly making it a more reasonable approach at first hand. To get a more physical grasp of the difference between them, they are compared mathematically. Writing the extrapolation (5.9) in function of the other gives

$$\boldsymbol{d}_1^{n+1} = \boldsymbol{d}_2^{n+1} - \left[\frac{1}{2}\boldsymbol{d}^{n} - \boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2}\right] \tag{5.30}$$

$$= \boldsymbol{d}_2^{n+1} - \frac{\Delta t^2}{2}\left[\frac{\boldsymbol{d}^{n} - 2\boldsymbol{d}^{n-1} + \boldsymbol{d}^{n-2}}{\Delta t^2}\right] \tag{5.31}$$

$$= \boldsymbol{d}_2^{n+1} - \frac{\Delta t^2}{2}\left[(\boldsymbol{d}'')^{n-1} + \mathcal{O}(\Delta t^2)\right] \tag{5.32}$$

$$= \boldsymbol{d}_2^{n+1} - \frac{\Delta t^2}{2}(\boldsymbol{d}'')^{n-1} + \mathcal{O}(\Delta t^4)\,. \tag{5.33}$$

This means that if the acceleration $(\boldsymbol{d}'')^{n-1}$ is increasing at $n-1$ that one should guess lower for the displacement, and vice versa if the acceleration is decreasing.

However, the question that now arises is which one is more applicable in which situation, and whether there is a method that may just be preferred over the other one in all cases. This will be investigated by a small experiment in MATLAB. One period of a discrete sinus-wave is perturbed at each point by a uniform, random value in $[-\sigma, \sigma]$. Next methods (5.9), (5.29) and the following extrapolation methods using four earlier

time-steps,

$$\boldsymbol{d}^{n+1} = 4\boldsymbol{d}^{n} - 6\boldsymbol{d}^{n-1} + 4\boldsymbol{d}^{n-2} - \boldsymbol{d}^{n-3}\,, \tag{5.34}$$

which is obtained for the interpolation polynomial using four points and is fourth-order accurate in time, and

$$\boldsymbol{d}^{n+1} = \frac{1}{35}\left[104\boldsymbol{d}^{n} - 112\boldsymbol{d}^{n-1} + 56\boldsymbol{d}^{n-2} - 11\boldsymbol{d}^{n-3}\right]\,, \tag{5.35}$$

which is obtained also by means of Taylor approximations and is third-order accurate, are compared for $\sigma \in \{0.1, 0.01, 0.001\}$. This results in figures 5.1, 5.2 and 5.3.

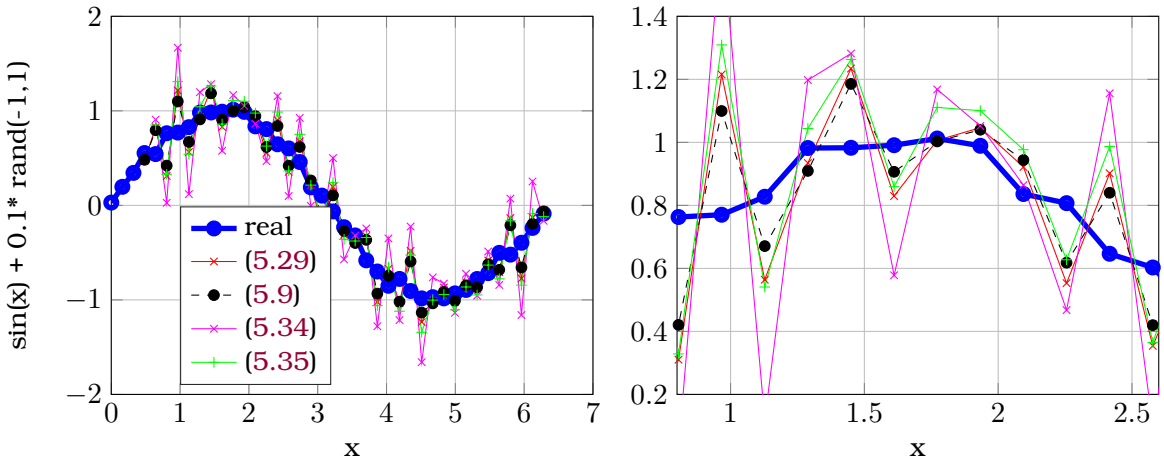


Figure 5.1: *Extrapolation methods test for* $\sigma = 0.1$.

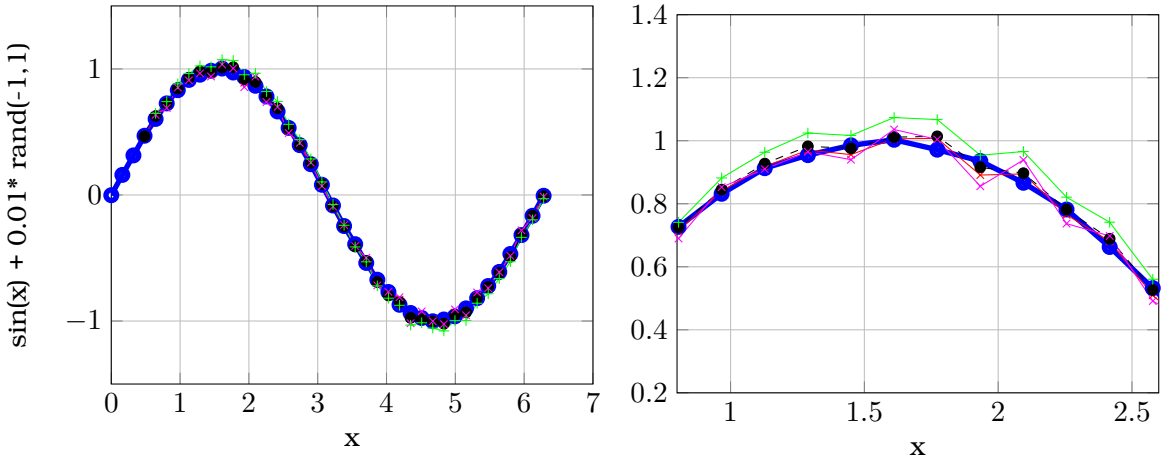


Figure 5.2: *Extrapolation methods test for* $\sigma = 0.01$. *Legend is given in figure 5.1.*

Before making stating any conclusions, it is important to keep in mind that this is a purely experimental case, and results are highly simplified compared to any other FSI problem. From the above figures, there are two main conclusions that can be made

- Methods considering more previous data do not necessarily give more accurate data;

- Higher values of σ (more variation) seem to be predicted the best by Taylor-based

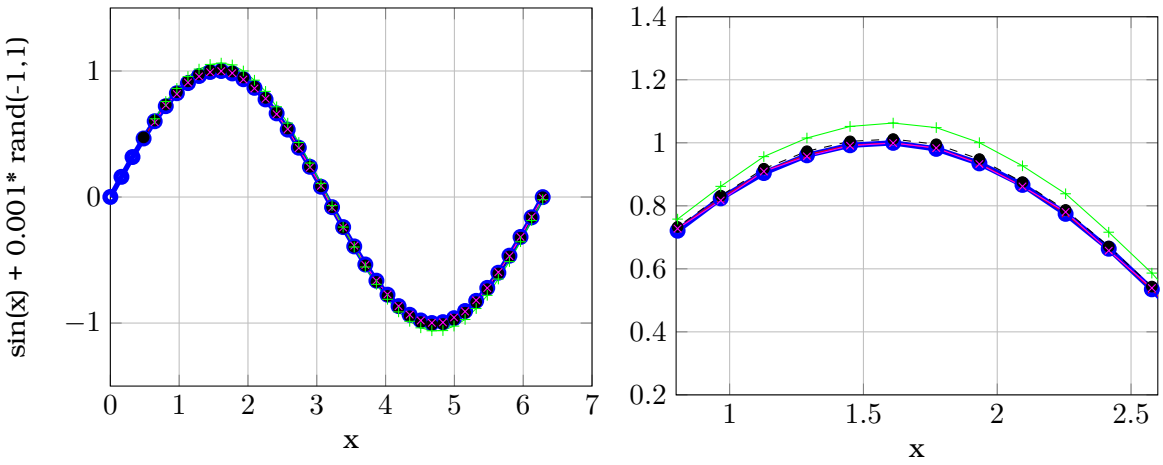


Figure 5.3: *Extrapolation methods test for* $\sigma = 0.001$. *Legend is given in figure 5.1*

methods, and lower values of σ (less variation) best predicted by the interpolating polynomials.

Although our first conclusion makes it seem as if methods using more past data are not worth it to be considered, Sachs et al [38] performed an extrapolation study on the Turek benchmark case [44] and concluded that using more past data for extrapolation of the interface displacement can in fact speed up computations a lot. Therefore, other cases (more realistic FSI cases) should still be considered to verify or restate any conclusions and consequently develop/choose a wise extrapolation method.
However, from this already a suggestion for an improvement can be made. One possibility would be to utilize a weighting parameter $\omega^{n+1} \in [0, 1]$ which describes the relative variability of the data up to and including time t^n; $\omega^{n+1} = 1$ denotes heavy variability and $\omega^{n+1} = 0$ means a smooth curve. Then, an extrapolation could be

$$\boldsymbol{d}^{n+1} = \omega^{n+1}\boldsymbol{d}_1^{n+1} + (1 - \omega^{n+1})\boldsymbol{d}_2^{n+1} \tag{5.36}$$

$$= \boldsymbol{d}_2^{n+1} - \omega^{n+1}\left[\frac{1}{2}\boldsymbol{d}^n - \boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2}\right]. \tag{5.37}$$

This way one would for example compensate for the initial variability in the displacement generated by the Pressure Fluctuation Method (PFM) (section 6.3) and once a modal frequency is reached, switch to having an extrapolation method suited for more smooth displacements.

One of most important features w.r.t. designing such a weighting parameter ω^{n+1} is to do it in a cheap and memory cost-effective way. For this reason, it is chosen that ω^{n+1} must not depend on displacement data from time-steps $(n-3, n-4, \dots)$ before the ones being used in the actual extrapolation method (5.36). This also makes sense as history dependency of the interface-displacement is problem-dependent (strong or weak coupling). In case of a PFM simulation, one could have a lot of variability at the start of the simulation (due to effects of randomness in PFM), whereas after quite some time, there might be smooth behaviour in the solid movement (e.g. reaching the modal frequency of a rod).

Including too much data in ω^{n+1} would mean that variability at a certain time could be corrupted, and hence over- or under-predicting the weighting.

Irrespective of the actual formula to model the variability, as described above, method (5.36) suggests calculation of single scalar value ω^{n+1}. Considering computationally expensive problems, it is worth it to hypothetically examine the idea of using a vector $\boldsymbol{\omega}^{n+1}$, having a value for all degrees of freedom. There are two reasons for this:

1. High performance computing software often prefers usage of distributed vectors, i.e. all processors contain a piece of the displacement vector $\boldsymbol{d}^n$. Hence, extra communication between the processors will be added in order to obtain a single value ω^{n+1}.

2. Structural equations are often accompanied with restrictive movement at the borders or in a specific direction. They are exactly specified, often very smooth in nature. If $\omega^{n+1} \approx 1$, then regardless of these constraints, heavy variability is assumed.

Based on the above, in this thesis a vectorized version will be implemented, reformulating eq. (5.36) as

$$\boldsymbol{d}^{n+1} = \boldsymbol{\omega}^{n+1} \odot \boldsymbol{d}_1^{n+1} + (\mathbb{1} - \boldsymbol{\omega}^{n+1}) \odot \boldsymbol{d}_2^{n+1}\,, \tag{5.38}$$

in which $\odot$ represents the Hadamard product, also known as element-wise vector multiplication. It is very important to keep in mind that this extrapolation method can only be beneficial for computationally expensive simulations. Using an element-wise operator introduces $\mathcal{O}(u)$ extra computations (u as in section 5.2).

Taking a look at figure 5.1, one sees the variability is associated with the actual data $\boldsymbol{d}^n$ differing much from the extrapolation guesses $\boldsymbol{g}^n$ that were made before. Henceforth, for each degree of freedom $i \in \{1, \ldots, u\}$, the distance between those two $|g_i^n - d_i^n|$ is incorporated in ω_i^{n+1}. Moreover, due to the constraint $\omega_i^{n+1} \in [0, 1]$, some type of normalization must be applied. All of the previous leads to the following definition

$$\omega_i^{n+1} := \frac{|g_i^n - d_i^n|}{|g_i^n - d_i^n| + |g_i^n - d_i^{n-1}|}\,. \tag{5.39}$$

In figure 5.4 supportive simplifications of data sets with guesses are given to display the behaviour of the weighting function given in eq. (5.39). In case the variability is large, ω_i^{n+1} goes to one, and if this is not the case, then ω_i^{n+1} tends to go towards zero. Thus $\omega_i^{n+1} \in [0, 1]$ and varies between the extrapolations $\boldsymbol{d}_1^{n+1}$ and $\boldsymbol{d}_2^{n+1}$ in the correct manner according the variability of the displacements.

Similarly as before, a small test is performed in MATLAB in order to get a taste of the efficiency of the suggested extrapolation (5.38). Because solely a **new** method is described using three past time-steps, namely $n-2, n-1$ and n, extrapolations (5.34) and (5.35) are left out (they use four time-steps). In the following, the **new** extrapolation's results are denoted by the subscript 'new', e.g. $\boldsymbol{d}_{\text{new}}^{n+1}$. As before, a subscript 1 refers to eq. (5.9) and a subscript 2 refers to eq. (5.29).

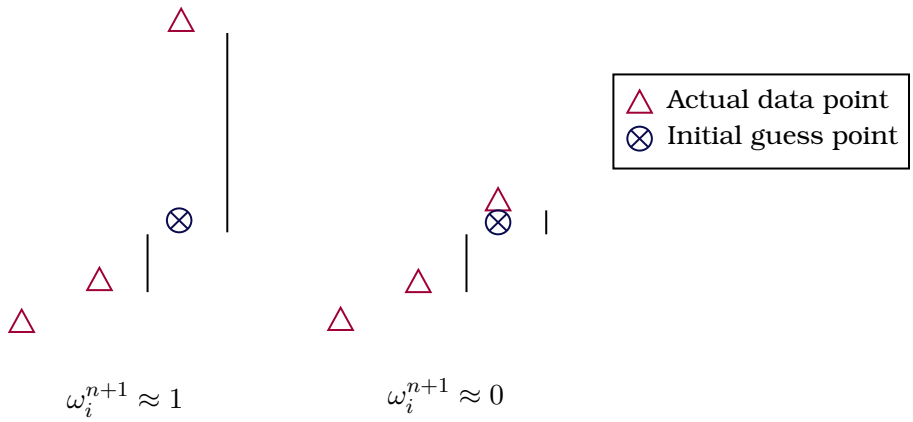


Figure 5.4: *An explanatory example of data points showing the behaviour of the weighting function (5.39). On the left it can be seen that if the difference between the guessed value and the actual displacement is large, which denotes much variability, the weighting leans towards one. On the other hand, if the values are close, there is nearly no variability and thus the weighting approaches zero.*

The sine waves $A \sin \mathbf{x} + \sigma \cdot \text{rand}\,(-1, 1)$ are used as a first attempt, in which $\mathbf{x}$ has N points. Combinations of different parameter values are chosen:

- $N = \begin{bmatrix} 20 & 40 & 60 & 80 \end{bmatrix}$;
- $A = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \end{bmatrix}$;
- $\sigma = \begin{bmatrix} 0.1 & 0.01 & 0.001 \end{bmatrix}$.

For all of the settings s (specific set of N, A and σ) and extrapolations $*$ considered, an error value $e^j_{s,*} = \sum_{n=4}^{N} \|\boldsymbol{g}^n_* - \boldsymbol{d}^n_*\|$ is calculated, where j denotes that this specific setting is examined for I iterations. The mean error $e_{s,*} = \left[\sum_{j=1}^{I} e^j_{s,*}\right] / I$ is thereafter extracted. In the experiment performed in MATLAB $I = 10000$.
In order to get a final error, the sum will be taken over all settings. Before doing such a summation, the mean errors $e_{s,*}$ will be normalized according to

$$\bar{e}_{s,*} = \frac{e_{s,*}}{\max\limits_{*} \{e_{s,*}\}}\,, \tag{5.40}$$

allocating equal significance to all different settings. For example, having extrapolations in the order (5.9), (5.29) and (5.36) with specific error values $e_{s,*}$ the normalization proceeds gives the values $\bar{e}_{s,*}$

$$0.0262 \quad 0.0226 \quad 0.0222 \rightarrow 1.0000 \quad 0.8604 \quad 0.8466 \tag{5.41}$$

Next, summing these values gives the final error

$$E_* = \sum_s \bar{e}_{s,*}\,. \tag{5.42}$$

For the values of N, A and σ given above, this means the total value for a certain extrapolation method is 60. Performing all the above steps results in table 5.1.

	E_1	E_2	E_{new}
Error sum	54.2850	45.1528	42.5963

Table 5.1: *The resulting errors (5.42) of the extrapolations (5.9), (5.29) and (5.36) resulting from the data* $A \sin \mathbf{x} + \sigma \cdot rand\,(-1, 1)$ *(*$\mathbf{x}$ *having* N *points) are calculated for values* $N = [20 \;\; 40 \;\; 60 \;\; 80]$, $A = [1 \;\; 2 \;\; 3 \;\; 4 \;\; 5]$ *and* $\sigma = [0.1 \;\; 0.01 \;\; 0.001]$.

From the results obtained it can be noticed that there is a lot of potential in the new extrapolation method (5.36). However, as mentioned before w.r.t. more past data in extrapolation methods, other cases should still be considered since the case investigated is very simplified compared to general FSI problems.

5.4 Manifold Mapping

The idea of utilising a different extrapolation method in order to have a better initial guess brings up the question if fluid and structure properties (such as viscosity ν and the modulus of elasticity E) can be of any help in order to devise a smart "FSI-extrapolation". Implicitly this is already being done in equation (5.9). Namely, as can be seen in the derivation via Taylor expansions in equations (5.25), (5.26) and (5.27), the velocity $\boldsymbol{d}'$ of the interface-displacement $\boldsymbol{d}$ is involved.

Explicitly however, this would come down to solving (very) simplified equations of both the fluid and structure. As the extrapolation step is seen as a *preparatory step* before actually solving the FSI problem at the current time step, it is of good practice to keep both parts separated.

Consequently it makes sense to solve a simplified model within the actual coupling algorithm. One of such methods is called the *Manifold Mapping* algorithm, which originates from multi-fidelity optimization [15], and has been applied in an FSI simulation context by Blom et al [6]. In order to realize the application of optimization-based algorithms the nonlinear problem (5.6) is rewritten as $\boldsymbol{d}^* = \arg\min_{\boldsymbol{d}} \|\boldsymbol{\mathcal{R}}(\boldsymbol{d})\|_2 = \arg\min_{\boldsymbol{d}} \|\boldsymbol{\mathcal{S}} \circ \boldsymbol{\mathcal{F}}(\boldsymbol{d}) - \boldsymbol{d}\|_2$. The manifold mapping combines a computationally inexpensive low-fidelity fluid-structure interaction model with a high-fidelity FSI model in order to accelerate the convergence of the coupling iterations of the high-fidelity model.

In order to have a clear understanding of the Manifold Mapping algorithm, general terminology is first used to shortly describe the vital concepts. Afterwards the problem will be converted to an FSI problem.

5.4.1 General algorithm

The fine model is given by $\boldsymbol{f}\colon X \subset \mathbb{R}^m \to X$ evaluating $\boldsymbol{x} \in X$ the control variable of the optimization problem. A fine model *cost function* $\mathcal{F}(\boldsymbol{x}) := \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{q}\|$ is defined to represent the difference between the fine model output $\boldsymbol{f}(\boldsymbol{x})$ and a design specification $\boldsymbol{q}$.

Consequently the following problem is solved

$$\boldsymbol{x}_{\boldsymbol{f}}^{*} = \underset{\boldsymbol{x} \in X}{\arg\min} \|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{q}\| \,. \tag{5.43}$$

The low-fidelity model being used as a surrogate evaluates a control variable $\boldsymbol{z} \in Z$, where $Z \subset \mathbb{R}^n$ and $n \leq m$ (preferably significantly smaller), given by $\boldsymbol{c}^{\boldsymbol{z}} \colon Z \to Z$. The major trick of the manifold mapping algorithm then follows by solving the coarse model $\boldsymbol{c}^{\boldsymbol{z}}$ on the fine data $\boldsymbol{x}$. One achieves this by defining the analogous cost function for the coarse model as $\mathcal{C}(\boldsymbol{x}) \coloneqq \|\boldsymbol{p}^{-1}(\boldsymbol{c}^{\boldsymbol{z}}(\boldsymbol{p}(\boldsymbol{x}))) - \boldsymbol{q}^{l}\|$, in which $\boldsymbol{p} \colon X \to Z$ maps the data between the different models. Consequently one defines $\boldsymbol{c}(\boldsymbol{x}) \coloneqq \boldsymbol{p}^{-1}(\boldsymbol{c}^{\boldsymbol{z}}(\boldsymbol{p}(\boldsymbol{x})))$ on X, resulting into the coarse model optimization problem

$$\boldsymbol{x}_{\boldsymbol{c}}^{*} = \underset{\boldsymbol{x} \in X}{\arg\min} \|\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{q}^{l}\| \,. \tag{5.44}$$

The vector $\boldsymbol{q}^{l}$ represents the design specification of the coarse model at iteration l. Indeed, the minimization process of the fine model is an iterative process in which $\boldsymbol{q}^{l}$ is updated. The mappings $\boldsymbol{p}$ and $\boldsymbol{p}^{-1}$ are obtained by means of radial basis function interpolation. However, due to them being incorporated in $\boldsymbol{c}(\boldsymbol{x})$, these will be implicitly assumed for clarity, and thus not further mentioned.

The question remaining is *how to iteratively update the design specification* $\boldsymbol{q}^{l}$ of the coarse model. In order to achieve this, the manifold mapping defines a mapping $\boldsymbol{S} \colon \boldsymbol{c}(X) \to \boldsymbol{f}(X)$ to correct the coarse model for misalignment compared to the fine model. $\boldsymbol{S}$ is defined as the function mapping $\boldsymbol{c}(\boldsymbol{x}_{\boldsymbol{f}}^{*})$ onto $\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{f}}^{*})$, and the tangent plane for $\boldsymbol{c}(X)$ at $\boldsymbol{c}(\boldsymbol{x}_{\boldsymbol{f}}^{*})$ onto the tangent plane for $\boldsymbol{f}(X)$ at $\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{f}}^{*})$. Hence the following affine function mapping is obtained

$$\boldsymbol{S}\left[\boldsymbol{c}(\boldsymbol{x})\right] = \boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{f}}^{*}) + \bar{\boldsymbol{S}}\left(\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{c}(\boldsymbol{x}_{\boldsymbol{f}}^{*})\right) \text{ with } \bar{\boldsymbol{S}} \coloneqq \boldsymbol{J}_{\boldsymbol{f}}(\boldsymbol{x}_{\boldsymbol{f}}^{*})\boldsymbol{J}_{\boldsymbol{c}}{}^{\dagger}(\boldsymbol{x}_{\boldsymbol{f}}^{*}) \,, \tag{5.45}$$

where $\boldsymbol{J}_{\boldsymbol{f}}$ represents the Jacobian of the fine model and $\boldsymbol{J}_{\boldsymbol{c}}$ represents the Jacobian of the coarse model. The dagger symbol $\dagger$ indicates a pseudo-inverse. This alignment mapping $\boldsymbol{S}$ has the property that again the following holds (locally, see Lemma 3 in [15])

$$\boldsymbol{x}_{\boldsymbol{f}}^{*} = \underset{\boldsymbol{x} \in X}{\arg\min} \|\boldsymbol{S}\left[\boldsymbol{c}(\boldsymbol{x})\right] - \boldsymbol{q}\| \,. \tag{5.46}$$

Moreover, if $\|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{q}\|$ is locally convex at $\boldsymbol{x}_{\boldsymbol{f}}^{*}$, then also $\|\boldsymbol{S}\left[\boldsymbol{c}(\boldsymbol{x})\right] - \boldsymbol{q}\|$ is assumed to be locally convex at $\boldsymbol{x}_{\boldsymbol{f}}^{*}$. Hence one can apply the pseudo-inverse to eq. (5.46) and write

$$\boldsymbol{x}_{\boldsymbol{f}}^{*} = \underset{\boldsymbol{x} \in X}{\arg\min} \|\boldsymbol{c}(\boldsymbol{x}) - \left[\boldsymbol{c}(\boldsymbol{x}_{\boldsymbol{f}}^{*}) - \bar{\boldsymbol{S}}^{\dagger}\left(\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{f}}^{*}) - \boldsymbol{q}\right)\right] \| \,. \tag{5.47}$$

As $\boldsymbol{x}_{\boldsymbol{f}}^{*}$ is the unknown value to be searched for, the norm above will be solved several times in an iterative process. This results in

$$\boldsymbol{x}^{\boldsymbol{l+1}} = \underset{\boldsymbol{x} \in X}{\arg\min} \|\boldsymbol{c}(\boldsymbol{x}) - \boldsymbol{q}^{l}\| \,, \tag{5.48}$$

where $\boldsymbol{q}^l \coloneqq \boldsymbol{c}(\boldsymbol{x}^l) - \bar{\boldsymbol{S}}^\dagger \left(\boldsymbol{f}(\boldsymbol{x}^l) - \boldsymbol{q}\right)$ and $\bar{\boldsymbol{S}}^\dagger$ evaluated at $\boldsymbol{x}^l$.

5.4.2 FSI context

Rephrasing the manifold mapping according to an FSI problem, the dimension of the design space X is equal to the number of degrees of freedom (DOF) at the fine fluid-structure interface. Hence $\boldsymbol{d} \in D$ is written to describe the interface displacement. In what follows, as before, subscript $\boldsymbol{c}$ refers to the coarse model, and the subscript f referring to the fine model is dropped to signify that this is the actual model. This gives

$$\boldsymbol{f}(\boldsymbol{x}) \leftrightarrow \boldsymbol{\mathcal{R}}(\boldsymbol{d}) \qquad \& \ \boldsymbol{q} = \boldsymbol{0}\,, \tag{5.49}$$

$$\boldsymbol{c}(\boldsymbol{x}) \leftrightarrow \boldsymbol{\mathcal{R}}_{\boldsymbol{c}}(\boldsymbol{d}) \coloneqq \boldsymbol{p}^{-1}\left(\boldsymbol{\mathcal{S}}_{\boldsymbol{c}} \circ \boldsymbol{\mathcal{F}}_{\boldsymbol{c}}(\boldsymbol{p}(\boldsymbol{d})) - \boldsymbol{p}(\boldsymbol{d})\right) \qquad \& \ \boldsymbol{q}_l \neq \boldsymbol{0}\,, \tag{5.50}$$

in which $\boldsymbol{x} \in X$ and $\boldsymbol{d} \in D$ with $D \subset \mathbb{R}^{u\times 1}$ (u again the DOF as in section 5.2). Now the mapping $\boldsymbol{S}\colon \boldsymbol{\mathcal{R}}_{\boldsymbol{c}}(D) \to \boldsymbol{\mathcal{R}}(D)$ is used to correct the coarse model for misalignment compared to the fine FSI model, changing eq. (5.46) into

$$\boldsymbol{d}^* = \underset{\boldsymbol{d}\in D}{\arg\min} \|\boldsymbol{S}\left[\boldsymbol{\mathcal{R}}_{\boldsymbol{c}}(\boldsymbol{d})\right]\|\,, \tag{5.51}$$

where

$$\boldsymbol{S}\left[\boldsymbol{\mathcal{R}}_{\boldsymbol{c}}(\boldsymbol{d})\right] = \boldsymbol{\mathcal{R}}(\boldsymbol{d}^*) + \bar{\boldsymbol{S}}\left(\boldsymbol{\mathcal{R}}_{\boldsymbol{c}}(\boldsymbol{d}) - \boldsymbol{\mathcal{R}}_{\boldsymbol{c}}(\boldsymbol{d}^*)\right) \text{ and } \bar{\boldsymbol{S}} \coloneqq \boldsymbol{\mathcal{R}}'(\boldsymbol{d}^*){\boldsymbol{\mathcal{R}}_{\boldsymbol{c}}'}^{\dagger}(\boldsymbol{d}^*)\,, \tag{5.52}$$

Similar as in section 5.2 $\boldsymbol{\mathcal{R}}'$ represents the Jacobian. Define $\boldsymbol{r}^l \coloneqq \boldsymbol{\mathcal{R}}(\boldsymbol{d}^l)$ and $\boldsymbol{r}_{\boldsymbol{c}}^l \coloneqq \boldsymbol{\mathcal{R}}_{\boldsymbol{c}}(\boldsymbol{d}^l)$, then eq. (5.48) rephrases as

$$\boldsymbol{d}^{\boldsymbol{l+1}} = \underset{\boldsymbol{d}\in D}{\arg\min} \|\boldsymbol{\mathcal{R}}_{\boldsymbol{c}}(\boldsymbol{d}) - \boldsymbol{r}_{\boldsymbol{c}}^l + \bar{\boldsymbol{S}}^\dagger \boldsymbol{r}^l\|\,. \tag{5.53}$$

However, the Jacobians used in $\bar{\boldsymbol{S}}^\dagger$ are not known, as the fluid and solid solvers of the coarse and fine models are considered to be black-box functions. For this reason $\bar{\boldsymbol{S}}^\dagger$ is approximated by a sequence $\mathbf{T}^l$, leading to the optimization problem

$$\begin{cases} \boldsymbol{d}^{\boldsymbol{l+1}} = \underset{\boldsymbol{d}\in D}{\arg\min} \|\boldsymbol{\mathcal{R}}_{\boldsymbol{c}}(\boldsymbol{d}) - \boldsymbol{q}^l\|\,, & \text{(5.54a)} \\ \boldsymbol{q}^l = \boldsymbol{r}_{\boldsymbol{c}}^l - \mathbf{T}^l \boldsymbol{r}^l\,. & \text{(5.54b)} \end{cases}$$

The calculations of $\mathbf{T}^l$ are performed following a similar approach as the calculation of the Jacobian in the IQN-ILS method. For $i = 0, \ldots, l-1$, the differences $\Delta\boldsymbol{r}^i \coloneqq \boldsymbol{r}^l - \boldsymbol{r}^i$ are defined (using a fixed-reference), and thereupon merged into the matrices

$$\underline{\mathbf{F}}^l = \begin{bmatrix} \Delta\boldsymbol{r}^{l-1} & \Delta\boldsymbol{r}^{l-2} & \ldots & \Delta\boldsymbol{r}^1 & \Delta\boldsymbol{r}^0 \end{bmatrix}, \tag{5.55}$$

and

$$\underline{\mathbf{C}}^l = \begin{bmatrix} \Delta\boldsymbol{r}_{\boldsymbol{c}}^{l-1} & \Delta\boldsymbol{r}_{\boldsymbol{c}}^{l-2} & \ldots & \Delta\boldsymbol{r}_{\boldsymbol{c}}^1 & \Delta\boldsymbol{r}_{\boldsymbol{c}}^0 \end{bmatrix}. \tag{5.56}$$

Similar as in the IQN-ILS algorithm it is beneficial to reduce the amount of fine model evaluations by including information from q (not the design specification $\boldsymbol{q}$) previous time steps as in equations (5.15) and (5.16), resulting in matrices

$$\mathbf{F}^l = \begin{bmatrix} \underline{\mathbf{F}}^l & \mathbf{F}^n & \dots & \mathbf{F}^{n-q+2} & \mathbf{F}^{n-q+1} \end{bmatrix} , \tag{5.57}$$

and

$$\mathbf{C}^l = \begin{bmatrix} \underline{\mathbf{C}}^l & \mathbf{C}^n & \dots & \mathbf{C}^{n-q+2} & \mathbf{C}^{n-q+1} \end{bmatrix} . \tag{5.58}$$

Denote the amount of rows and columns of $\mathbf{F}^l$ and $\mathbf{C}^l$ by u and v. Due to the usage of q previous time steps, similarly as in IQN-ILS, it may occur that v has to be limited to u removing columns from the right. Once the matrices are obtained, their singular value decompositions $\mathbf{F}^l = \boldsymbol{U_f \Sigma_f V_f}^T$ and $\mathbf{C}^l = \boldsymbol{U_c \Sigma_c V_c}^T$ are calculated.

A desirable property in choosing the way $\mathbf{T}^l$ is calculated would then be that $\lim_{l\to\infty} \mathbf{T}^l = \boldsymbol{\mathcal{R}}_{\boldsymbol{c}}'(\boldsymbol{d^*})\boldsymbol{\mathcal{R}}'^{\dagger}(\boldsymbol{d^*})$. Echeverría proves in [15] that under certain assumptions this holds for the obvious choice of $\mathbf{T}^l = \mathbf{C}^l \left(\mathbf{F}^l\right)^{\dagger}$. Furthermore, the manifold mapping can be based on any matrix $\bar{\boldsymbol{S}}$ for which it holds that $\boldsymbol{\mathcal{R}}_{\boldsymbol{c}}'^{\dagger}(\boldsymbol{d^*})\bar{\boldsymbol{S}}^{\dagger} = \boldsymbol{\mathcal{R}}'^{\dagger}(\boldsymbol{d^*})$. This leads to a more general matrix $\bar{\boldsymbol{S}} \coloneqq \boldsymbol{\mathcal{R}}'(\boldsymbol{d^*})\boldsymbol{\mathcal{R}}_{\boldsymbol{c}}'^{\dagger}(\boldsymbol{d^*}) + \left(\boldsymbol{I} - \boldsymbol{U_c U_c}^T\right)\boldsymbol{A}$ to be approximated, in which $\boldsymbol{A}$ can be any $u \times u$ matrix. The particular cases $\boldsymbol{A} = \left(\boldsymbol{I} - \boldsymbol{U_f U_f}^T\right)$ and $\boldsymbol{A} = \boldsymbol{I}$ are mentioned in [15] to be stabilizing the manifold mapping algorithm considerably. The former is chosen by Blom et al [6] such that more information from the fine model is included, leading to the approximation matrices

$$\mathbf{T}^l = \mathbf{C}^l \left(\mathbf{F}^l\right)^{\dagger} + \left(\boldsymbol{I} - \boldsymbol{U_c U_c}^T\right)\left(\boldsymbol{I} - \boldsymbol{U_f U_f}^T\right) . \tag{5.59}$$

Within each Manifold Mapping iteration the coarse model optimization (5.54a) has to be performed. Rewriting this problem into a non-linear system of equations

$$\boldsymbol{\mathcal{R}_c}(\boldsymbol{d}) = \boldsymbol{\mathcal{S}_c} \circ \boldsymbol{\mathcal{F}_c}(\boldsymbol{d}) - \boldsymbol{d} = \boldsymbol{q}^l , \tag{5.60}$$

shows that this problem can be solved by any other single-level method existing. Hence, often the previously described IQN-ILS Algorithm 5.5 is preferred. The complete manifold mapping applied in an FSI context is given in Algorithm 5.6.

ALGORITHM 5.6: The manifold mapping (MM) algorithm

$l = 0$
$\boldsymbol{d}^{n+1,0} = \frac{5}{2}\boldsymbol{d}^{n} - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2} \quad (\boldsymbol{d}^{0} := \boldsymbol{d}^{n+1,0})$
$\boldsymbol{r}^{0} = \boldsymbol{\mathcal{R}}(\boldsymbol{d}^{0})$
if $q = 0$ *or* $n = 0$ **then**
  $\mathbf{T}^{0} = \boldsymbol{I}$
else
  construct $\mathbf{T}^{l}$ as in eq. (5.59)
end
for $l = 0$ **to** l_{max} **do**
  if $l = 0$ *and* ($q = 0$ *or* $n = 0$) **then**
    $\boldsymbol{d}^{l+1} = \boldsymbol{d}^{l} + \omega \boldsymbol{r}^{l}$
  else
    $\boldsymbol{q}^{l} = \boldsymbol{r}_{\boldsymbol{c}}^{l} - \mathbf{T}^{l}\boldsymbol{r}^{l}$
    $\boldsymbol{d}^{\boldsymbol{l+1}} = \underset{\boldsymbol{d}\in D}{\arg\min} \|\boldsymbol{\mathcal{R}}_{\boldsymbol{c}}(\boldsymbol{d}) - \boldsymbol{q}^{l}\|$
  end
  $\boldsymbol{r}^{l+1} = \boldsymbol{\mathcal{R}}(\boldsymbol{d}^{l+1})$
  if $\|\boldsymbol{r}^{l+1}\|_2 \leq \epsilon_0$ **then**
    break
  end
  construct $\mathbf{F}^{l+1}$ and $\mathbf{C}^{l+1}$ as in eqs. (5.55), (5.56), (5.57) and (5.58)
  calculate singular value decomposition $\mathbf{F}^{l+1} = \boldsymbol{U_f}\boldsymbol{\Sigma_f}\boldsymbol{V_f}^{T}$
  calculate singular value decomposition $\mathbf{C}^{l+1} = \boldsymbol{U_c}\boldsymbol{\Sigma_c}\boldsymbol{V_c}^{T}$
  update $\mathbf{T}^{l+1} = \mathbf{C}^{l+1}\left(\mathbf{F}^{l+1}\right)^{\dagger} + \left(\boldsymbol{I} - \boldsymbol{U_c}\boldsymbol{U_c}^{T}\right)\left(\boldsymbol{I} - \boldsymbol{U_f}\boldsymbol{U_f}^{T}\right)$
  $l = l + 1$
end

Chapter 6

Efficient modelling of fluid behaviour

The development of high-fidelity approaches to model fluid behaviour has been an important topic for several years already as Computational Fluid Dynamics (CFD) as subject exists longer than Fluid-Structure Interaction (FSI). FSI being a combination of CFD and Computational Solid Mechanics (CSM) is at least as involved computationally. High-end CFD methods are therefore inherited to perform efficient FSI.

These methods under consideration mostly arise from the quest for efficient modelling of turbulence. Hence a short description is given of ways to achieve this. Also an efficient calculation procedure, called the PIMPLE algorithm, for solving the Navier-Stokes equations is given that stabilizes solutions and tries to minimize CPU time.

6.1 Turbulence modelling

The chaotic disturbances in a fluid, seemingly having random behaviour are what is classified as turbulence. Most fluid flows in nature are turbulent in nature, making it very interesting to study. Aside from having interest in turbulence due to it being an intriguing feature of nature, it also important to understand it for many engineering applications as it induces a higher diffusivity and enhanced energy dissipation. Understanding turbulence is very difficult however, as it is non-repeatable and has a large range of length and time scales at which eddies develop. These scales vary from the smallest scales (Kolmogorov scales) to the largest scales, having eddies with a size comparable with the geometry, and the range of scales increase with Reynolds number.

Over the past two centuries small bits were discovered by the likes of many great physicists; a brief history is given in [31]. These developments can be divided into three movements, the *statistical*, *structural* and *deterministic* movement. The *statistical* viewpoint was motivated along the idea that the complex structure is completely random, leading to a separation of the mean flow and the fluctuation components. Experiments in the past revealed specific coherent structures and correlation between spatial and temporal positions, which lead to the thought of turbulence not being completely random. This was the origin of the *structural* movement. The last *deterministic* movement started later as researchers were incapable of solving the Navier-Stokes equations directly. Only for a few decades one has been able to retrieve complex flow patterns of turbulence to study the behaviour under different conditions.

In general, three major methods are classified to solve fluid dynamics; Direct Numerical Simulation (DNS), Large Eddy Simulation (LES) and Reynolds Averaged Navier-Stokes (RANS). The DNS approach solves the Navier-Stokes equations directly and thus can give very accurate results, describing complex behaviour in all scales. The demands for a proper DNS simulation are very strict, requesting very fine grids and small time-steps in order to resolve small eddies. DNS simulations are therefore mostly used to create benchmark data for validating cheaper methods, as it is very expensive computationally. One step towards cheaper simulation is by solving the large eddies exactly and modelling smaller scales, hence the name being Large Eddy Simulation. The largest eddies are often generated or influenced by the domain, and thus these can be resolved by the mesh itself. Smaller eddies tend to become isotropic in nature and are therefore easier to model. Along the statistical movement, as said before, the idea is to model the total turbulent behaviour as a sum of an averaged flow and its fluctuations components. Reynolds Averaged Navier-Stokes does this, making it a very efficient method. Because all simulations in this work are performed in a U-RANS approach, the strategy of RANS is shortly explained.

Any fluid characteristic, such as velocity and pressure show fluctuations in turbulent flow. This makes it possible to separate the general physical property f into its mean behaviour $\overline{f}$ and the fluctuating components f' as

$$f(\mathbf{x},t) = \overline{f}(\mathbf{x},t) + f'(\mathbf{x},t)\,, \tag{6.1}$$

which is called the Reynolds decomposition. Different averaging procedures have been introduced in order to achieve this, namely

- Time averaging: $\overline{f}(\mathbf{x})_{\mathrm{T}} = \lim_{T\to\infty} \frac{1}{T}\int_t^{t+T} f(\mathbf{x},t)\,\mathrm{d}t$, suitable for statistically stationary turbulence. It is important to note T must be large enough compared to the time scale of the fluctuations f', hence T going to infinity;
- Spatial averaging: $\overline{f}(t)_{\Omega} = \lim_{|\omega|\to\infty} \frac{1}{|\Omega|}\int_{\Omega} f(\mathbf{x},t)\,\mathrm{d}\Omega$, suitable for homogeneous flows;
- Ensemble averaging: $\overline{f}(\mathbf{x},t)_{\mathrm{E}} = \lim_{N\to\infty} \frac{1}{N}\sum_{n=1}^{N} f_n(\mathbf{x},t)$, this is the most general method with N identical experiments.

As it is very convenient for derivation of the RANS equations, the time averaging procedure is utilized. For stationary or homogeneous turbulent flows, all three averaging procedures give the same results according to the *ergodic hypothesis*, and thus this does not confine the analysis to a specific case. In chapter 9 of [18], the Reynolds time-averaging procedure is completely detailed, leading to the following adapted conservation of mass (cf. eq. (3.7a))

$$\nabla\cdot\overline{\mathbf{v}} = 0 \tag{6.2}$$

and using $\nabla\cdot(\mathbf{v}\otimes\mathbf{v}) = \mathbf{v}(\nabla\cdot\mathbf{v}) + \mathbf{v}\cdot\nabla\mathbf{v}$ in order to return to a more general formulation, the adapted conservation of formulation (cf. eq. (3.7b))

$$\frac{\partial\overline{\mathbf{v}}}{\partial t} + \nabla\cdot(\overline{\mathbf{v}}\otimes\overline{\mathbf{v}}) - \nu\nabla^2\overline{\mathbf{v}} = -\frac{1}{\rho_0}\nabla\overline{p} + \mathbf{g}_f - \nabla\cdot(\overline{\mathbf{v}'}\otimes\overline{\mathbf{v}'})\,. \tag{6.3}$$

Two important terms are detailed, namely the the time derivative $\partial\overline{\mathbf{v}}/\partial t$ and the additional term $\nabla \cdot (\overline{\mathbf{v}'} \otimes \overline{\mathbf{v}'})$.

In essence the time averaging procedure removes time dependence, thus the velocity is independent of time in eq. (6.3) $\mathbf{v} = \mathbf{v}(\mathbf{x})$. In most modern practical formulations the $\partial\overline{\mathbf{v}}/\partial t$ is included to obtain a time varying mean flow result. McDonough ([31], subsection 2.1.2) describes the most-widely used arguments supporting retention of the time derivative. One speaks of Unsteady RANS (U-RANS); for small time-steps, unsteady behaviour of mean quantities can be captured. U-RANS methods are extensively being used for industrial applications as they are relatively cheap for computing turbulent fluid flows. However, lack of knowledge in the fluctuating components of velocity and pressure come at a cost.

There is only one additional term in (6.3) compared to the original equation, namely the divergence of $\tau_\mathrm{t} := -\overline{\mathbf{v}'} \otimes \overline{\mathbf{v}'}$. The term τ_t is named the Reynolds stress tensor, as it acts as a supplementary stress term. In total there are thus more unknown quantities compared to the instantaneous equations, but the same amount of equations, which means that there is a closure problem. In order to close the system and solve the system, one thus has to add more equations. Up till to today, the exact equations are not yet known, meaning they have to be approximated by turbulence models. These turbulence models define the Reynolds stress tensor to be a function of known variables. Several different theories exist, the most applied hypothesis is the Boussinesq approximation. Joseph Boussinesq suggested that turbulence of a flow is associated with a higher fluid viscosity; increasing turbulence causes fluid flows to become more irregular, having more eddies on different scales, and thus leading to easier dissipation of properties such as heat. Practically one can achieve such behaviour by increasing the kinematic viscosity ν. Such an increase is accounted for by a new term called the *eddy* or *turbulent viscosity* ν_t by approximating the Reynolds stress tensor as

$$\tau_\mathrm{t} = \nu_\mathrm{t} \left(\nabla\overline{\mathbf{v}} + \nabla\overline{\mathbf{v}}^T \right) - \frac{2}{3} k \mathbf{I} \,, \tag{6.4}$$

in which the term $k = \frac{1}{2}\overline{\mathbf{v}' \cdot \mathbf{v}'}$ is the turbulent kinetic energy. This again introduces the unknowns ν_t and k. The $k-\epsilon$ model by Jones and Launder [21] calculates these values by modelling the transport behaviour of the turbulent kinetic energy k and the energy dissipation rate ϵ according to

$$\nu_\mathrm{t} = C_\mu \frac{k^2}{\epsilon} \,, \tag{6.5}$$

where $C_\mu = 0.09$ is an experimentally obtained constant. The exact formulation of the system to be solved is given in [31]. The $k-\epsilon$ model is very popular for usage in high Reynolds flows, but performs less well near walls. Therefore usually it is accompanied by wall-functions. A small adaptation leads to the $k-\omega$ model (Wilcox [47]). In this model turbulent viscosity is expressed as

$$\nu_\mathrm{t} = C_\mu \frac{k}{\omega} \qquad \text{and} \quad \omega = \frac{\epsilon}{k} \,, \tag{6.6}$$

ω being the specific rate of dissipation. The advantage is that wall-functions are not needed, but sudden gradient changes are more troublesome for this model. The turbulence model put to use in this work is the $k-\omega$ SST model (Menter [30]), a mixture of the two above, designed to combine the advantageous aspects of the previous models.

6.2 Wall modelling

As mentioned above, turbulence models make use of wall-functions. Semantically, it is easy to deduce that these functions serve as a manner to approximate fluid behaviour near structure walls. Many wall-functions exist and different formulas are implemented amongst different software packages in order to describe to same physics. Therefore, considering that wall-functions for the turbulence kinetic energy k and the specific rate of dissipation ω are frequently being put to use in this work, a general description is given below, encapsulating the basic concepts of wall modelling.

The additional stress tensor τ_t induces higher diffusivity, which leads to very different behaviour compared with laminar flow. Figure 6.1 shows the difference in flow behaviour for laminar and turbulent in a single duct.

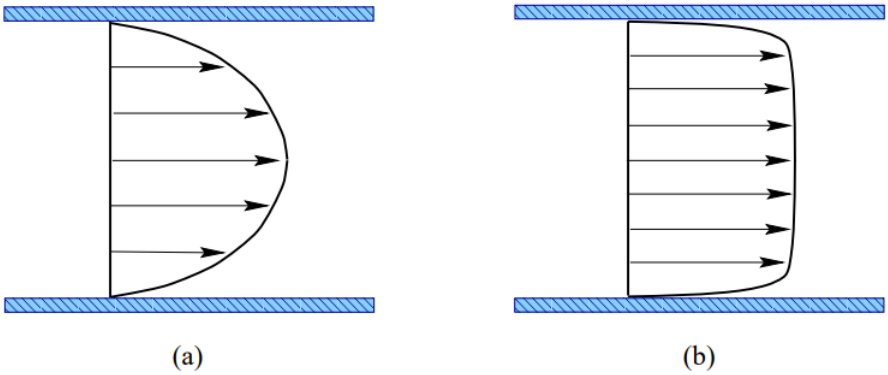


Figure 6.1: *Fluid velocity profiles in a duct for (a) laminar and (b) turbulent flow*

From this it can be seen that the the wall shear stress τ_w is much higher in the turbulent case, which is defined as

$$\tau_w = \mu \left.\frac{\partial v}{\partial y}\right|_{y=0}, \tag{6.7}$$

in which μ the dynamic viscosity, v is flow velocity along the boundary and y is the distance from the boundary. One can see that very close to the wall, the velocity profile is nearly linear with distance y, and is dominated by viscosity, hence this is being called the *viscous sublayer*. Towards the center of the duct, the velocity profile seems constant. From perturbation theory, by means of the method of asymptotic expansions, it is known that at least a third region having a different solution needs to be described. The specific physical behaviour of the fluid in each region near the wall is what is called *the law of the wall*, displayed in figure 6.2. Two dimensionless parameters are introduces to further clarify the picture, y^+ the dimensionless wall-distance and v^+ the dimensionless velocity,

defined as

$$y^+ := \frac{y v_\mathrm{t}}{\nu}, \qquad \text{and} \qquad v^+ := \frac{\overline{v}}{v_\mathrm{t}}, \tag{6.8}$$

in which $\overline{v}$ denotes the characteristic velocity in the outer region and v_t represents the friction velocity in the viscous sublayer. One of the most remarkable aspect from this law, is that purely on the basis of dimensional grounds it can be derived that these regions are connected according to a logarithmic function

$$v^+ = C_1 \log y^+ + C_2, \tag{6.9}$$

which is famously known as the '*log-law*'. The constants C_1 and C_2 are determined from experimental data. It is remarked that in the figure a fourth *buffer layer* is shown,

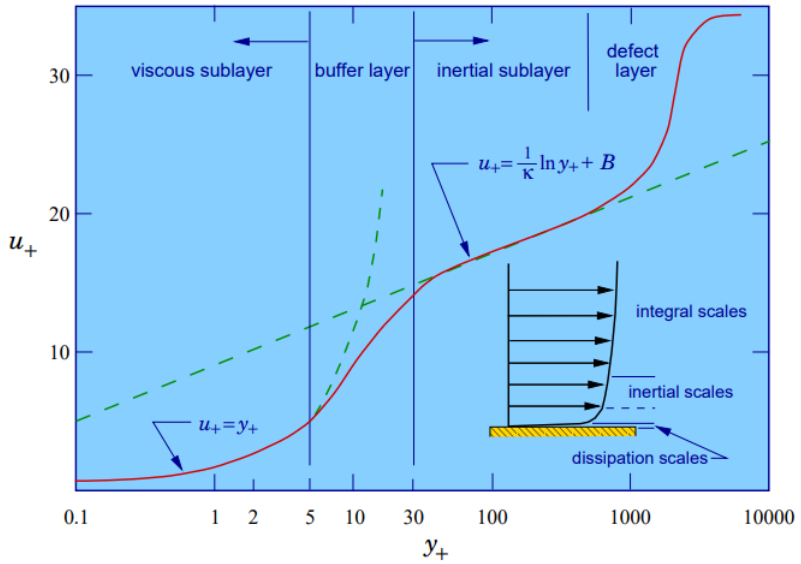


Figure 6.2: *Visualization of the law of the wall. A slightly different notation is used in this figure, in which* $u_+ = v^+$ *[31].*

connecting the viscous and the inertial sublayer. Having the knowledge on the behaviour of the fluid at hand, it seems redundant to employ fine gridding required to capture behaviour on the scales of the buffer layer and smaller. Wall-functions are empirical relations that satisfy the physical behaviour near the walls. To this end, the first cell center is placed in the inertial sublayer, where the log-law holds, setting the specific velocity as a boundary condition at the outer edge of the buffer layer (or further) for the empirical wall-functions.

It is important to assure that the mesh is not set up to have a first cell near the wall within the buffer layer. As the friction velocity is defined by $v_\mathrm{t} = \sqrt{\tau_\mathrm{w}/\rho}$ it is difficult in practice to achieve this exactly, due to τ_w, v_t and thus y^+ being values that need to be calculated on the fly. Consequently, usually a first guess of the off-the-wall distance of the nearest cell is made, that is at $y^+ \approx 60$ at least. During simulation these assumptions are checked.

The exact empirical wall-functions are simplified model equations (for example from

the $k-\omega$ SST turbulence model) in the viscous and inertial sublayer (logarithmic region). This way a solution ω^+_{vis} and ω^+_{log} can be obtained for the dimensionless specific rate of dissipation. The behaviour in the buffer layer is then approximated for example by taking the euclidean norm

$$\omega^+ = \sqrt{(\omega^+_{\text{vis}})^2 + (\omega^+_{\text{log}})^2} \tag{6.10}$$

In [28], a thorough description is given of how wall-functions are implemented in OpenFOAM.

6.3 Pressure Fluctuation Model

Due to the efficient manner in which U-RANS approaches can describe average behaviour of fluid flow, U-RANS models are widely being put to use in the industries. The U-RANS model sacrifices the knowledge of fluctuation flow variables such as the turbulent velocities $\mathbf{v}'$ and the fluctuating pressures p'. In order to describe turbulence-induced vibrations of nuclear reactor components, Kottapalli [27] integrated a Pressure Fluctuation Model (PFM) in NRG-FSIFOAM during the task of his Master thesis project. Figure 6.3 displays the workings of the PFM model in a flowchart, encircled by the dashed line. It can be seen that the PFM serves as an additional algorithm to the U-RANS solver in the complete FSI software package.

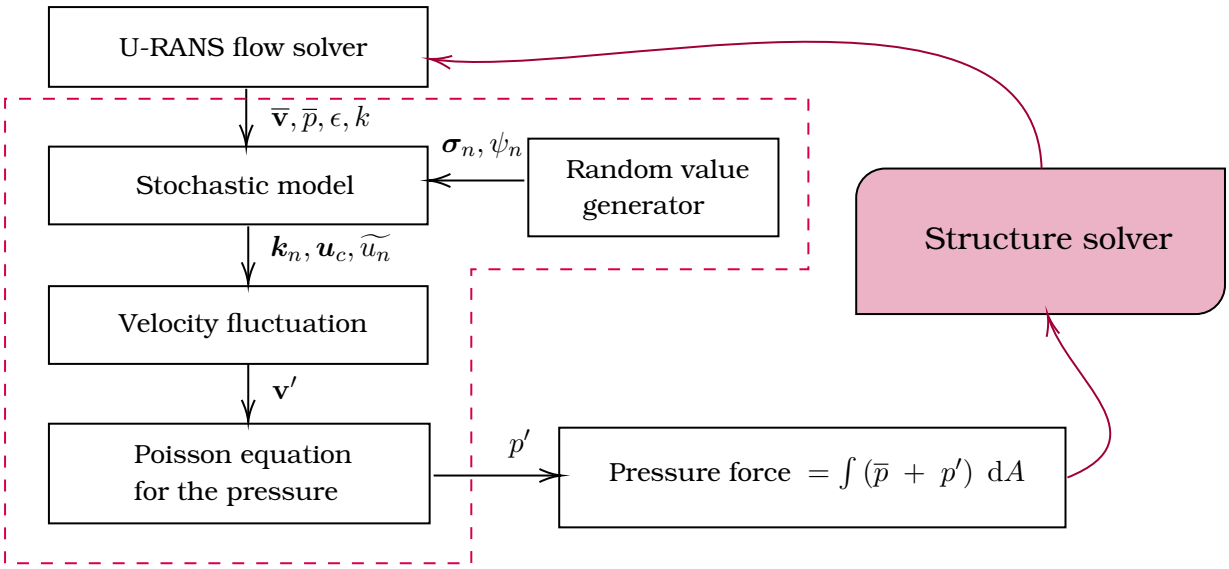


Figure 6.3: *Diagram of U-RANS with the Pressure Fluctuation Model in a general FSI software package.*

Following the flowchart, the fluctuating velocity vector

$$\mathbf{v}'(\mathbf{x}, t) = \sum_{n=1}^{N} \widetilde{u_n} \cos\left(\boldsymbol{k}_n \cdot (\mathbf{x} - t\boldsymbol{u}_c) + \psi_n + \omega_n t\right) \boldsymbol{\sigma}_n \,, \tag{6.11}$$

is calculated on the basis of random variables and the output of the U-RANS solver. In

equation (6.11), $\widetilde{u_n}$ is the amplitude, ψ_n is the phase and $\boldsymbol{\sigma}_n$ is the unit direction vector of the n^{th} mode associated with the wave vector $\boldsymbol{k}_n$, ω_n is the characteristic angular frequency and $\boldsymbol{u}_c$ is the convection velocity respectively. As can be seen, the range of eddies in the turbulent field, which is continuous, is discretized into a chosen amount of modes. More specifically a Fourier discretization is chosen, as this fits well to the naturally periodic behaviour of eddies.

On the basis of random number generators, $\boldsymbol{\sigma}_n$ (and thus $\boldsymbol{k}_n$) and the phase ψ_n are calculated. Once these have been obtained, the energy spectrum with respect to all the modes n is determined. From the amount energy the different eddies carry, the magnitude of fluctuation $\widetilde{u_n}$ can readily be obtained. These elements form the building blocks of the fluctuations velocity $\mathbf{v}'$. At last, using the Poisson equation for pressure, this velocity is translated to a fluctuating pressure p' that is added to the mean pressure $\overline{p}$ in order to be imposed on the in the interface of the structure. A more thorough description of U-RANS coupled with PFM can be found in chapter 5 in Kottapalli's thesis [27].

6.4 PIMPLE algorithm

The governing incompressible Navier-Stokes equations (3.8a, conservation of mass) and (3.8b, conservation of momentum) have four unknowns, namely the velocity $\mathbf{v}$ in all three directions and the pressure p. As the conservation of mass does not contain the pressure, a trick must be used to solve the momentum equation. This is also known as the *pressure-momentum coupling problem*.

By applying the divergence operator and the conservation of mass in a witty manner, several terms can be scrapped, giving the Poisson equation for the pressure p (this same equation is used in the PFM algorithm). Now having these new equations, the requested variables can be found by sequentially solving them. Strategically, pressure-momentum coupling algorithms have been designed to assure conservation of mass and momentum in a stable and efficient manner. Under these the `SIMPLE`, `PISO` and `PIMPLE` algorithms are the most well known:

- `SIMPLE` [35] — Semi-Implicit-Method-Of-Pressure-Linked-Equations, is an algorithm being put to use for steady-state simulations. It is slower in general, as each value $\mathbf{v}$ or p is under-relaxated before being used as input. A major advantage is that it can run simulations in a stable manner for Courant numbers larger than one;

- `PISO` [20] — Pressure-Implicit-Split-Operator, is a faster algorithm used for transient application. The velocity is calculated once by means of the momentum equations and under-relaxation is not applied. Its biggest disadvantage it that it can only run stable for low Courant numbers, meaning that restrictions have to be put on meshes and/or time-steps in order to have stable simulations.

- `PIMPLE` — merged `SIMPLE-PISO`, tries to combine best of both worlds. This allows for more stable transient simulations with a larger time-step.

The general mechanism goes as follows. Firstly the algorithms semi-discretize the momentum equation (ALE formulation)

$$\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{c} \cdot \nabla)\,\mathbf{v} - \nu \nabla^2 \mathbf{v} = -\frac{1}{\rho_0}\nabla p + \mathbf{g}_f\,, \tag{6.12}$$

in time, resulting in a matrix equation. The trick of the PIMPLE algorithm is splitting the matrix up to its diagonal and non-diagonal parts and extracting $\mathbf{v}$ (semi-discrete solution) to obtain (ρ_0 and $\mathbf{g}_f$ are left out for simplicity)

$$\begin{aligned} [\mathbf{v}\text{Eqn}] &= -\nabla p\,, \\ A\mathbf{v} &= -\nabla p + \mathbf{H}(\mathbf{v})\,, \\ \mathbf{v} &= \frac{\mathbf{H}}{A} - \frac{1}{A}\nabla p\,, \end{aligned} \tag{6.13}$$

which can be done since A is diagonal and contains non-zero elements. The last of these equations is called 'the momentum-corrector equation'. Next, the obtained values for the velocity can then be filled in the previously obtained Poisson equation

$$\nabla \cdot \frac{1}{A}\nabla p = \nabla \cdot \left(\frac{\mathbf{H}}{A}\right)\,, \tag{6.14}$$

which is used as corrector for the pressure. These new values for p at its turn are used to correct $\mathbf{v}$, looping until a certain tolerance is obtained. The overall structure of the `PIMPLE` algorithm at each particular time-step is given in algorithm 6.1. Two major loops define the `PIMPLE` algorithm; the outer loop, using *n_oCorr* iterations at maximum, solves the actual momentum equation and has the second loop in each iteration, the second inner loop uses the corrector equations for the momentum and the pressure. Similar as for the `SIMPLE` algorithm, the velocity $\mathbf{v}$ and pressure p are under-relaxated.

ALGORITHM 6.1: PIMPLE algorithm

for *oCorr* $= 0$; *oCorr* $<$ *n_oCorr* **do**
  Solve momentum: $[\mathbf{U}\text{Eqn}] = -\nabla p$
  relax $\mathbf{U}$
  for *corr* $= 0$; *corr* $<$ *nCorr* **do**
    Evaluate: $\mathbf{H}(\mathbf{U}), A$
    Correct pressure: $\nabla \cdot \frac{1}{A}\nabla p = \nabla \cdot \left(\frac{\mathbf{H}}{A}\right)$
    relax p
    Correct velocity: $\mathbf{U} = \frac{\mathbf{H}}{A} - \frac{1}{A}\nabla p$
  end
  Correct turbulence parameters
  relax turbulence parameters
  Calculate RMS residual r_{RMS} momentum equation
  if $r_{RMS} < tol_{PIMPLE}$ **then**
    break
  end
end

Chapter 7

Comparison of impulse-forced and PFM-forced FSI simulations

This chapter assesses the developed FSI solver for its accuracy by examining flow-induced vibrations (FIV) of different rod configurations in turbulent axial flow. A single cylindrical bare rod is considered first, of which the geometry is provided by the study of De Santis et al. [12, 13]. The dynamics of the single bare rod are investigated by subjecting it through working fluid forces of water and liquid metal for different mesh resolutions via an initial impulsive force and the pressure fluctuation model (PFM, see section 6.3). In a similar fashion as in the paper of De Santis et al., accuracy of the different numerical solvers is investigated through extraction of vibration frequencies and damping ratios. Having this knowledge, analogously to the single bare rod case, a helically wire-wrapped rod submerged in turbulent axial flow of water and liquid metal is simulated. The geometrical set-up is also extracted from the study of De Santis et al. By considering these case studies, both the accuracy of the NRG-FSIFOAM library, as the potential of the PFM model to replicate realistic turbulent behaviour in nuclear applications is examined.

7.1 Flow-Induced Vibrations of a bare rod

In this test case a single bare rod is submerged by water or lead-bismuth in axial flow to examine the vibrational movement. These vibrations are created either by an impulsive force at the center of the rod or either the PFM model. As is mentioned in chapter 2, many different reactors types are being developed using liquid metals as a moderator. Consequently, the density ratio of the solid and fluid ρ_s/ρ_f is small, and often lower than one. Accordingly the added mass, being the mass of fluid which is accelerated by the structure, is very large. Strong-coupling algorithms (see chapter 5) can thus not be avoided. The bare rod case is therefore a useful test case to examine FSI-coupling software. In the following parts, the complete description of the geometry, the mesh models, the numerical considerations for running the fluid/solid solver and the results with a discussion are given of solid-only and FSI simulations.

7.1.1 Physical and Geometrical Properties

The considered solid is a uniform bare rod made out of stainless steel. It has the following properties:

- $L = 700$ mm (Length);
- $D_s = 6.55$ mm (External diameter);
- $t = 0.51$ mm (Thickness);
- $\rho = 7500 \ \text{kg} \cdot \text{m}^{-3}$ (Density);
- $E = 200$ GPa (Young's modulus);
- $\nu = 0.3$ (Poisson ratio).

The geometrical dimensions of the components are the same as those of the fuel rods used in the test facility built at SCK-CEN to study flow-induced vibration of a cluster of seven rods in liquid metal, with the diameter of the rods the same as that of the MYRHHA reactor and the length half of the actual value.

The fluid domain is a cylindrical tube in which the rod is positioned at the center. Geometrically it is described as:

- $L = 700$ mm (Length);
- $D_f = 42.6$ mm (External diameter).

Both water (H_20) and liquid metal lead-bismuth eutectic (PbBi) are considered to flow through the axial direction. Water has a density $\rho_{H_20} = 1000 \ \text{kg} \cdot \text{m}^{-3}$ and dynamic viscosity $\mu_{H_20} = 9.23 \cdot 10^{-4}$ Pa $\cdot$ s. For the second fluid these values become $\rho_{\text{PbBi}} = 10364$ kg $\cdot$ m^{-3} and $\mu_{\text{PbBi}} = 1.938 \cdot 10^{-3}$ Pa $\cdot$ s. In all of the cases, an inflow velocity of 2 m$\cdot$s^{-1} is considered. These intrinsic variables induce a Reynolds number of $Re_{H_20} = 92308$ and $Re_{\text{PbBi}} = 455631$, based on the external diameter of the tube. Although actual applications require higher fluid velocities, the current value induces sufficient turbulence to study the frequencies of the bare rod.

7.1.2 Meshes and numerical considerations

Four different solid meshes are used to investigate accuracy of the NRG-FSIFOAM library. Copied over from [13], the coarsest mesh is structured with 3 elements along the radial direction and 36 along the angular direction, giving 108 quadrilateral cells. This mesh is then extruded along the axial direction with 200 divisions. The cross-sectional mesh and part of the axial grid are displayed in figure 7.1. In total the structured grid consists of 21600 elements. Apart from this mesh, two refinements (Mesh2 and Mesh3) and a conformal mesh for FSI (see further), named Mesh4, are used. The exact mesh specifications are in table 7.1.

	radial	angular	axial	elements
Coarse	3	36	200	21600
Mesh2	5	48	400	96000
Mesh3	8	60	600	288000
Mesh4	3	80	400	96000

Table 7.1: *Solid meshes to approximate the steel rod.*

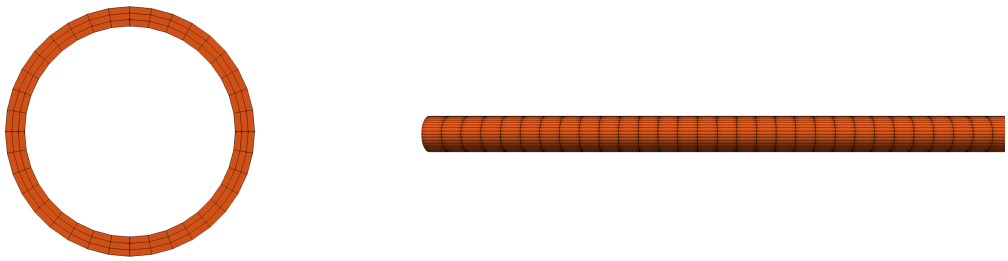


Figure 7.1: *Cross-section (left) and first 30 elements out of 200 in the axial direction (right) of the coarsest rod grid. In the radial direction there are 3 divisions and in the angular 36 divisions.*

To study the frequency, the cylindrical rod ideally is perturbed by an impulsive force applied at the center of the rod. However to make it computationally easier, this impulsive force is approximated by the following function,

$$\delta_F = \begin{cases} \alpha \cdot \mathrm{e}^{-t^2/2\sigma^2}, & \text{if } 0 \leq t \leq 2.5 \cdot 10^{-3}, \\ 0, & \text{otherwise,} \end{cases} \tag{7.1}$$

with $\sigma = 0.0007$ fixed, α chosen after several tests to obtain a displacement that is similar in magnitude to the STAR-CCM+ results of De Santis et al. The impulsive force is applied in positive x- and y-direction. Practically, a patch from 310mm up to 410mm is distinguished to evenly apply δ_F. An example of a rod mesh for solving the numerical problem is displayed in figure 7.2. There are four physical surfaces defined on the rod, namely, the inlet boundary (not visible), the free parts (yellow), the forced patch (cyan) and the outlet boundary (red). These are defined for application of different boundary conditions on different parts of the domain.

A linear elastic solver, implemented in the DEAL-II C++ library, is used to solve the structural dynamics together with the theta-method as a time-marching scheme. As a second-order implicit scheme has been used by De Santis et al., $\theta = 0.5$ (Crank-Nicholson) was preferred. However, Crank-Nicholson is known to show unstable behaviour, hence, unless mentioned, $\theta = 0.6$ is being used.

Two essentially different numerical fluid domains are considered, based on the off-the-wall height of the first cell. For both water and PbBi flow a $y^+ \approx 1$ wall-resolved mesh and a $y^+ \approx 60$ wall-function mesh are employed, all fully structured meshes of hexahedral elements. The meshes being applied for water flow are visualized in figure 7.3 and the complete cylindrical fluid domain is visualized in figure 7.4. It is worthwhile mentioning

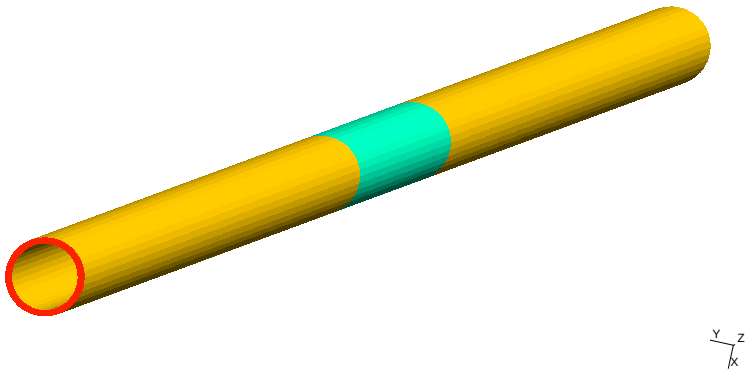


Figure 7.2: *A rod mesh considered for simulations. Different colors denote different physical surfaces; the cyan color denotes the patch to which the function δ_F (7.1) is evenly applied, the outlet boundary is red, the free parts are yellow and the inlet boundary is not visible.*

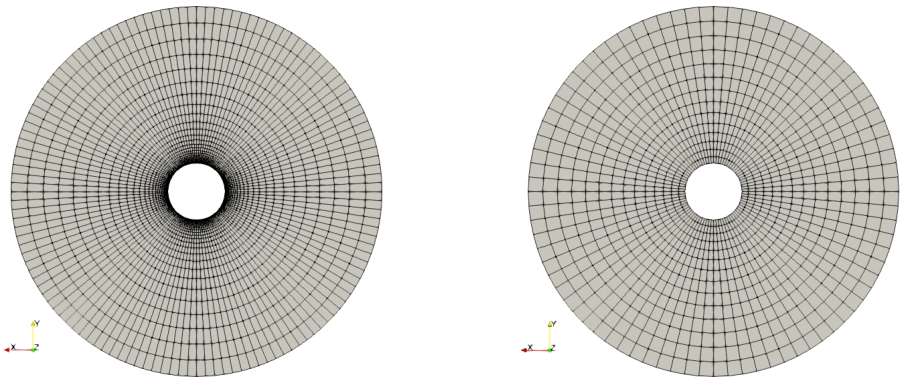


Figure 7.3: *Cross-section of the fluid meshes considered for the FSI simulations of a bare rod in axial water flow. Off-the-wall heights of the first cell are specifically chosen s.t. a $y^+ = 1$ mesh (left) and a $y^+ = 60$ mesh (right) are generated.*

that different fluid meshes are used for water and PbBi flow. This is due to the fact that both liquids give different Reynolds numbers for the same inlet velocity. The specific mesh details are given below in table 7.2. Mesh4 in table 7.1 is indeed conformal to the $y^+ \approx 60$ mesh for water flow, as both have 80 angular elements and 400 axial elements. Meshes need to be sufficiently resolved in the normal direction towards the fluid-structure

	mesh type	radial	angular	axial	elements
H_2O	$y^+ \approx 1$	32	112	400	1433600
	$y^+ \approx 60$	14	80	400	448000
PbBi	$y^+ \approx 1$	32	120	400	1536000
	$y^+ \approx 60$	22	80	400	704000

Table 7.2: *Numerical fluid domains (cylindrical tubes) considered for the FSI simulations of a bare rod submerged by water and PbBi.*

interface to prevent under-prediction of wall-shear stresses. Hence, wall-function models are considered in the $y^+ \approx 60$ mesh. Wall-function meshes are usually opted for to

reduce computational costs. As mentioned, the Reynolds numbers for all cases are large enough to have turbulence. The U-RANS approach with the k-ω SST turbulence model (section 6.1) is adopted in order to solve for the unsteady turbulent flow. All governing equations, i.e. the continuity, momentum and turbulence equations are discretized in space with a second-order upwind scheme and discretized in time with the second-order BDF-2 scheme. All fluid calculations are performed with the OpenFOAM library. For the sake of reproduction, all exact spatial discretization methods are summarized in table 7.3.

	Scheme
ddtSchemes	BDF-2
gradSchemes	Gauss linear
divSchemes	Gauss linearUpwind Gauss linear
laplacianSchemes	Gauss linear corrected
interpolationSchemes	linear
snGradSchemes	corrected

Table 7.3: *Finite volume settings for the fluid solver OpenFOAM in the single bare rod case.*

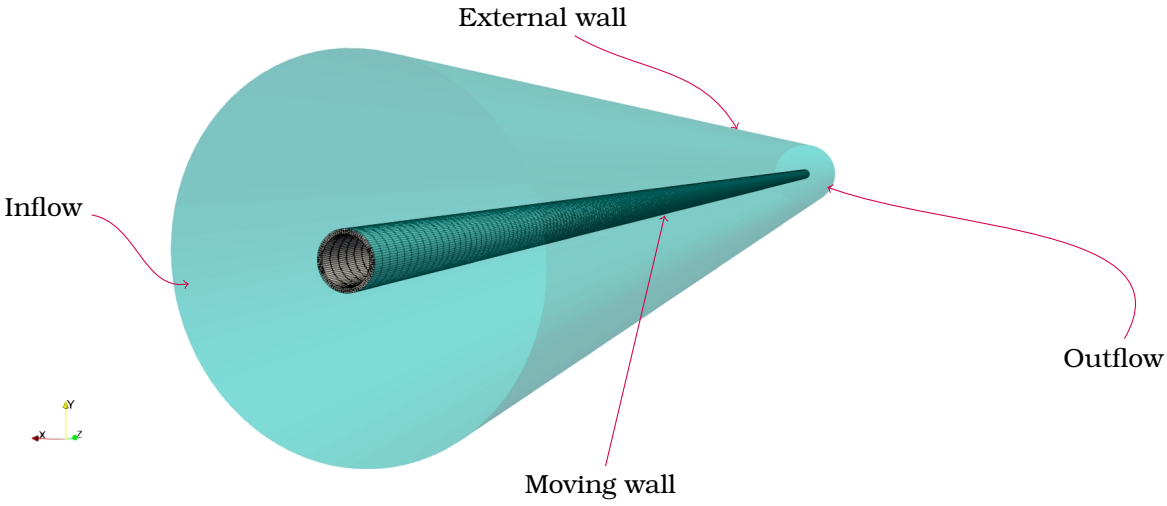


Figure 7.4: *The cylindrical domain of the single bare rod case and an example solid mesh.*

Mesh movement of the fluid domain is solved in the ALE formulation as described in chapter 4 by means of thin plate spline RBFs. The preCICE coupling library connects both the DEAL-II and the OpenFOAM solvers by means of the IQN-ILS algorithm (section 5.2) using the second-order accurate extrapolation (5.9) as an initial guess. At maximum 60 coupling iterations are used together with reuse of 8 earlier time-steps in the calculation of the approximation of the inverse Jacobian matrix.

Variable	$t = 0$	Inflow	Outflow	External wall	Moving wall
$\mathbf{v}$	$\mathbf{0}$	$[0, 0, 2]$	zero gradient	slip	structural velocity
p	0	zero gradient	0	zero gradient	zero gradient
k	0.375	0.375	zero gradient	slip	kqRWallFunction
ω	559	559, $l_t = 2$ mm	zero gradient	slip	omegaWallFunction

Table 7.4: *Boundary and initial condition settings for the OpenFOAM solver in case a $y^+ \approx 60$ wall-function mesh is employed. The variable l_t represents the turbulent length scale which is being mentioned as the 'turbulentMixingLenthFrequencyInlet'. In case a $y^+ \approx 1$ wall-resolved mesh simulation is set-up, only the value k at the moving wall changes to a uniform 1e-12 value. The units of all variables are respectively $m \cdot s^{-1}$, $m^2 \cdot s^{-1}$, $m^2 \cdot s^{-1}$ and s^{-1} for $\mathbf{v}$,p, k and ω.*

7.1.3 Boundary and initial conditions

The inlet boundary of the solid mesh is assumed to be clamped, this means that the rod is fixed in all directions and no movement is allowed. At the outlet, restricted movement of the rod is allowed in the axial (z) direction, which means the right end of the rod is pinned. In case an impulsive force is applied to the center of the rod, these forces are translated as boundary conditions to a patch in the middle of the rod as can be seen in figure 7.2. In summary this means for the rod displacement $\mathbf{u} := [u_x, u_y, u_z]$ that

- Inlet: $\mathbf{u}(x, y, 0, t) = 0$;
- Outlet: $u_x(x, y, L, t) = u_z(x, y, L, t) = 0$,

$\forall x, y$ and $t > 0$. The solid rod starts in the non-deformed positioning, meaning that $\mathbf{u}(x, y, z, 0) = 0$ for all x, y and z.

In table 7.4 all essential boundary and initial conditions for the fluid solver in order to perform the bare rod FSI simulations for wall-function meshes ($y^+ \approx 60$) are described. The only difference in the boundary conditions between the $y^+ \approx 1$ and $y^+ \approx 60$ meshes lies in the use of wall-functions. At the moving wall only the value of the turbulent kinetic energy k changes to a uniform value of 1e-12. The specific turbulent dissipation rate ω still uses wall-function modelling, since ω increases exponentially towards the wall. This way OpenFOAM allows users to model ω irrespective of the mesh resolution.

The inflow of the liquid is assumed to be fully turbulent in all different cases, having a turbulent intensity TI = 25% and the turbulent length scale taken at $l_t = 2$ mm. By filling in these values in the following formulas $k = \frac{3}{2}(\bar{\mathbf{v}} \cdot \mathrm{TI})^2$ and $\omega = \frac{k^{1/2}}{C_\mu^{1/4} l_t}$ the values 0.375 and 559 in the table can be found, with $\bar{\mathbf{v}} = 2$ the mean flow velocity and $C_\mu = 0.09$ a turbulence model constant. Many different variants, mostly scaled by different constants, exist for the calculations of these values. It depends on the considered software which ones to apply; in this case the used formulas are appropriate for the OpenFOAM solver.

The slip condition tries to resemble a rod being in an infinitely large fluid compound. In figure 7.3 one can see application of the slip wall is reflected in the amount elements near the outer wall. Using slip conditions, computationally cheaper problems are solved, due to the absence of wall resolved parts in the mesh. More realistic boundary conditions

can be used, however a trade-off has to be made between computational expenses and accuracy.

7.1.4 Solid-only results and discussion

In the following, all results from the solid domain are extracted for a center point on the outer hull of the mesh; more specifically the point with coordinates $x = -0.003275, y = 0$ and $z = 0.35$. Since all points on the center of the mesh undergo a similar displacement, one can assume this is an appropriate measure for the actual behaviour of the rod.

Firstly, a coarse mesh simulation of 200 time-steps with $\Delta t = 2.5 \cdot 10^{-4}$ s is performed, resulting in nearly 3 periods of oscillation in displacement. The time-step Δt is chosen so that the impulse function δ_F (7.1) is taken into account for a considerable amount of time-steps, in this case 10. Next, solid-only simulations are again performed to take a look at the influence of the mesh resolution in DEAL-II, using a time-step of $\Delta t = 10^{-4}$ s for 400 iterations. In all performed simulations Crank-Nicholson ($\theta = 0.5$) is used.

By means of a sine-fitting method in MATLAB, the first 3 natural frequencies are calculated and compared to analytical frequencies of the center of the rod. This fitting method calculates the parameters a, ω and ϕ via non-linear least-squares s.t. $a \sin(\omega \cdot t + \phi)$ approximates the data well. At last, the natural frequency $f = 2\pi/\omega$ is extracted. This is repeated three times by each time subtracting the previous fitting data from the data. Hence several natural frequencies are obtained.

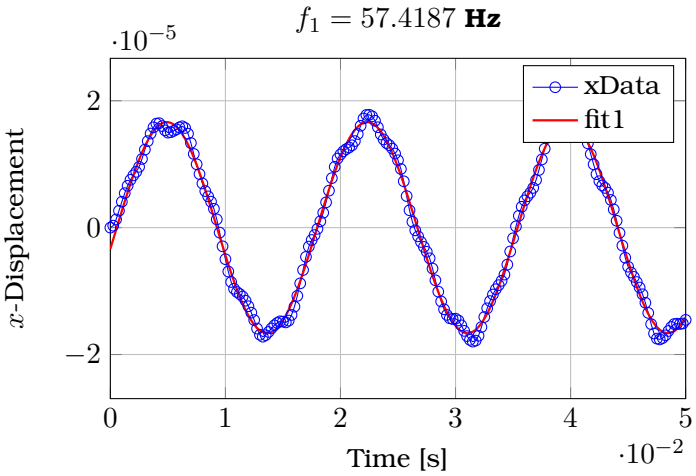


Figure 7.5: *x-displacement of a center point of the rod and an approximation of the 1st natural frequency.*

Remark 5. *The explained method for calculating frequencies is preferred over a basic FFT-based method, because for short data capture FFT methods can produce very inaccurate results.*

In figures 7.5 and 7.6 the MATLAB sine-fit results of the x-displacement are displayed together with an approximation of the frequency for the first coarse mesh simulation. One can notice that the 3rd natural frequency is obtained before the 2nd. This can be explained by looking at figure 7.5; the 3rd mode is clearly more dominantly present.

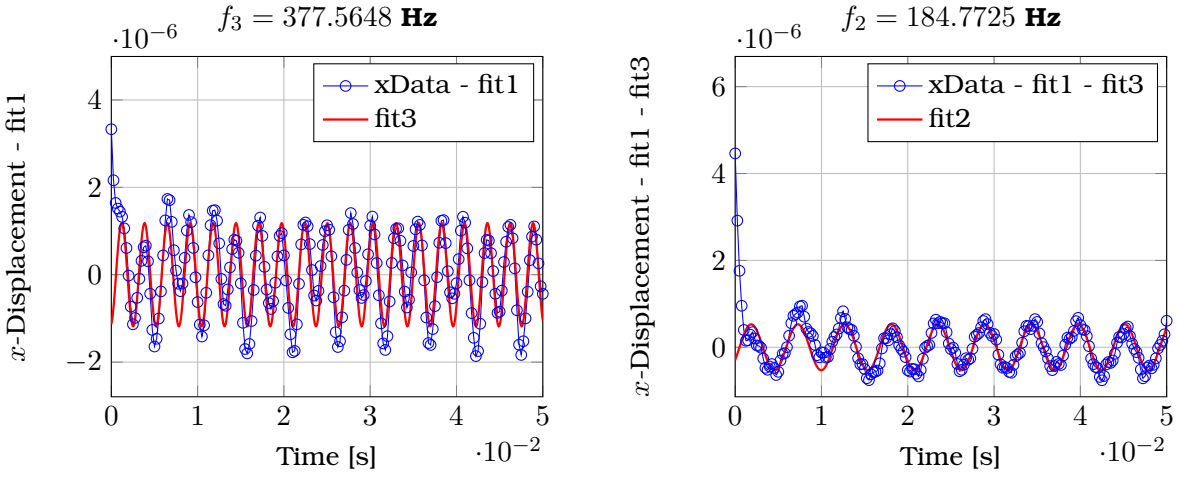


Figure 7.6: *x-displacement minus the first sine-fit and x-displacement minus the first and second sine-fit values in* MATLAB *of a center point of the rod. Both the third and second natural frequencies are approximated.*

In the first two rows of table 7.5 the first three eigen-frequencies are compared against the approximated frequencies from the simulation. De Santis et al. obtained these values by means of the Lanczos algorithm [37] implemented in the open-source finite element solver `Code_Aster`, using the coarse mesh together with the same boundary conditions. In general the frequencies are a bit under-predicted. The results however describe the problem well enough to be used for FSI simulation.

	f_1 [Hz]	f_2 [Hz]	f_3 [Hz]
Lanczos	60	195	406
Coarse ($\Delta t = 2.5 \cdot 10^{-4}$ s)	57.4187	184.7725	377.5648
Coarse	57.3974	184.5471	386.5273
Mesh2	55.7090	179.7732	374.7398
Mesh3	55.3974	179.1088	374.9095
Mesh4	55.7570	179.7733	375.0329

Table 7.5: *Comparison of the first three natural frequencies of the different rod meshes (table 7.1) for both the simulation of $\Delta t = 2.5 \cdot 10^{-4}$ s and $\Delta t = 10^{-4}$ s.*

Next, the results of the mesh sensitivity study are investigated. The y-displacements are given in figure 7.7 (the x-displacements are very similar and thus not shown). Table 7.5 summarizes the first three natural frequencies of the different rod grids used. It can be noticed that the natural frequencies decrease compared to the results from `Code_Aster`, which can be explained due to the Lanczos algorithm only being tested on the coarse grid.

7.1.5 FSI results and discussion

Next, the displacements from the same central point as before ($x = -0.003275, y = 0$, $z = 0.35$) are extracted, to investigate the influence of a streaming fluid (water, PbBi) on the natural frequencies of the bare rod. The damping factors are calculated as well accompanied with the frequencies. A different fitting method compared with the solid solver is

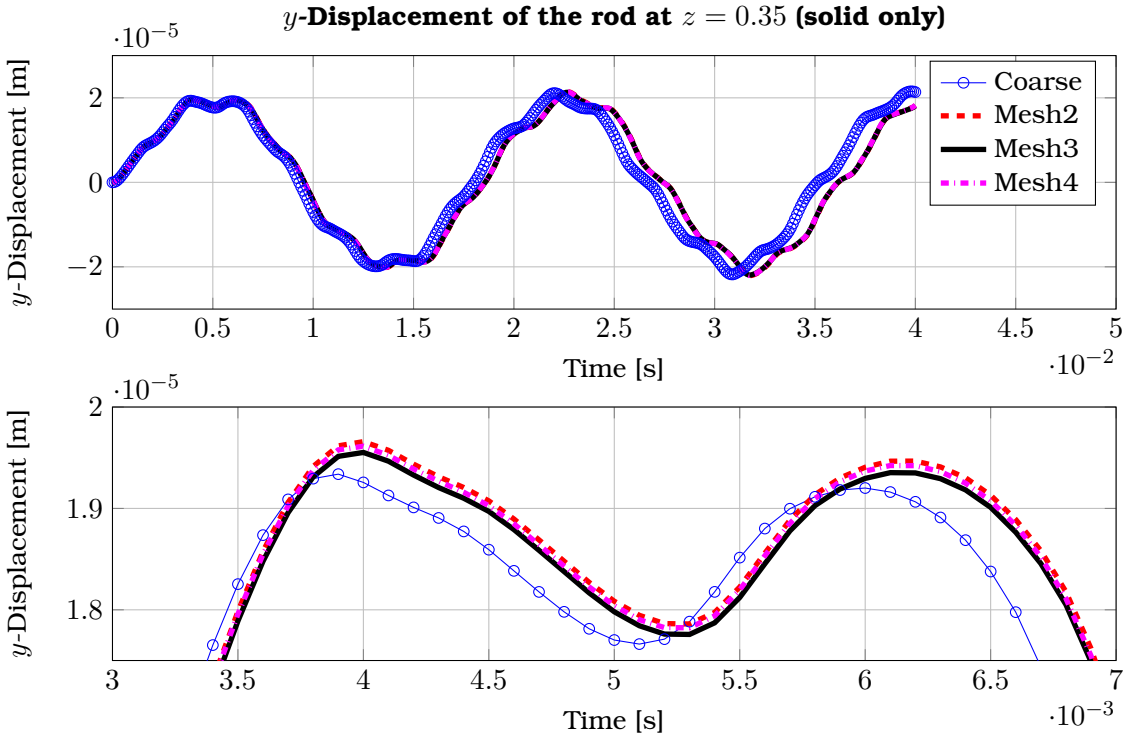


Figure 7.7: *y-displacements of a center point of the rod for all used solid meshes given in table 7.1.*

used to obtain the modal characteristics. The extracted displacements are approximated by the following function

$$\sum_{i=1}^{N_m} a_i \mathrm{e}^{-\chi_i t} \sin(\omega_i \cdot t + \phi_i)\,, \tag{7.2}$$

in which N_m represents the amount of modes considered. Via a non-linear least-squares fitting, the parameters a_i, χ_i, ω_i and ϕ_i are approximated. At last, it is possible to obtain the natural frequency (f_i) and the modal damping ratio (ξ_i) according to the equations

$$\chi_i = 2\pi\xi_i f_i \qquad \text{and} \qquad \omega_i = 2\pi\sqrt{1-\xi_i^2}\, f_i\,. \tag{7.3}$$

The exponential term $\mathrm{e}^{-\chi_i t}$ is left out when radial displacement history of PFM results are being processed. Essentially this results in a similar fitting approach as is in the solid-only case. The PFM algorithm will excite the natural frequency of the rod, consequently the displacement is not being gradually damped. Therefore only the natural frequencies are extracted in these cases.

In all simulations, both with an impulsive force as with the PFM algorithm (section 6.3), a time-step of $\Delta t = 10^{-4}$ s is used. The amount of time-steps of the simulations is different for each case; the FSI simulations are memory- and time-consuming, consequently if enough data was obtained, the simulations were stopped. Furthermore Crank-Nicholson ($\theta = 0.5$) is only used for the first case, namely the $y^+ \approx 60$ fluid mesh of water flow.

The results are displayed according to two categories, first water flow is shown, followed by the results obtained from the lead-bismuth eutectic (PbBi). These on their turn

are subdivided by first showing wall-function mesh ($y^+ \approx 60$) results and then the wall-resolved mesh ($y^+ \approx 1$) results. PFM simulation details are given for the wall-function mesh cases.

Remark 6. *It is important to mention that ahead of performing all the FSI simulations, a fluid-only simulation was done in order to have a fully developed fluid flow through the domain. Comparison is made with a operating nuclear reactor in which the fluid is continually flowing at a steady pace. In Appendix A, a short description of the fluid-only simulation of water flow using the wall-function mesh ($y^+ \approx 60$) is given to show that fully developed flow is obtained.*

In figure 7.8 the extracted x- and y-displacement over time concerning the coarse mesh (table 7.1) are shown and compared with results obtained by De Santis et al. The impulsive force pushes the rod to its maximum amplitude. From this point on the rod moves with a constant frequency and diminishing amplitude as a consequence of the dissipative effects of the fluid submerging the rod. Moreover, it can be seen that the x- and y-displacement are equal because of the symmetrical properties of the case.

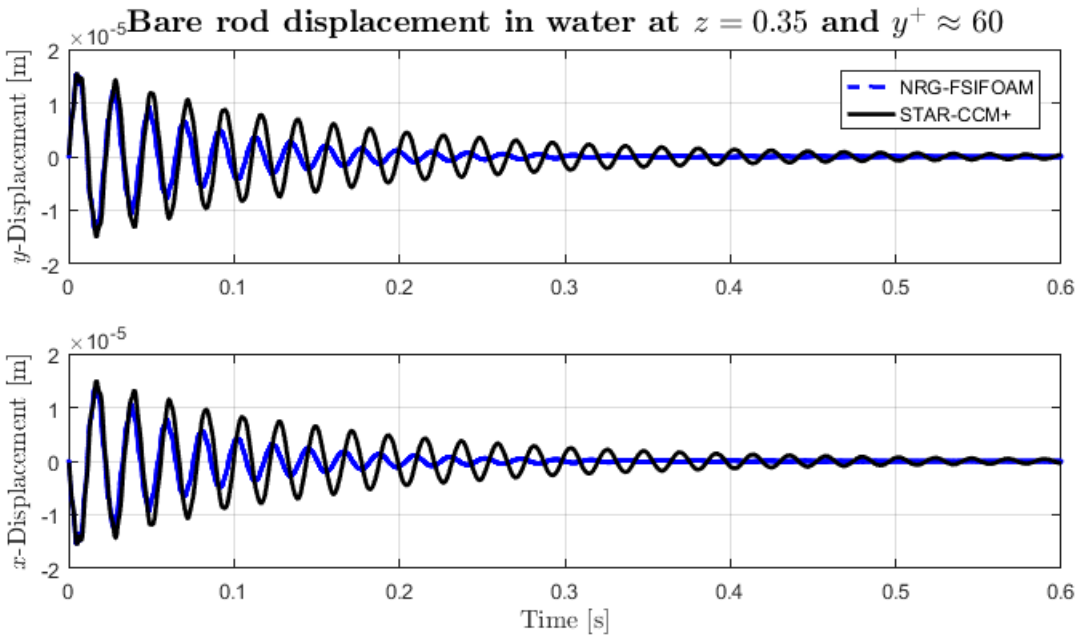


Figure 7.8: *Displacement history of the center point of a bare rod in the y-direction (top) and x-direction (bottom) in water flow, generated by means of a wall-function ($y^+ \approx 60$) fluid mesh. The results generated with NRG-FSIFOAM are compared to results generated by STAR-CCM+.*

Remark 7. *In order to make it easier to compare with the results obtained from STAR-CCM+, the x-displacement is shown in the $-x$-direction, although an impulsive force pushes the rod in the positive direction at $t = 0$.*

Thereafter, similar as for solid-only simulations, a mesh sensitivity study is performed for the current case. Figure 7.9 represents the displacement in 400 time-steps for all

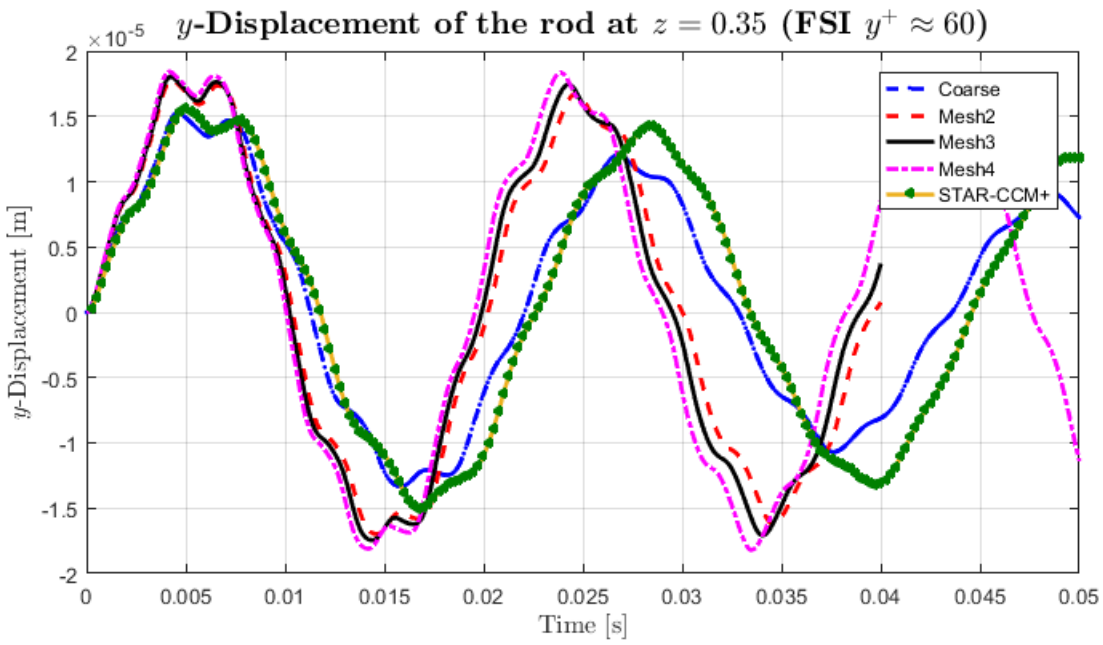


Figure 7.9: *Comparison of all meshes in table 7.1 of the y-displacement history of the center point of a bare rod in water flow, generated by means of a wall-function ($y^+ \approx 60$) fluid mesh.*

meshes in table 7.1 together with STAR-CCM+ results. From both the shown figures it can already be observed that the frequency is higher then the STAR-CCM+ results. Moreover, opposite to the solid-only frequencies where the frequency decreased with finer solid meshes, now the frequencies increase. The difference in the damping of both the coarse mesh simulations executed with NRG-FSIFOAM and STAR-CCM+ stands out the most however. Based on the first two periods of oscillation in figure 7.9, this large difference in damping seems to level out when meshes are refined.

A possible explanation is due to usage of the nearest-neighbour interpolation to map the obtained stresses onto the structure elements. The rod only having 200 elements in the axial direction, compared with the fluid domain having 400 elements, will often be mapped a much higher stress value onto the structure elements than should be the case for this part of the rod. Hence, the rod is damped out more quickly. Increasing the amount of axial elements to 400 maps the stresses more accurately for each part of the rod, which decreases the damping ratio significantly. In order to see if this is truly the cause, one can try to use more accurate mapping methods, such as by means of radial basis functions described in chapter 4.

These observations are confirmed in table 7.6 comparing the vibration frequencies and the damping ratios of the first three modes for all considered meshes. The frequencies of a PFM simulation are given as well together with the difference in percentage of the coarse mesh for both the NRG-FSIFOAM and the STAR-CCM+ results. A clear trend is visible in the manner with which the frequencies and the damping ratios of the 1st and 3rd mode change along with finer meshes. The 2nd mode doesn't adhere to this, and even shows very large dissipative values in the damping ratio. Partially this can be explained through

	solver/mesh	f_1 [Hz]	f_2 [Hz]	f_3 [Hz]	ξ_1	ξ_2	ξ_3
STAR-CCM+	Coarse	45.40	145.60	296.29	2.165%	1.013%	1.317%
NRG-FSIFOAM	Coarse	46.55	146.46	317.59	4.463%	3.677%	1.080%
	Coarse-PFM	46.83	145.03	338.02			
	Mesh2	51.04	143.48	345.25	1.580%	2.875%	0.422%
	Mesh3	51.85	135.83	350.92	1.283%	10.392%	0.170%
	Mesh4	52.87	140.44	356.76	1.254%	10.817%	0.150%
Difference		2.53%	0.59%	7.19%	100.14%	262.98%	21.94%

Table 7.6: *Comparison between the frequencies and the damping ratios of the first three modes for a bare rod in water. The $y^+ \approx 60$ fluid mesh is used, together with different solid mesh resolutions, which are described in table 7.1. The frequencies of the PFM simulation are extracted from the data from 0.5s on to the end. The difference as a percentage is of the first two rows.*

the 2nd mode being more difficult to capture; the solid-only case in subsection 7.1.4 noted retrieval of the 3rd mode before the 2nd. Complementary to the fact that this mode is less prominently visible, the fitting method for retrieval of the data requires initial guesses for the parameters in eq. (7.2). A slight change in the values of these initial guesses therefore has a substantial influence on the parameter values w.r.t. 2nd mode. In the remainder all three modes are still calculated, however, critical observations will mostly be based on the 1st and 3rd modes.

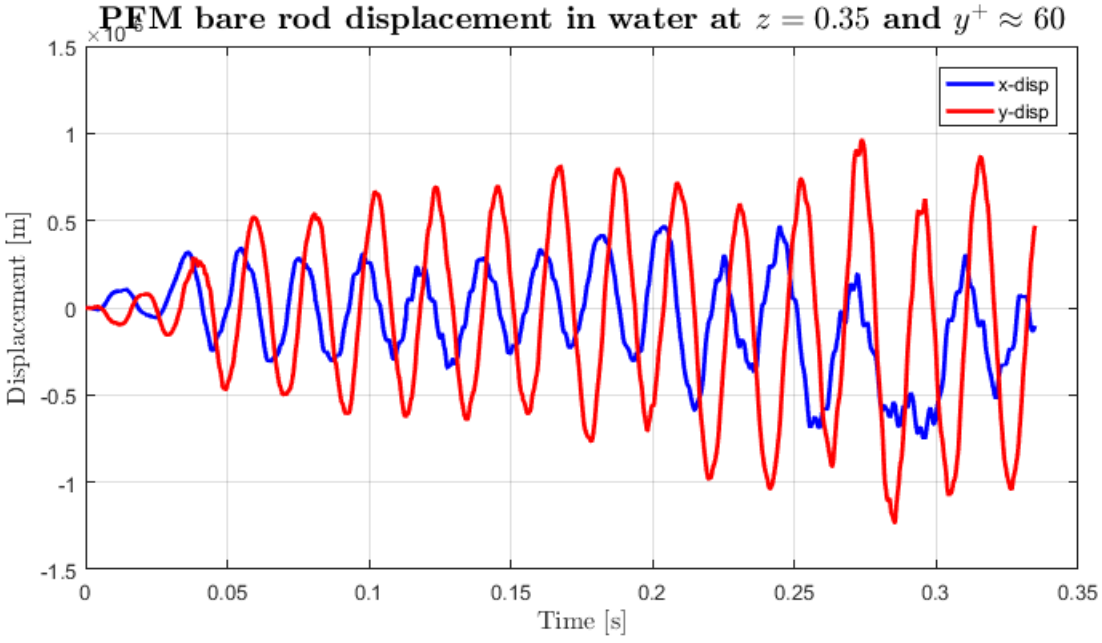


Figure 7.10: *Displacement history of the center point of a bare rod in the x- and y-direction in water flow, generated by means of a PFM simulation using a wall-function ($y^+ \approx 60$) fluid mesh.*

The displacements associated with the frequencies of the PFM simulation in table 7.6 are given in figure 7.10. All external forces exerted on the bare rod originate from turbulence. Starting from a non-deformed state, the rod initially experiences little impact

from the turbulent forces. Around $t = 0.025$s it is excited to move along its natural frequencies. It is clear that the pattern is more irregular than in the previous cases, this is due to the random terms (section 6.3) calculated by the PFM. As a consequence, symmetry is lost and it is more challenging to extract modal characteristics. In order to do so, the onset of the vibration is left out in the calculation of a fitting function (such as eq. (7.2)), and thus the fit is only based on the excited natural frequencies of the rod. For this case specifically this means the data from 0.5s on to the end is used. It can be concluded from table 7.6 that the found frequencies of both the impulsive forced rod simulation and the PFM simulation agree fairly well with each other.

In a similar manner as for the wall-function cases, the x- and y-displacement of the bare rod (coarse mesh) over time is extracted for water flow using a wall-resolved ($y^+ \approx 1$) mesh, this is visualized in figure 7.11. This time around the results show a lower natural frequency than the STAR-CCM+ results from De Santis et al. Quantitative data is gathered in table 7.7, in which it can be seen that the damping ratios are again much larger than the STAR-CCM+ results, but marginally smaller compared with the wall-function results. In [13] it is reported by De Santis et al. that for water flow, the two grid resolutions agree very well with one another, having a maximal difference of 0.7% in the frequencies. This difference between the grid resolutions is 11.8% for the NRG-FSIFOAM results. Several sources in both the software packages could produce these large differences in the behaviour of the rod.

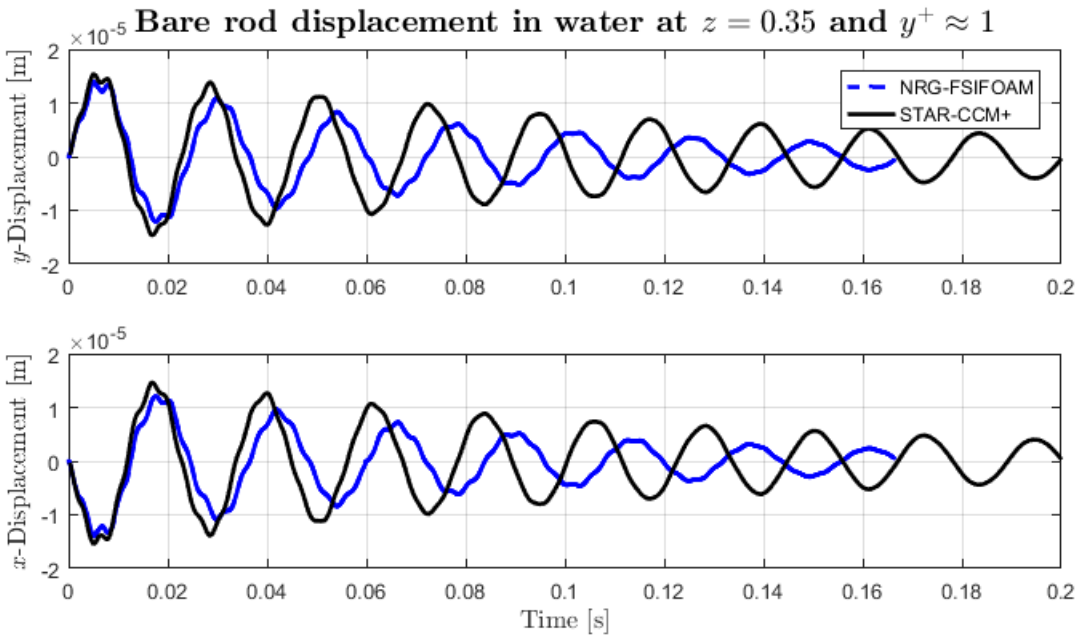


Figure 7.11: *Displacement history of the center point of a bare rod in the y-direction (top) and x-direction (bottom) in water flow, generated by means of a wall-resolved ($y^+ \approx 1$) fluid mesh. The results generated with NRG-FSIFOAM are compared to results generated by STAR-CCM+.*

As explained in chapter 6, different software use different wall-function models, which could partially explain the difference between the wall-function ($y^+ \approx 60$) and wall-resolved ($y^+ \approx 1$) results in NRG-FSIFOAM, compared with STAR-CCM+ having very

	solver	f_1 [Hz]	f_2 [Hz]	f_3 [Hz]	ξ_1	ξ_2	ξ_3
STAR-CCM+	U-RANS	45.08	145.54	295.93	2.479%	1.130%	1.361%
NRG-FSIFOAM	U-RANS	41.95	133.48	284.09	4.266%	2.888%	0.837%
Difference		7.46%	9.04%	4.17%	72.09%	155.59%	62.60%

Table 7.7: *Comparison between the frequencies and the damping ratios of the first three modes for a bare rod in water. The $y^+ \approx 1$ fluid mesh is used.*

similar results. The difference however is still quite large, a major influence due to a difference in wall-functions seems less likely. Moreover, the wall-function results of NRG-FSIFOAM and STAR-CCM+ compare fairly well, making it seem as if the $k-\omega$ SST model in OpenFOAM over-predicts the wall shear stress τ_t without wall-functions. In the past, there have been complaints in the OpenFOAM (cfd-online) community, describing that the implementation of Menter's $k-\omega$ SST model is very sensitive to mesh quality and initial conditions, leading to stability issues with wall-resolved meshes as a consequence. When fluid solutions do converge with wall-resolved meshes, it thus can not be excluded that parameter values for k and ω are incorrect. The difference between the wall-resolved results of NRG-FSIFOAM and STAR-CCM+ could be an indication agreeing with the claim of OpenFOAM's $k-\omega$ SST model to be less accurate without wall-functions. In order to distinctively pick out a reason, both NRG-FSIFOAM's and STAR-CCM+'s $k-\omega$ SST model with and without wall-functions will have to investigated in a more thorough manner.

Akin to the above results, the dynamics of the bare rod are investigated when it is submerged in a lead-bismuth eutectic (PbBi). In figures 7.12 and 7.13 the x- and y-displacement history over time are shown of the impulsive force simulation and the PFM simulation, respectively. Compared with the wall-function grid results of water flow, the PbBi results fit better to STAR-CCM+ results, especially the difference in damping ratio is less. Also noteworthy to mention is that the frequency of the NRG-FSIFOAM results are slightly lower, in contrast with water flow, where the frequency was higher with the wall-function grid.

	solver	f_1 [Hz]	f_2 [Hz]	f_3 [Hz]	ξ_1	ξ_2	ξ_3
STAR-CCM+	U-RANS	22.40	73.88	150.14	10.583%	3.993%	2.121%
NRG-FSIFOAM	U-RANS	21.89	65.08	153.29	15.432%	4.895%	3.068%
	U-RANS + PFM	22.04	65.63	154.33			
Difference		2.33%	13.52%	2.10%	45.82%	22.59%	44.65%

Table 7.8: *Comparison between the frequencies and the damping ratios of the first three modes for a bare rod in PbBi flow. The $y^+ \approx 60$ fluid mesh is used. The frequencies of the PFM simulation are extracted from two parts and then averaged, namely the data between 0.05s and 0.17s and data from 0.21s until the end.*

These observations are quantified in table 7.8. The difference of the damping ratio ξ_1 is equal to 45.82%, whereas for water flow this is 100.14%. One can explain this by considering the two major forces working on the structure, namely the contribution of viscous forces and the contribution resulting from the inertia of the displaced fluid surrounding the rod. PbBi having a density of approximately ten times the density of

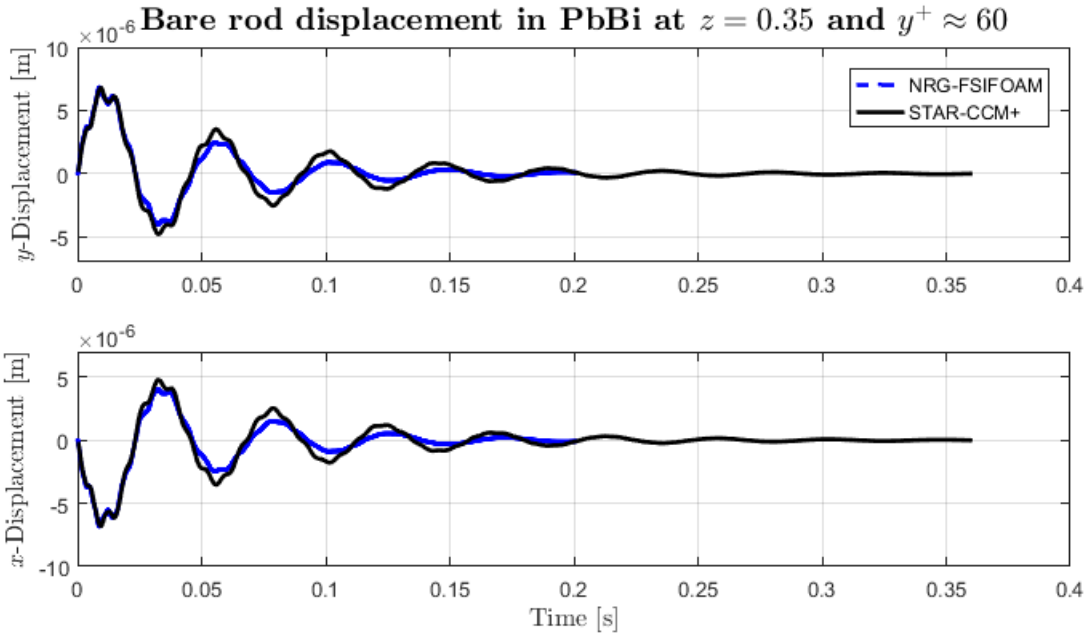


Figure 7.12: *Displacement history of the center point of a bare rod in the y-direction (top) and x-direction (bottom) in PbBi flow, generated by means of a wall-function ($y^+ \approx 60$) fluid mesh. The results generated with NRG-FSIFOAM are compared to results generated by STAR-CCM+.*

water has a larger fraction of the rod behaviour owing to the inertia of the fluid itself. Hence, the damping is less affected compared with water flow.

The impulsive force results and the PFM results again show very good agreement with each other, even more so as for water flow. On first sight, it is very difficult to see this from figure 7.13, therefore frequencies were extracted from data between 0.05s and 0.17s and data from 0.21s until the end, and eventually averaged to obtain the values in table 7.8. Although the natural frequency of the rod is excited with the PFM algorithm, it can be noticed an irregular movement pattern remains existing due to turbulent forces.

	solver	f_1 [Hz]	f_2 [Hz]	f_3 [Hz]	ξ_1	ξ_2	ξ_3
STAR-CCM+	U-RANS	22.24	72.62	149.54	10.909%	4.333%	2.310%
NRG-FSIFOAM	U-RANS	18.44	59.09	128.77	14.650%	6.168%	2.926%
Difference		20.61%	22.90%	16.13%	34.29%	42.35%	26.67%

Table 7.9: *Comparison between the frequencies and the damping ratios of the first three modes for a bare rod in PbBi flow. The $y^+ \approx 1$ fluid mesh is used.*

At last, x- and y-displacements of a shorter simulation with the wall-resolved PbBi mesh are shown in figure 7.14. From the knowledge obtained, similar conclusions can be made: using a wall-resolved fluid grid decreases the natural frequencies drastically and due to the much larger density of PbBi, the difference in the damping ratios is much less as for the wall-resolved water grid. These conclusions are verified in numbers in table 7.9. Moreover, comparing the damping ratios ξ_1 and ξ_3 with the ones from the wall-function results in table 7.8, again it is noticed that the values of the wall-resolved grid are a bit

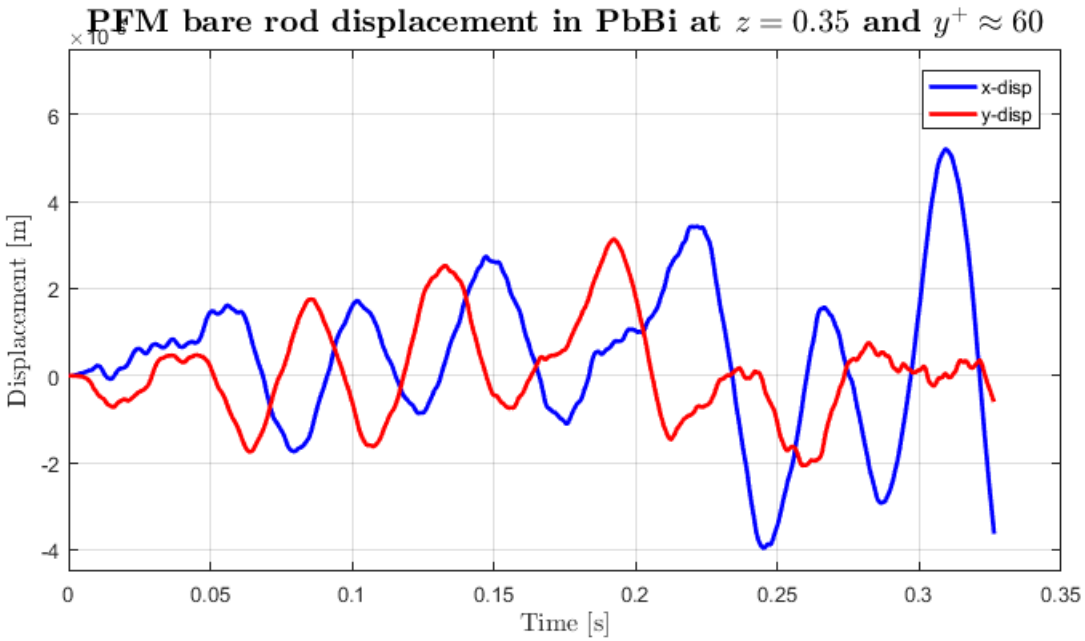


Figure 7.13: *Displacement history of the center point of a bare rod in the x- and y-direction in PbBi flow, generated by means of a PFM simulation using a wall-function ($y^+ \approx 60$) fluid mesh.*

lower. It appears, as this also was the case for water flow, that the wall-function treatment in OpenFOAM tends to slightly over-predict the wall shear stress τ_t.

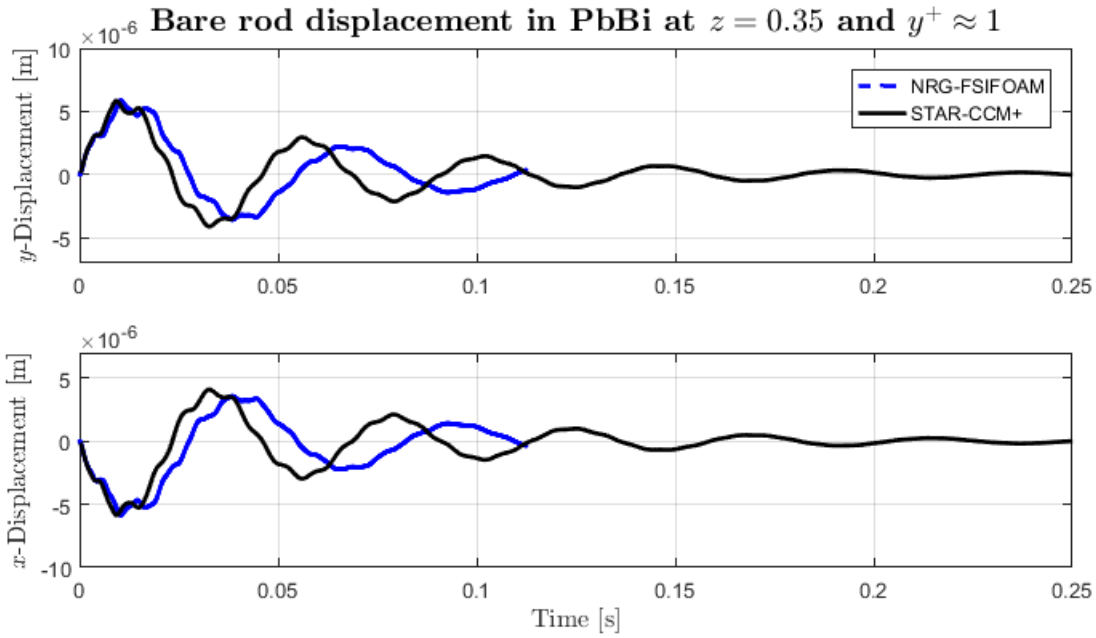


Figure 7.14: *Displacement history of the center point of a bare rod in the y-direction (top) and x-direction (bottom) in PbBi flow, generated by means of a wall-resolved ($y^+ \approx 1$) fluid mesh. The results generated with NRG-FSIFOAM are compared to results generated by STAR-CCM+.*

7.2 Flow-Induced Vibrations of a wire-wrapped rod

Following the strategy of the previous single bare rod case, similar simulations are performed for a wire-wrapped rod of the same dimensions. In reality, nuclear fuel assemblies are equipped with additional mechanical instruments that reduce the amount of vibration. One of these concepts is the use of a wire-spacer helically wrapped around the fuel rod to prevent the rods from hitting one another. The fluid flow having to follow the helical wire, will therefore induce a different oscillation pattern on the rod, making it a challenging FSI case to investigate. Because the main description of the wire-wrapped case will be mostly the same as the bare rod case, only the differences of the properties are highlighted.

7.2.1 Physical and Geometrical Properties

Only the solid rod is extended with a wire. Therefore only the way this wire is wrapped around the rod is mentioned. All different properties both for the fluid and solid are mentioned in subsection 7.1.1.

The wire is considered to be circular in a fixed diameter $d_s = 1.8$ mm, helically wrapped around with a fixed pitch $P = 265$ mm. This results in a total amount of 2.6 revolutions around the rod. Moreover, in order to avoid numerical complications by having point contact, a flat region is assumed to be in between the rod and the wire, having a height $h = 0.07$ mm. The geometry (as a mesh) is shown in figure 7.15.

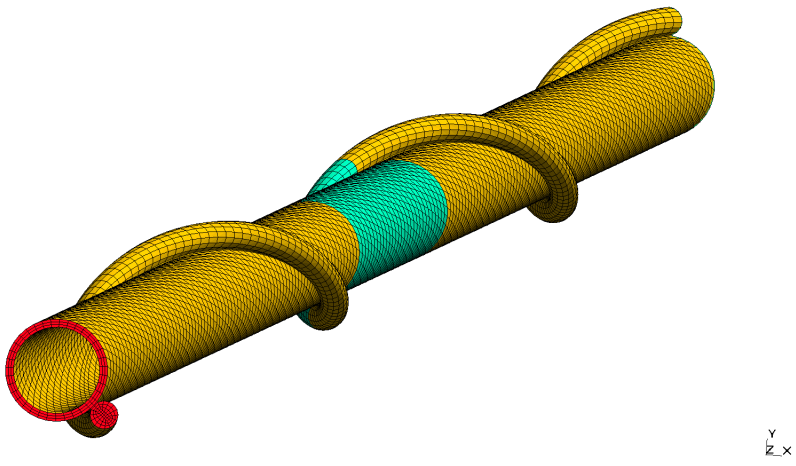


Figure 7.15: *The wire-wrapped rod mesh considered for simulations. Different colors denote different physical surfaces; the cyan color denotes the patch to which the function δ_F (7.1) is evenly applied, the outlet boundary is red, the free parts are yellow and the inlet boundary is not visible.*

7.2.2 Meshes, discretizations, B.C. and I.C.

The solid mesh has 170 quadrilateral elements in the cross-section, which are extruded and twisted in agreement with the wire spacer pitch P with 200 axial elements. In total

this means the solid solver uses a mesh of 34000 elements. This mesh is visualized in figure 7.15, in which the coloured patches represent physical surfaces which are constrained by differing boundary conditions, the cyan patch being used to apply the impulsive force δ_F (7.1). Moreover, a second mesh (denoted by 'Fine') is once used in a mesh sensitivity check validated in water flow. This solid mesh has the same subdivision of elements in the cross-section, but a total of 400 elements in the axial direction. DEAL-II solves the structural dynamics of the rod in all cases with the theta-method, $\theta = 0.6$ and a time-step $\Delta t = 5 \cdot 10^{-5}$s. Only the wall-function PFM simulation of water flow uses a time-step of $\Delta t = 10^{-4}$s.

In the wire-wrapped rod case, generating a fully structured mesh as for the single bare rod case is complicated. For this reason an unstructured prism layers/polyhedral mesh is generated in the cross section of the domain. Thereafter this mesh is extruded and twisted simultaneously w.r.t. the pitch of the wire by 400 divisions in the axial direction. Prism layers are surrounding the FSI walls, except for the small gap between the wire and the rod, because the velocity is assumed to be small in this region. Below, the fluid wall-resolved ($y^+ \approx 1$) mesh and the wall-function ($y^+ \approx 60$) mesh are shown for water flow in figure 7.16. In table 7.10 a summary is given of the total amount of cells of the

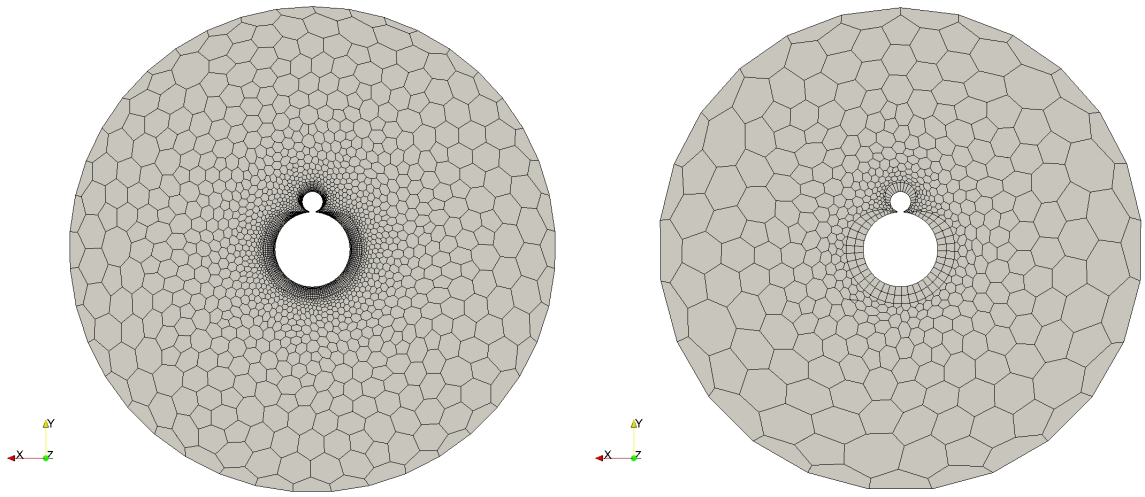


Figure 7.16: *Cross-section of the fluid meshes considered for the FSI simulations of a wire-wrapped rod in axial water flow. Off-the-wall heights of the first cell are specifically chosen s.t. a $y^+ = 1$ mesh (left) and a $y^+ = 60$ mesh (right) are generated. Near the FSI interface, prims layers are adopted, in the other parts of the domain polyhedral elements are considered.*

fluid domain in all cases. As before, the U-RANS approach with the k-ω SST model is adopted to solve the transient simulation. The discretization techniques, boundary and initial conditions all remain the same as in the bare rod case, these are given in tables 7.3 and 7.4. As well the IQN-ILS method with maximum 60 coupling iteration and reuse of 8 time-steps is taken over in order to couple the two solvers.

	mesh type	elements
H_2O	$y^+ \approx 1$	1211200
	$y^+ \approx 60$	398000
PbBi	$y^+ \approx 1$	1656400
	$y^+ \approx 60$	722800

Table 7.10: *Numerical fluid domain considered for the FSI simulations of a bare rod in tube with axial water flow.*

7.2.3 Results and discussion

Analogously to the bare rod case, time history displacements of a single mesh point are extracted to generate results; it is assumed the surrounding mesh points undergo a similar displacement, making it a valuable approach to establish results on the behaviour of the complete wire-wrapped rod. The actual mesh of the wire-wrapped rod is slightly different essentially than the meshes in the bare rod case, leading to different coordinates being used, namely $x = -0.0032744, y = 6.28778 \cdot 10^{-5}$ and $z = 0.35$.

The complete methodology and structure of the discussion is similar to that of the bare rod case; frequencies and damping ratios are extracted first for water flow, showing both results for impulse forced and PFM simulations, thereafter the same is done for PbBi results.

Figures 7.17 and 7.18 display the time history of the x- and y-displacement for the wall-function grid in water flow. At first sight it can be noticed that the NRG-FSIFOAM shows a higher frequency compared with the STAR-CCM+ results. Differences in the solid mesh are dominated by a decrease in damping ratio for an increasing amount of axial elements. Also a slight increase in the natural frequencies is observed. This agrees well with the observations made for the bare rod case. Perhaps the most interesting details from these figures are related to the loss of symmetry of the FSI case; due to the wire-spacer the x- and y-displacements are not the same and the x-displacement is not oscillating around zero displacement towards the end of the simulation.

	solver/mesh	f_1 [Hz]	f_2 [Hz]	f_3 [Hz]	ξ_1	ξ_2	ξ_3
STAR-CCM+	U-RANS	46.97	149.20	302.00	2.151%	0.925%	1.165%
NRG-FSIFOAM	U-RANS	49.47	161.85	330.83	3.246%	2.459%	2.087%
	U-RANS + PFM	50.46	158.26	334.74			
	U-RANS/Fine	51.14	165.42	338.00	2.143%	1.321%	1.845%
Difference		5.32%	8.48%	9.55%	50.91%	165.84%	79.14%

Table 7.11: *Comparison between the frequencies and the damping ratios of the first three modes for a wire-wrapped rod in water flow. The $y^+ \approx 60$ fluid mesh is used, and compared for a coarse and fine solid mesh. The frequencies of the PFM simulation are extracted from the data between 0.11s up onto the end.*

As a consequence of the loss in symmetry, it matters from which data the modal parameters are extracted. Therefore, in all cases it is chosen to use the y-displacement for this purpose. Frequencies and damping ratios for the impulsive force simulations and a PFM simulation are reported in table 7.11. In general, the results show very similar

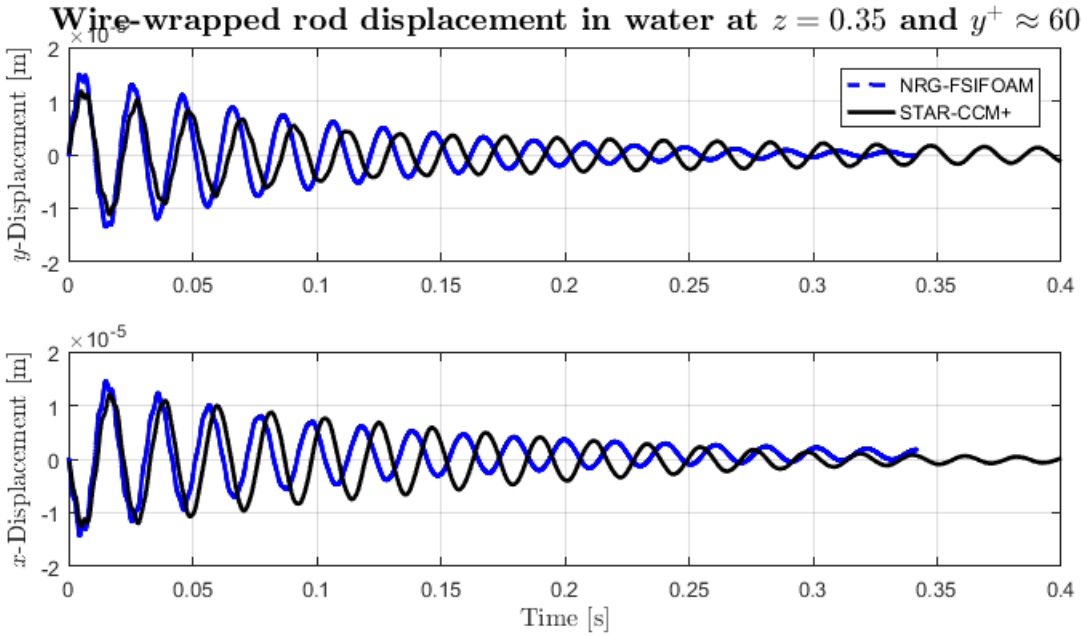


Figure 7.17: *Displacement history of the center point of a wire-wrapped rod in the y-direction (top) and x-direction (bottom) in water flow, generated by means of a wall-function ($y^+ \approx 60$) fluid mesh. The results generated with NRG-FSIFOAM are compared to results generated by STAR-CCM+.*

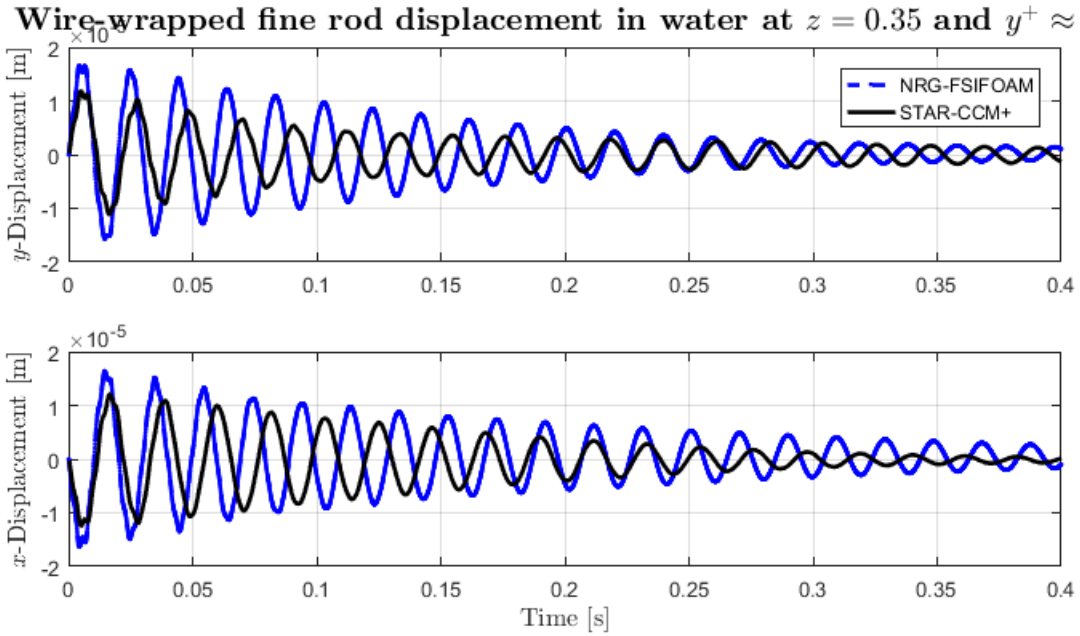


Figure 7.18: *Displacement history of the center point of a wire-wrapped rod in the y-direction (top) and x-direction (bottom) in water flow, generated by means of a wall-function ($y^+ \approx 60$) fluid mesh and a finer solid mesh, having double the amount (400) of elements in the axial direction. The results generated with NRG-FSIFOAM are compared to results generated by STAR-CCM+.*

behaviour as in the bare rod case, having increasing values of frequency and diminishing damping ratio with mesh refinement. The dominating frequency f_1 varies a bit more from

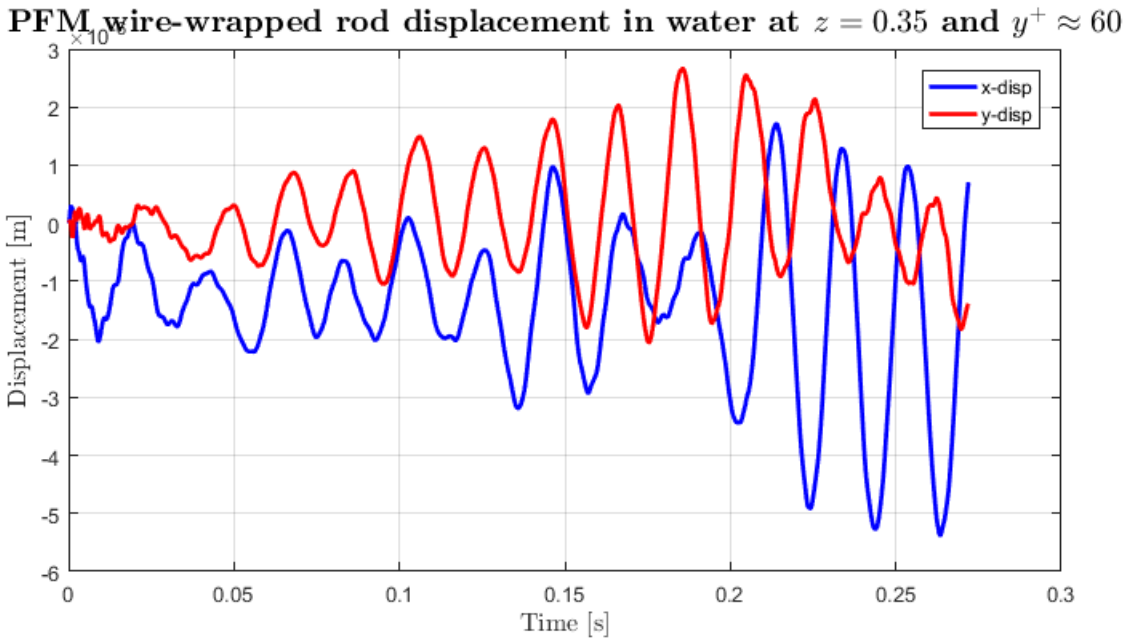


Figure 7.19: *Displacement history of the center point of a wire-wrapped rod in the x- and y-direction in water flow, generated by means of a PFM simulation using a wall-function ($y^+ \approx 60$) fluid mesh.*

the STAR-CCM+ data in the wire-wrapped case, having a difference of 5.32%, compared to 2.59% for the bare rod case.

The displacements affiliated with the PFM results (with the coarse solid mesh) in table 7.11 are given in figure 7.19. It is interesting that the chaotic behaviour of the turbulence almost immediately sends the wire-wrapped rod along a path in the x-displacement not centred around the zero displacement. The frequencies of the 1st and 3rd mode are a bit higher than obtained with the impulsive forcing.

The time history of the x- and y-displacement of the wall-resolved ($y^+ \approx 1$) mesh for water flow are reported in figures 7.20 and 7.21 for the impulsive force and the PFM simulation, respectively. Again, akin to behaviour of the bare rod, a substantial decrease in the frequency is observed for the wall-resolved fluid grid. In this case one can see a more distinct difference in the movement in x- and y-direction, especially the damping in the x-direction is larger. The frequencies and damping ratios are reported in table 7.12. Again, there is large difference between the wall-function grid results and the wall-resolved grid results, having a maximum difference in the frequencies of 20.89%, whereas Desantis et al. reported a value of 0.3% for this difference. In order to single out an explanation for this behaviour, further investigation needs to be done on the $k - \omega$ SST models of both NRG-FSIFOAM and STAR-CCM+.

Running simulations of PbBi fluid submerging a wire-wrapped rod is computationally very intensive. Moreover, from the bare rod case it is known that the frequency decreases drastically in PbBi flow, and thus less periods of oscillation are captured for the same simulation time as in water. Therefore it is noted at forehand that the results shown below are relatively short. The impulsive force simulation and PFM simulation results

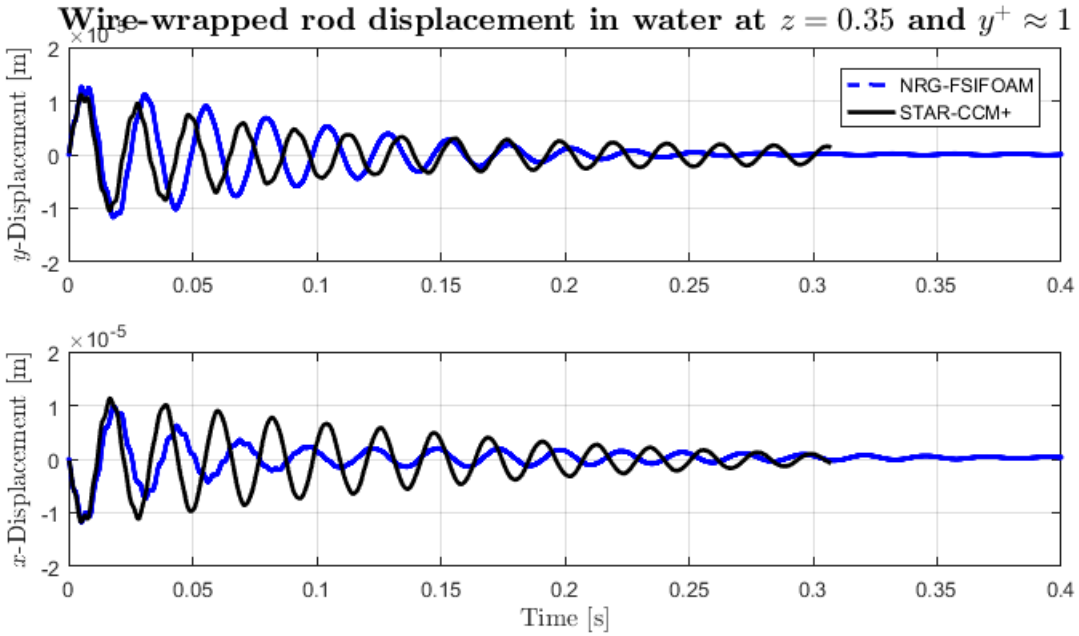


Figure 7.20: *Displacement history of the center point of a wire-wrapped rod in the y-direction (top) and x-direction (bottom) in water flow, generated by means of a wall-resolved ($y^+ \approx 1$) fluid mesh. The results generated with NRG-FSIFOAM are compared to results generated by STAR-CCM+.*

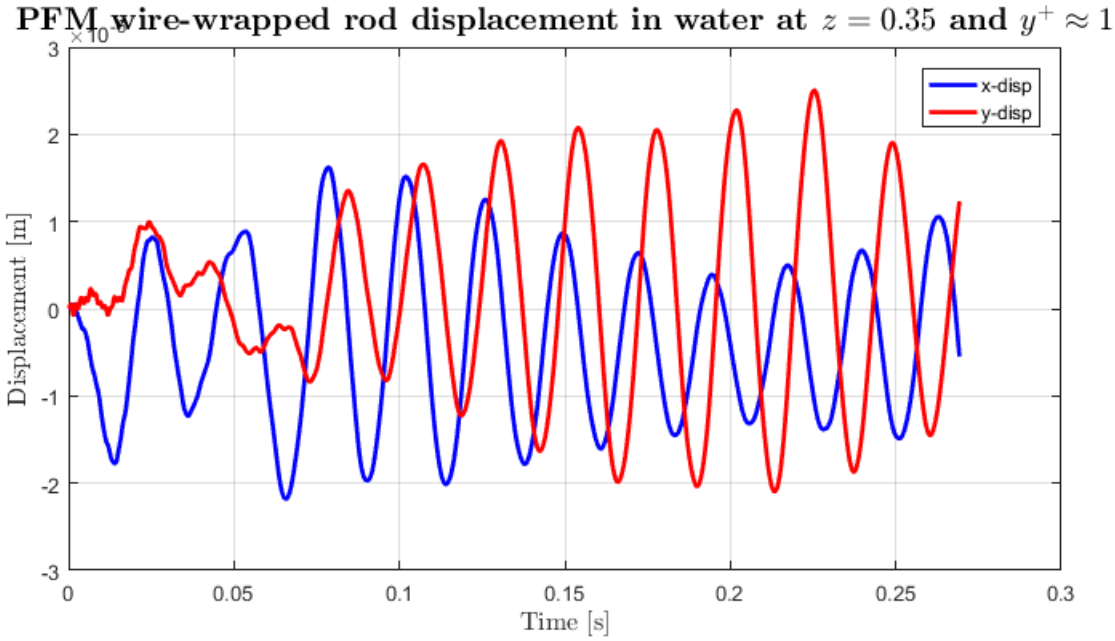


Figure 7.21: *Displacement history of the center point of a wire-wrapped rod in the x- and y-direction in water flow, generated by means of a PFM simulation using a wall-resolved ($y^+ \approx 1$) fluid mesh.*

for the wall-function meshes are given in figures 7.22 and 7.24, respectively. The essential movement of the rod looks very different between the two solvers; it seems as if the x-displacement of the NRG-FSIFOAM results aligns more with the y-displacement of the STAR-CCM+ results and vice-versa. This becomes more evident when plotting the

	solver	f_1 [Hz]	f_2 [Hz]	f_3 [Hz]	ξ_1	ξ_2	ξ_3
STAR-CCM+	U-RANS	46.82	149.08	301.81	2.425%	0.983%	1.146%
NRG-FSIFOAM	U-RANS	40.92	162.89	282.71	4.082%	9.705%	1.840%
	U-RANS + PFM	42.34	163.82	279.63			
Difference		14.42%	9.26%	6.76%	68.33%	887.28%	60.56%

Table 7.12: *Comparison between the frequencies and the damping ratios of the first three modes for a wire-wrapped rod in water flow. The $y^+ \approx 1$ fluid mesh is used. The frequencies of the PFM simulation are extracted from the data between 0.07s up onto the end.*

irregular trajectories as in figure 7.23. More than likely the rod was rotated in a 90 degree position compared with the STAR-CCM+ simulation. Also due to the higher density of the lead-bismuth eutectic, it is now more clearly visible that the final positioning of the wire-wrapped rod tends towards a different than zero displacement in both directions, which is much larger in the case of PbBi flow.

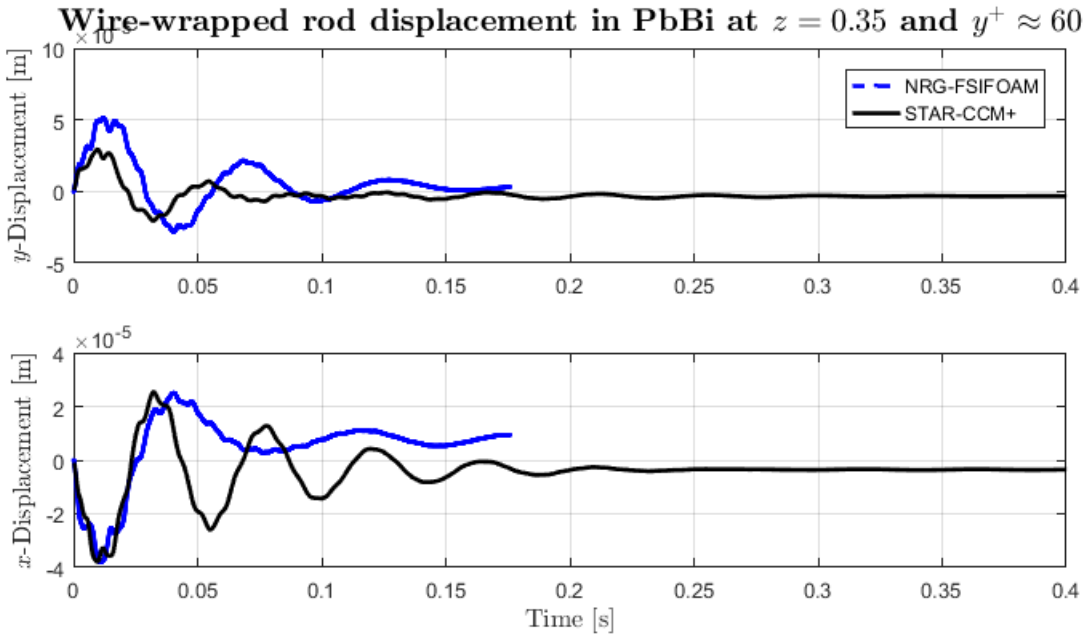


Figure 7.22: *Displacement history of the center point of a wire-wrapped rod in the y-direction (top) and x-direction (bottom) in PbBi flow, generated by means of a wall-function ($y^+ \approx 60$) fluid mesh. The results generated with NRG-FSIFOAM are compared to results generated by STAR-CCM+.*

Quantitative data of both the simulations is reported in table 7.13. Compared to the bare rod case, where it was remarked for PbBi flow that the frequency decreased slightly, now the frequency is quite a bit more lower than the STAR-CCM+ case. Moreover, this is the first case where the damping ratio is lower than the one obtained via STAR-CCM+. Based on the assumption that the rod is rotated by 90 degrees, also the frequencies and damping ratios of the x-displacement are given in table 7.13. The difference in the damping ratio ξ_1 is then only 13.29%, which aligns more with the previously obtained results, i.e., higher damping ratio for the NRG-FSIFOAM solver.

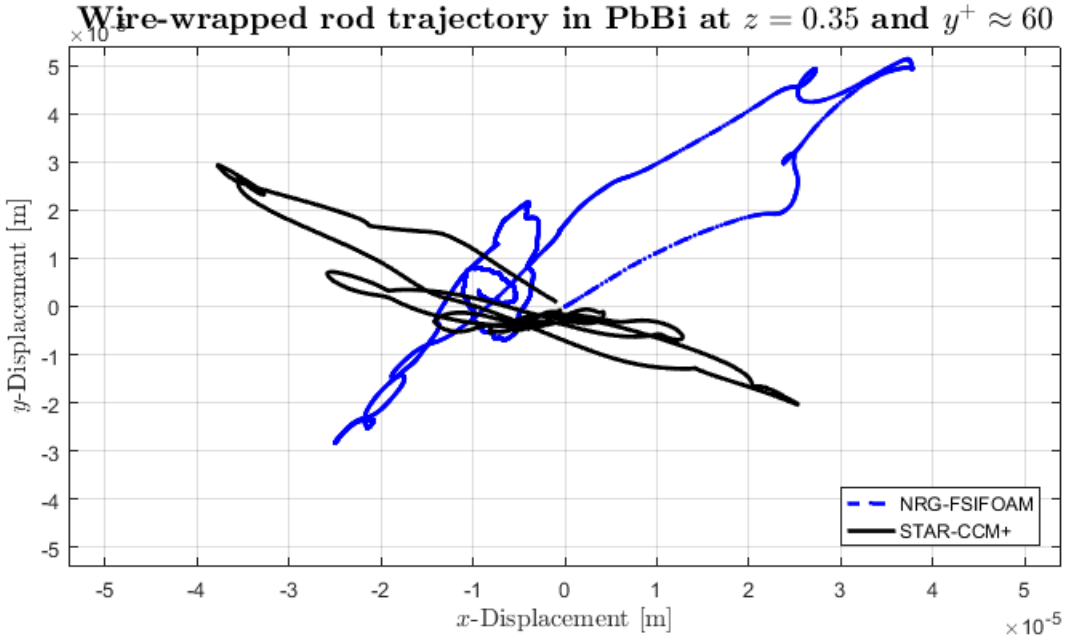


Figure 7.23: *Trajectory path of the center point of a wire-wrapped rod in PbBi flow, generated by means of a wall-function ($y^+ \approx 60$) fluid mesh. The results generated with NRG-FSIFOAM are compared to results generated by STAR-CCM+.*

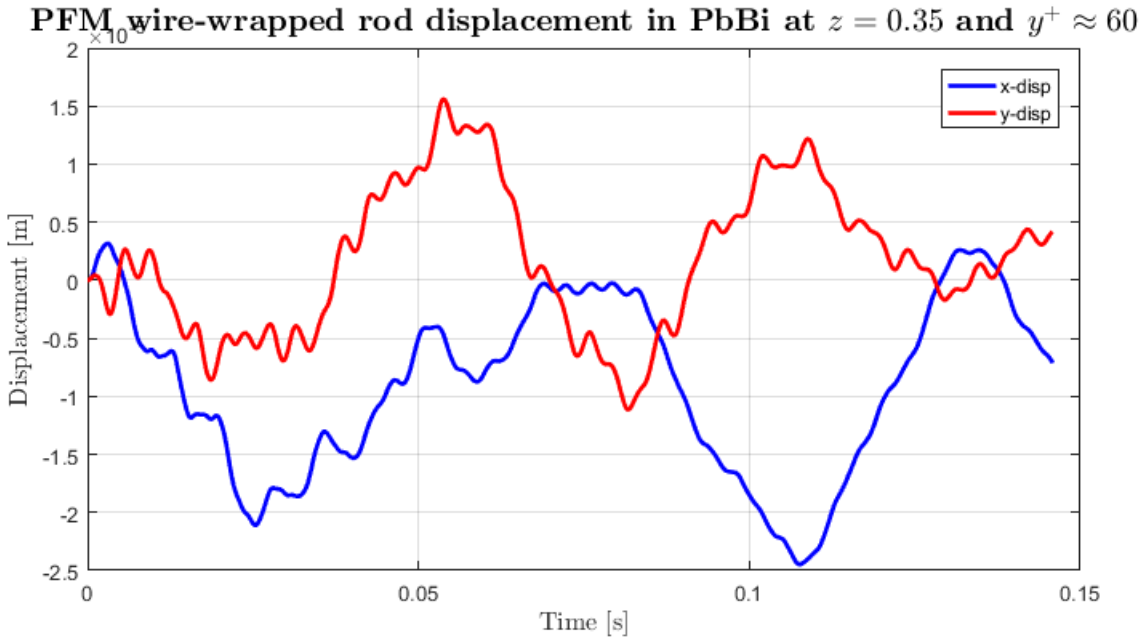


Figure 7.24: *Displacement history of the center point of a wire-wrapped rod in the x- and y-direction in PbBi flow, generated by means of a PFM simulation using a wall-function ($y^+ \approx 60$) fluid mesh.*

Very similar time history behaviour is obtained in figures 7.25 and 7.26 for the wall-resolved PbBi fluid grid, forced by an impulse and the PFM, respectively. Again, assuming the rod was rotated 90 degrees, the frequencies and damping ratios are also extracted for the x-displacement in table 7.14. It can be seen that no PFM data is given, as approximately only half a period of oscillation was obtained using the PFM method. This

	solver	f_1 [Hz]	f_2 [Hz]	f_3 [Hz]	ξ_1	ξ_2	ξ_3
STAR-CCM+	U-RANS	23.59	72.65	152.90	23.364%	8.159%	2.006%
NRG-FSIFOAM	U-RANS	17.83	55.62	122.94	15.185%	1.434%	2.887%
	U-RANS (x)	16.08	27.59	122.71	26.470%	14.105%	4.143%
	U-RANS + PFM	17.54	60.50	124.56			
Difference		32.31%	30.62%	24.37%	53.86%	468.97%	43.92%
Difference (x)		46.70%	163.32%	24.60%	13.29%	78.88%	106.53%

Table 7.13: *Comparison between the frequencies and the damping ratios of the first three modes for a wire-wrapped rod in PbBi flow. The $y^+ \approx 60$ fluid mesh is used. Both the x- and y-displacement are compared with the y-displacement of the STAR-CCM+ results. Extraction of the frequencies of the PFM data is done using to full data.*

is not enough to do accurate prediction of the natural frequencies. Similar as in the bare rod case, the following observations can be made: using a wall-resolved mesh decreases the natural frequencies, the damping ratio ξ_1 decreases and in percentages it very clear that the damping ratios are closer to each other for both solvers in PbBi flow than in water flow.

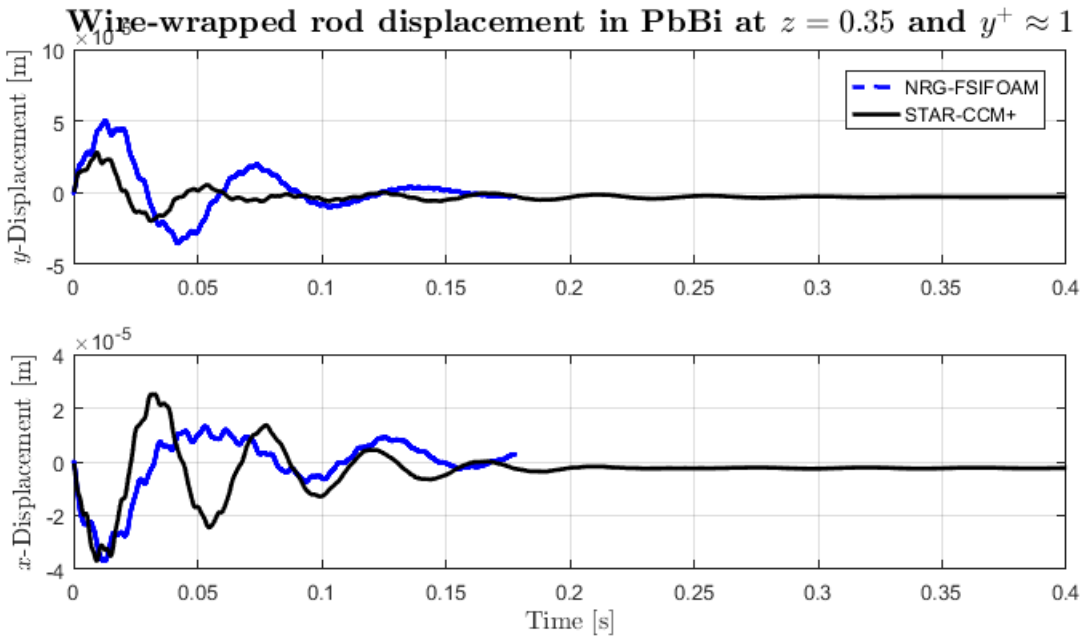


Figure 7.25: *Displacement history of the center point of a wire-wrapped rod in the y-direction (top) and x-direction (bottom) in PbBi flow, generated by means of a wall-resolved ($y^+ \approx 1$) fluid mesh. The results generated with NRG-FSIFOAM are compared to results generated by STAR-CCM+.*

	solver	f_1 [Hz]	f_2 [Hz]	f_3 [Hz]	ξ_1	ξ_2	ξ_3
STAR-CCM+	U-RANS	24.02	72.32	154.62	23.604%	8.847%	1.901%
NRG-FSIFOAM	U-RANS	16.83	38.18	114.27	15.305%	2.524%	2.880%
	U-RANS (x)	13.07	26.35	112.00	22.422%	14.244%	2.066%
Difference		42.72%	89.42%	35.31%	54.22%	250.52%	51.50%
Difference (x)		83.78%	174.46%	38.05%	5.27%	61.00%	8.68%

Table 7.14: *Comparison between the frequencies and the damping ratios of the first three modes for a wire-wrapped rod in PbBi flow. The $y^+ \approx 1$ fluid mesh is used. Both the x- and y-displacement are compared with the y-displacement of the STAR-CCM+ results.*

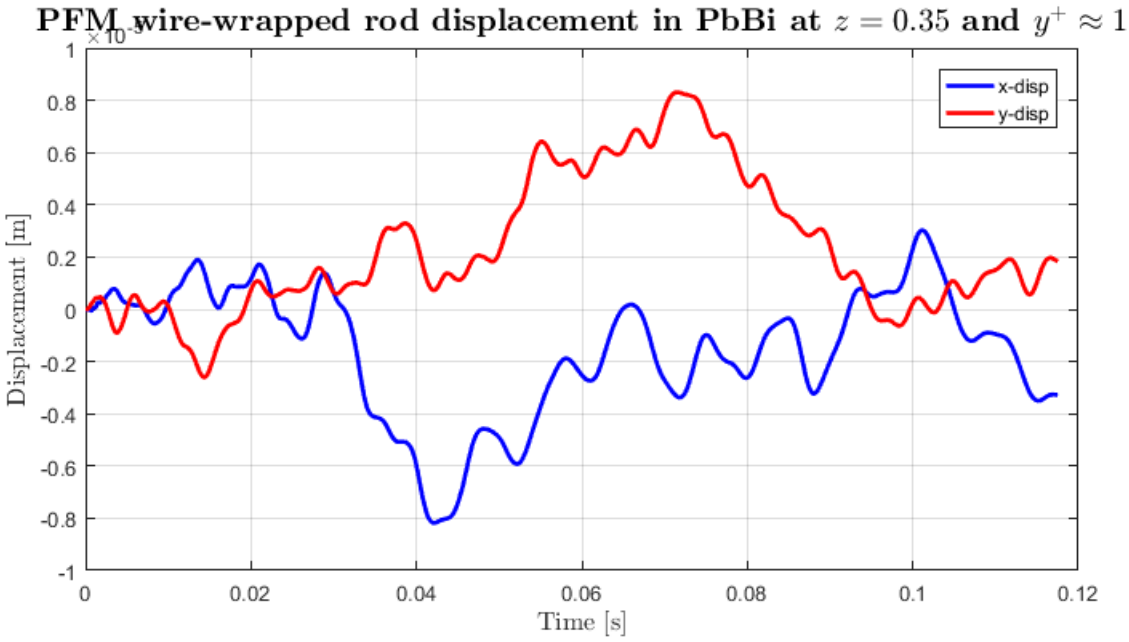


Figure 7.26: *Displacement history of the center point of a wire-wrapped rod in the x- and y-direction in PbBi flow, generated by means of a PFM simulation using a wall-resolved ($y^+ \approx 1$) fluid mesh.*

Chapter 8

Assessment of improvements to the fluid-structure coupling in NRG-FSIFOAM

In this chapter, a short discussion is given on the suggested improvements (see chapter 5) to the coupling in NRG-FSIFOAM in order to obtain an overall better performance. Under these, first the influence of different extrapolation methods is investigated, thereafter the current status of the Manifold Mapping algorithm in NRG-FSIFOAM is mentioned.

8.1 Comparison of different extrapolation methods

Letting a numerical solver for a Fluid-Structure Interaction (FSI) problem march through time requires an initial state of the system on each time step. The numerical solver pursues the actual stresses and positioning of the FSI system from this initial state. Especially in the case of strong coupling (see section 5.1.2) in FSI, sub-iterated schemes are often used, which consequently places plenty importance in having a proper initial guess for the actual system state at each time step. This way, a large amount of time being consumed by the numerical solver in coupling can possibly be reduced.

In section 5.3, a theoretical discussion with a small test is given on extrapolation methods with regard to FSI. A novel extrapolation method, given by eq. (5.38), has been suggested, manually adapting for smooth or largely unpredictable behaviour of the structure displacement. This method shows promising results on a simplified test case. For this reason, further investigation on extrapolation is studied to see if advancement in computational efforts has been made for realistic FSI problems.

To fully catch the potency of the extrapolation method (5.38), the bare rod case in a lead-bismuth eutectic (section 7.1) being displaced by an impulsive force is considered. Firstly, due to the impulsive force in the first few steps the rod is pushed in a direction that is less expected, afterwards, smooth oscillation of the rod prevails. All simulations are run for $t = 0.05$ s with $\Delta t = 0.0001$ s. Next to the extrapolations eq. (5.9) ($\boldsymbol{d}^{n+1} = \frac{5}{2}\boldsymbol{d}^{n} - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2}$), eq. (5.29) ($\boldsymbol{d}^{n+1} = 3\boldsymbol{d}^{n} - 3\boldsymbol{d}^{n-1} + \boldsymbol{d}^{n-2}$) and the new one, which are based on three previous time-steps, also the first-order accurate extrapolation $\boldsymbol{d}^{n+1} = \boldsymbol{d}^{n}$ and the second-order accurate extrapolation $\boldsymbol{d}^{n+1} = 2\boldsymbol{d}^{n} - \boldsymbol{d}^{n-1}$ are considered. All coupling iterations are performed by means of the IQN-ILS method (section 5.2) with the

reuse of information from eight previous time steps. The results are summarised in table 8.1.

Method	$\boldsymbol{d}^{n+1} = \boldsymbol{d}^{n}$	$\boldsymbol{d}^{n+1} = 2\boldsymbol{d}^{n} - \boldsymbol{d}^{n-1}$	(5.9)	(5.29)	new
Time $[s]$	99269	92712	83296	86573	97604
Total iterations	2298	2084	1961	1965	2131
Average per time step	4.296	4.168	3.922	3.930	4.262

Table 8.1: *Information on the CPU costs related to the IQN-ILS method in a bare rod simulation submerged in PbBi with a wall-function mesh. Several extrapolation methods are compared to one another.*

Contrary to what was expected in section 5.3, the extrapolation (5.9) on average leads to less iterations on the smooth part of the rod displacement. Moreover, the difference in the amount of iterations of the second-order accurate extrapolation (5.9) and the third-order accurate extrapolation (5.29) is small. Consequently, the new extrapolation results are of no surprise to perform less well. Especially due to the fact that the novel extrapolation (5.36) spends a relatively large amount of time on the computation of $\boldsymbol{\omega}^{n+1}$ there is quite a gap with the methods (5.9) and (5.29).

The main question that has arisen is *why there is nearly no difference between the results of extrapolations (5.9) and (5.29).* The key concept of the new extrapolation method (5.36) is to combine both of them via a variability parameter. Any improvement is immediately nullified if the difference in speed of extrapolations (5.9) and (5.29) is absent for nearly any FSI problem. Having this knowledge determines if further investigation of extrapolations with a weighting parameter can possibly increase efficiency or not.

Possibly this is due to the fact that the small test case in MATLAB compares to the actual value, whereas an actual FSI simulation does not. A convergence error based on differences of residuals on subsequent iterations is used. Hence, smoothness does not necessarily influence the amount of coupling iterations. However, there might be a big difference in accuracy compared to reality.

Assuming in general there is a significant difference in eqs. (5.9) and (5.29), more questions arise:

- *Is the bare rod case submerged in a lead-bismuth eutectic qualified in order to take advantage of the difference in extrapolations (5.9) and (5.29)?*
- *How many coupling iterations are requested for a general FSI case s.t. the extra computing time is negligible?*

Answering the questions requires evaluation of a simulation for which it is assured that the software will spend more time in the coupling iterations. For this the Pressure Fluctuation Model (PFM) is applied to introduce more variable behaviour. From the previous section it is known that the PFM algorithm takes a short period of time to excite the natural frequencies. Hence a strongly-coupled PFM simulation can serve as the ideal test for the new extrapolation method. To this end, the wire-wrapped rod with wall-function water grid is simulated according to the description of section 7.2.1. All methods based

on three previous time-steps are applied to this case, namely eq. (5.9), eq. (5.29) and the new extrapolation. Table 8.2 summarizes the results in a similar fashion as the previous case, except for the fact that the PFM simulation is not constraint to a final time. Therefore the data is as well compared at the final time-step of the shortest PFM simulation, namely using eq. (5.29) for 799 steps.

Method	(5.9)	(5.29)	new
Time-steps	1416 / 799	799	2719 / 799
Time [s]	795588 / 482779	620780	1578926 / 496859
Total iterations	18812 / 11194	12609	35527 / 11400
Average per time step	13.285 / 14.010	15.781	13.066 / 14.268

Table 8.2: *Information on the CPU costs related to the IQN-ILS method in a wire-wrapped rod PFM simulation submerged in water with a wall-function mesh. Extrapolation methods using three previous time-steps are compared.*

First of all, it is remarked that a greater amount of coupling iterations are needed for this case with IQN-ILS algorithm. This has a great effect on the performance difference of the three methods, as now the new extrapolation method performs the least amount of iterations on average. Looking at the total simulation time, the third-order accurate extrapolation method ($\boldsymbol{d}^{n+1} = \frac{5}{2}\boldsymbol{d}^{n} - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2}$, eq. 5.9) performs the best. Both questions posed earlier can therefore be answered partially:

- The bare rod case considered indeed does not seem to employ enough coupling iterations to perform faster than one of the other extrapolations (5.9) and (5.29).
- The wire-wrapped rod on the average did use slightly less amount of iterations on the average. However the won time from this is overcompensated by the extra calculations involved of the weighting vector $\boldsymbol{\omega}^{n+1}$. Possibly the new extrapolation can be more beneficial with a computationally more involved FSI case with stronger coupling, although this is not assured.

8.2 Manifold mapping

The manifold mapping algorithm implemented in the preCICE library has been integrated in the NRG-FSIFOAM library by modifying the fluid adapter. This makes it possible to use two different fluid models in parallel, under the constraint of having the same parts of the fluid-structure interface on the same processors.

The well-known Turek benchmark case [44] was used to put the manifold mapping to the test. The results unfortunately were put to a stop before the essential dynamics of the structure came to existence. The log files of the fluid solvers showed inconsistencies in the Courant numbers, having consistently increasing values of the fine model, resulting in a very unstable simulation. Even if both fluid resolutions are chosen the same, the same unstable behaviour prevails. A distinct cause is not there as the preCICE library mentions the manifold mapping algorithm to be in an experimental state. Moreover, there are many different aspects such as the mapping between the fluid models themselves that

play a major part in all of the simulation. Henceforth further investigation is needed to validate the efficiency of the manifold mapping algorithm.

Chapter 9

Conclusions

At NRG, an advanced software package name NRG-FSIFOAM has been developed to solve Fluid-Structure Interaction (FSI) problems in nuclear reactors computationally. In the modern day, energy demands are increasing dramatically, making nuclear engineers develop more powerful reactors designs. Hence, adaptations such as the increase of coolant flow, applying a different coolant liquid or even reconsidering different structural components are considered. Before designing any reactor, much testing needs to done, therefore there is a need for accurate and efficient evaluation of FSI systems, to prevent unwanted events. The first set research objective thereupon is formulated as

Verify NRG-FSIFOAM's coupling and pressure fluctuation model for its accuracy compared with different (commercial) software.

To this end a single rod from a fuel assembly of a nuclear reactor core is singled out, and simulated under different circumstances, these being submerged by both water and a lead-bismuth eutectic (PbBi). Bare rod simulations are performed, and in order to get a simulation leaning more towards reality, also a wire-wrapped rod is subjected to these turbulent flows. At last, to test the capabilities of the wall-functions in NRG-FSIFOAM, that can be used to reduce overall efficiency, different wall-resolved ($y^+ \approx 1$) and wall-function ($y^+ \approx 60$) fluid meshes are put to the test. The exact same simulations were previously performed with the commercial software STAR-CCM+.

From the mesh-sensitivity analyses, both solid-only and FSI simulations, it can be concluded that the solid meshes copied from STAR-CCM+ (selected after a mesh-sensitivity of it's own) are not sufficiently refined in the current settings. Finer meshes showed a drastic decrease in the damping ratio and also a slight change in the modal frequencies. When using the same meshes, damping ratios are higher than obtained by STAR-CCM+. Possibly due to using the nearest-neighbour mapping, the accuracy is lost.

In general, for both the bare rod and wire-wrapped rod in water and PbBi, the wall-function results align better to the STAR-CCM+ results than the wall-resolved cases. A distinctive cause is not singled out, but more than likely both the difference in implemented wall-functions between the solvers and the known unstable behaviour of the $k-\omega$ SST model in OpenFOAM for wall-resolved cases play a major role in this. The wall-resolved vibrational parameters are defined by a much lower frequency in general.

NRG-FSIFOAM is able to capture the same interesting features in the rod behaviour that were found with STAR-CCM+. These include the fact that PbBi simulations have a higher portion of the rod movement being defined by inertia of the fluid (due to higher fluid density) and thus showing more accurate results in PbBi simulations, fitting closer to STAR-CCM+. Moreover, due to the wire-spacer, symmetry of the FSI simulation is lost, displaying that the wire-wrapped rod follows an elliptic-like trajectory pattern that tends to a final position different from the initial position. Finally it is remarked, that the Pressure Fluctuation Model (PFM) produces very similar results to the impulse-forced simulation results from NRG-FSIFOAM. It is thus very likely that if the standard U-RANS results are accurate, the same holds for the PFM results. However, this can also be a disadvantage if the opposite is the case. The small differences between the classical U-RANS and U-RANS with PFM models can possibly be improved by for example introducing anisotropic turbulence for the eddy length scales where isotropy is not a valid assumption.

The second aspect of testing nuclear reactor components, is to be able to achieve this within a reasonable time-frame. It is for this reason an additional research objective was stated as

Improve the efficiency of the fluid-structure interaction coupling method, by maintaining at the same time the stability and the accuracy of the method.

Within the framework of this thesis project, a new extrapolation has been developed that manually takes advantage of the variability in the displacement of the structure by means of a weighting parameter, making it convenient when running PFM simulations. The same impulsive force bare rod case in PbBi flow and PFM-forced wire-wrapped simulation in water have been used as test cases to evaluate the method. The stronger the coupling of the case, the better the new extrapolation method performs. In the first bare rod in PbBi case, the average amount of coupling iterations is around 4. The calculations of the weighting parameter overshadow the potentially close guess at the time-step. In the second wire-wrapped rod case, the average amount of iterations is around 13, and now much time is made up compared with the extrapolations of which it is based. However, at best, the new extrapolation performs equal as the originally applied extrapolation. Further investigation is needed to determine if an even more involved FSI case can make the new extrapolation the clear winner.

Besides extrapolation methods, the Manifold Mapping has been integrated in NRG-FSIFOAM to have a more efficient coupling method. The current status is very unstable however and has not delivered good results yet.

Chapter 10

Suggestions for future research

During the course of this Master thesis project many research opportunities were stumbled upon that could not be pursued due to either being to large for the size of the project, or either in order to not defect to much from the original research objectives. Hence, in this chapter, a short summary of suggestions for future research is given.

10.1 Radial basis functions as a mapping method in preCICE

In all of the simulation performed, the nearest-neighbour method is used to map the structure displacement to interface of the fluid domain and to map the fluid stresses to the interface of the structure. Newer versions of the preCICE library [8] have more stable implementation of parallel radial basis function mappings. These methods theoretically provide more accurate results.

10.2 Profound investigation of the $k-\omega$ SST model in NRG-FSIFOAM

From the bare rod and wire-wrapped rod simulations in chapter 7 a distinct winner between the examined software packages (NRG-FSIFOAM and STAR-CCM+) was not determined. Moreover, the results obtained with NRG-FSIFOAM showed more deviation than the ones from STAR-CCM+. From the discussion, the most likely cause of the differences are due to $k-\omega$ SST being put to use in both NRG-FSIFOAM and STAR-CCM+, either through unstable behaviour in OpenFOAM for wall-resolved meshes, or either through application of different wall-functions. There are two routes in which the research can be done:

- If possible, one can go through the source code files related to the $k-\omega$ SST models. As OpenFOAM is open-source software, this gives no trouble, STAR-CCM+ however is a commercial package, making it more difficult. In order to see if there is a large difference in the implemented wall-functions, one can use a simple benchmark case to validate this difference is prominently existing in basic problems.

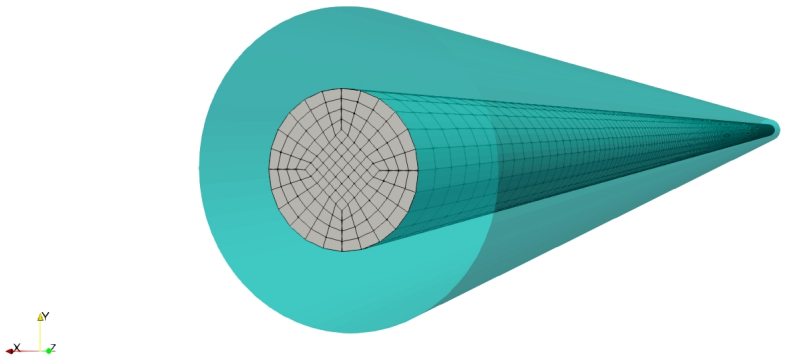


Figure 10.1: *Fluid-structure domain of a flexible brass beam in turbulent water [40].*

- As plowing through heaps of source codes is not the preferred method choice, a better choice would be to first compare NRG-FSIFOAM and STAR-CCM+ to an experimental case. Shams et al. [40] have performed a PFM simulation of a flexible brass beam (figure 10.1) in turbulent water of which experimental data is known. Through a similar study with a wall-function and wall-resolved fluid mesh, early results could possibly already rule out certain explanations for the difference between the software packages.

10.3 Further development of extrapolation methods

The exploration of a better extrapolation method tailored for fluid-structure interaction problems is a topic attempted as a primer within this project. As a consequence, very specific choices have been made, such as

- a single weighting parameter $\boldsymbol{\omega}^{n+1}$ having values $\omega_i^{n+1} \in [0, 1]$ in order to switch between to well-known extrapolation methods given in eq. (5.38);
- defining the weighting values on the basis of three data points, given in (5.39).

This gives much room for development. Perhaps a different definition for ω_i^{n+1} can be used on the basis of more/less previous values and guesses to have a better weighting on the variability in the system. Even fundamentally, also a completely different extrapolation can be chosen. In this work, better efficiency was strived for, but achieving better stability is also a great step towards a more evolved software package. A suggestion follows the trend of flux limiters in finite volume theory (see Koren [26]). Having the following continuous equation

$$\frac{\partial c}{\partial t} + \frac{\partial f(c)}{\partial x} = 0\,, \tag{10.1}$$

in which c represents a concentration and f is the flux of that concentration, and having a mesh as displayed in figure 10.2, one likes to satisfy the positive coefficients rule for the

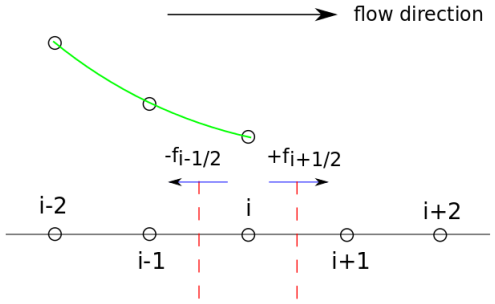


Figure 10.2: *General 1D finite volume mesh with flux f.*

node concentrations $c_i^{n+1} = c_i^{n+1}(\dots, c_{i-2}^n, c_{i-1}^n, c_i^n, c_{i+1}^n, c_{i+2}^n, \dots)$,

$$\dots, \frac{\partial c_i^{n+1}}{\partial c_{i-2}^n} \geq 0\,, \frac{\partial c_i^{n+1}}{\partial c_{i-1}^n} \geq 0\,, \frac{\partial c_i^{n+1}}{\partial c_i^n} \geq 0\,, \frac{\partial c_i^{n+1}}{\partial c_{i+1}^n} \geq 0\,, \frac{\partial c_i^{n+1}}{\partial c_{i+2}^n} \geq 0\,, \dots \quad , \tag{10.2}$$

which is natural for the convection and diffusion phenomena. A flux limiter ϕ is introduced in order to calculate the concentration on the wall of the finite volume as

$$c_{i+1/2}^n = c_i^n + \frac{1}{2}\phi(r_{i+1/2}^n)(c_i^n - c_{i-1}^n), \qquad \text{with } r_{i+1/2}^n := \frac{c_{i+1}^n - c_i^n}{c_i^n - c_{i-1}^n} \tag{10.3}$$

acting as smoothness ratio. This way one blends a non-positive (higher order) scheme with a positive (lower order scheme), thus resulting in a positive higher order scheme. The main advantage is that by translating this theory into the extrapolation methods, most of the obtained results can be directly put into application.

10.4 Stability of the Manifold Mapping

The manifold mapping (section 5.4) has been integrated in NRG-FSIFOAM and has been tested on the Turek benchmark case [44]. The simulations however prematurely stopped due to unstable behaviour. A distinct cause could not be determined as the preCICE library mentions the manifold mapping algorithm to be in an experimental state. Also, many aspects are involved in the algorithms where things might go wrong. Further investigation can identify the cause of instability. If so, much time can be won in the coupling of an FSI simulation.

Appendices

Appendix A

Fluid-only simulation of the bare rod

Before performing an FSI simulation to compare the dynamical behaviour of the rod immersed by a fluid, first the fluid domain needs to be initialized to a steady-state solution.

For all fluid domains considered (Figure 7.3), a static mesh fluid solver is applied for a duration of $t = 3.5$s with a time-step $\Delta t = 0.01$s. Solution values extracted from probes in the domain and residuals of the solvers are shown to visualize that a statistically steady system has been obtained. Four probes have been placed in all of the meshes on the positions in the table below, which are visualized in green on the $y^+ \approx 60$ mesh for water in Figure A.1.

	x	y	z
Probe1	0.0	0.01	0.35
Probe2	0.0	-0.01	0.35
Probe3	0.0	0.01	0.69
Probe4	0.0	-0.01	0.69

Table A.1: *Probes locations used for extraction of variables in the fluid domain*

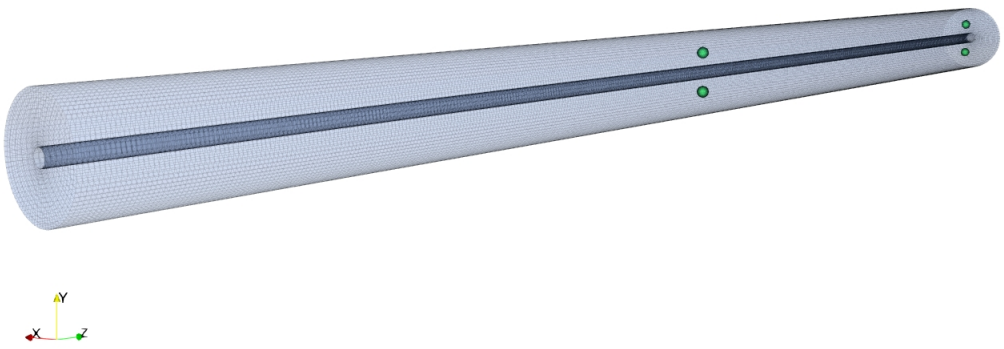


Figure A.1: *Probes (green) used for extraction of variables displayed on the $y^+ \approx 60$ mesh for water.*

Looking at the residuals of linear solvers in Figure A.2, convergence towards a statistically steady state is observed. Additionally strange jumps in the residuals of the turbulent

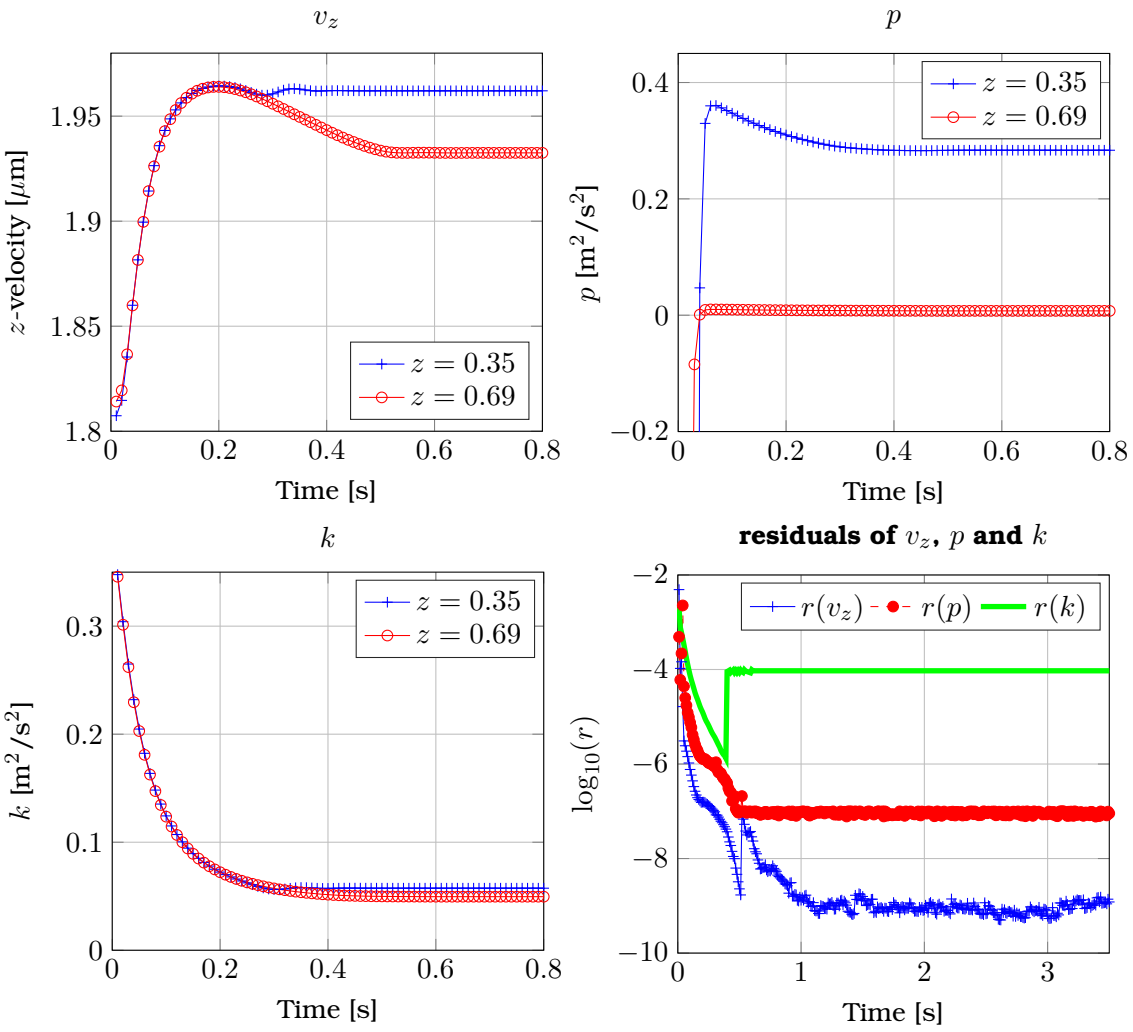


Figure A.2: *The velocity v_z, the pressure p and the turbulent kinetic energy k at the probe locations in table A.1 of a fluid-only simulation of water using a wall-function ($y^+ \approx 60$) mesh. Also the linear solver residuals of the three variables over time are displayed.*

kinetic energy k and the axial velocity v_z can be observed. To understand this, an understanding of the PIMPLE algorithm used by OpenFOAM is required. The PIMPLE algorithm is briefly explained in chapter 6.

In OpenFOAM for all the calculated variables, $\mathbf{v}, p$ and k (turbulent kinetic energy), a linear solver has to be specified together with tolerances that have to be achieved. For the velocity $\mathbf{v}$ this tolerance is set at 10^{-7} and for k at 10^{-4}. The jumps visible in Figure A.2 result from having achieved the tolerance criterion from the start of the time-step, and consequently not having to perform any linear solver iterations. The jump in $r(k)$ is at $t = 0.40$, whereas the jump in $r(v_z)$ is at $t = 0.52$. Moreover due to relaxation, the residuals are not consistent at each transition from solver to solver, and hence the jumps are present.

The PIMPLE algorithm itself also has a root mean square residual for the momentum equation to determine to stop the for loop or to keep correcting for more precise variables. This is a better quantity in general to notice convergence to steady-state of the fluid. In

Figure A.3 below, one can see that this indeed happens.

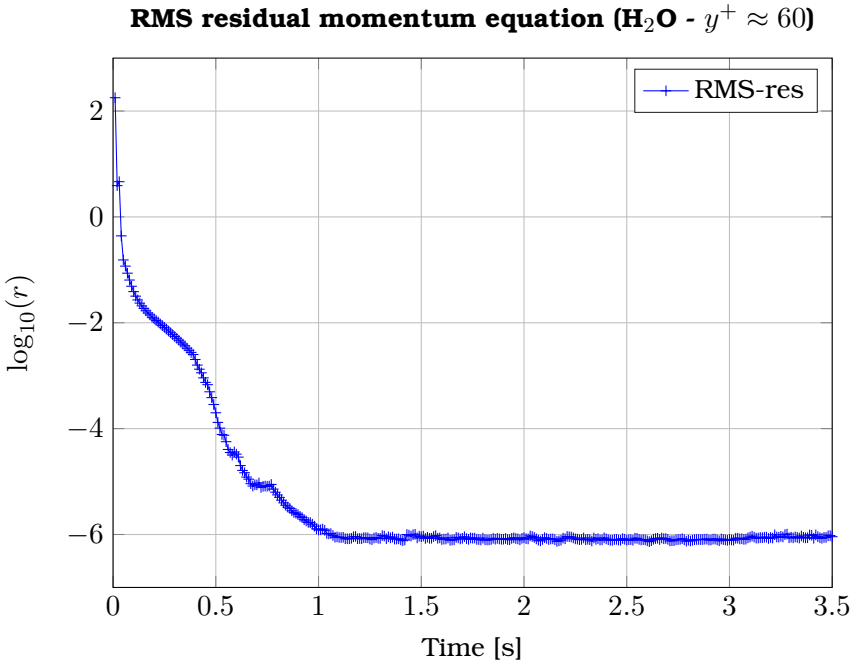


Figure A.3: *Root mean square residual of the momentum equation solved by the* `PIMPLE` *algorithm in OpenFOAM for a fluid only simulation of water on the* $y^+ \approx 60$ *mesh.*

Appendix B

Derivation of extrapolation method (5.29)

In this short appendix, the derivation of the extrapolation formula $\boldsymbol{d}_2^{n+1} = 3\boldsymbol{d}^n - 3\boldsymbol{d}^{n-1} + \boldsymbol{d}^{n-2}$ is given by means of Taylor series expansions. The approach is slightly different in context compared with the manner in which $\boldsymbol{d}_1^{n+1} = \frac{5}{2}\boldsymbol{d}^n - 2\boldsymbol{d}^{n-1} + \frac{1}{2}\boldsymbol{d}^{n-2}$ was derived in section 5.3. Instead of multiplying the Taylor expansions with values craftily chosen to obtain the necessary coefficients of the extrapolation $\boldsymbol{d}_1^{n+1}$, now, parameters α, β and γ are set to be the coefficients that need to be obtained to derive the highest-order accurate extrapolation taking three previous data points into account. The extrapolation to be sought thus is

$$\boldsymbol{d}^{n+1} = \alpha\boldsymbol{d}^n + \beta\boldsymbol{d}^{n-1} + \gamma\boldsymbol{d}^{n-2}\,. \tag{B.1}$$

Next, Taylor approximations with a perturbation of $\Delta t, 2\Delta t$ and $3\Delta t$ are filled in to obtain

$$\begin{aligned}\boldsymbol{d}^{n+1} = \quad & \alpha\left[\boldsymbol{d}^{n+1} - (\boldsymbol{d}')^{n+1}\Delta t + \tfrac{1}{2}(\boldsymbol{d}'')^{n+1}\Delta t^2 - \tfrac{1}{6}(\boldsymbol{d}''')^{n+1}\Delta t^3\right] \\ & +\beta\left[\boldsymbol{d}^{n+1} - 2(\boldsymbol{d}')^{n+1}\Delta t + 2(\boldsymbol{d}'')^{n+1}\Delta t^2 - \tfrac{4}{3}(\boldsymbol{d}''')^{n+1}\Delta t^3\right] \\ & +\gamma\left[\boldsymbol{d}^{n+1} - 3(\boldsymbol{d}')^{n+1}\Delta t + \tfrac{9}{2}(\boldsymbol{d}'')^{n+1}\Delta t^2 - \tfrac{9}{2}(\boldsymbol{d}''')^{n+1}\Delta t^3\right] + \mathcal{O}(\Delta t^4)\,.\end{aligned} \tag{B.2}$$

In order to derive $\boldsymbol{d}_2^{n+1}$, the following algebraic system must be solved

$$\begin{aligned}\alpha + \beta + \gamma &= 1 \\ -\alpha - 2\beta - 3\gamma &= 0 \\ \frac{1}{2}\alpha + 2\beta + \frac{9}{2}\gamma &= 0\end{aligned}\,, \tag{B.3}$$

that is rewritten in matrix notation and reduced by means of admissible row-operations as follows

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ -1 & -2 & -3 & 0 \\ \frac{1}{2} & 2 & \frac{9}{2} & 0 \end{array}\right] \xrightarrow[R_2|R_2+R_1]{R_3|-2R_3+R_1} \left[\begin{array}{ccc|c} 1 & 1 & 1 & 1 \\ 0 & -1 & -2 & 1 \\ 0 & -3 & -8 & 1 \end{array}\right] \xrightarrow[R_1|R_1+R_2]{R_3|R_3-3R_2} \left[\begin{array}{ccc|c} 1 & 0 & -1 & 2 \\ 0 & -1 & -2 & 1 \\ 0 & 0 & -2 & -2 \end{array}\right]\,.$$

From these last equations, it can be easily deducted that $\alpha = 3$, $\beta = -3$ and $\gamma = 1$. In other words, the extrapolation method $\boldsymbol{d}_2^{n+1}$ is the highest-order accurate extrapolation in time that can be obtained under the restriction of using solely three previous time-steps.

Bibliography

[1] **Anonymous:** *Nuclear power reactors*, World nuclear association, retrieved September 1st, 2018, from http://www.world-nuclear.org/information-library/nuclear-fuel-cycle/nuclear-power-reactors/nuclear-power-reactors.aspx

[2] **Anonymous:** *Nuclei: applications of nuclear fission*, Review of the universe, retrieved September 2nd, 2018, from https://universe-review.ca/F14-nucleus05.htm

[3] **Anonymous:** *World energy needs and nuclear power*, World nuclear association, retrieved September 3rd, 2018, from http://www.world-nuclear.org/information-library/current-and-future-generation/world-energy-needs-and-nuclear-power.aspx

[4] **W. Bangerth, D. Davydov, T. Heister, L. Heltai, G. Kanschat, M. Kronbichler, M. Maier, B. Turcksin and D. Wells:** *The `deal.II` Library, Version 8.4*, in: Journal of Numerical Mathematics, Vol. 24 (2016), pp. 135-141

[5] **J. W. Banks; W. D. Henshaw and D. W. Schwendeman:** *An analysis of a new stable partitioned algorithm for FSI problems. Part I: Incompressible flow and elastic solids*, in: Journal of Computational Physics, Vol. 269 (2014), pp. 108-137

[6] **D. S. Blom; A. H. van Zuijlen and H. Bijl:** *Multi-level acceleration with manifold mapping of strongly coupled partitioned fluid-structure interaction*, in: Computer Methods in Applied Mechanics and Engineering, Vol. 296 (2015), pp. 211-231

[7] **R. Boman:** *Développement d'un formalisme Arbitraire Lagrangien Eulérien tridimensionnel en dynamique implicite. Application aux opérations de mise à forme*, Université de Liège, retrieved September 21st, 2018, from http://bictel.ulg.ac.be/ETD-db/collection/available/ULgetd-05172010-091736/

[8] **H.-J. Bungartz; F. Lindner; B. Gatzhammer; M. Mehl; K. Scheufele; A. Shukaev and B. Uekermann:** *preCICE - A fully parallel library for multi-physics surface coupling*, in: Computers & Fluids, Vol. 141 (2016), pp. 250-258

[9] **P. Causin; J. F. Gerbeau and F. Nobile:** *Added-mass effect in the design of partitioned algorithms for fluid-structure problems*, in: Computer Methods in Applied Mechanics and Engineering, Vol. 194 (2005, pp. 4506-4527

[10] **A. de Boer; M. S. van der Schoot and H. Bijl:** *Mesh deformation based on radial basis function interpolation*, in: Computers and Structures, Vol. 85 (2007), pp. 784-795

[11] **J. Degroote:** *Patitioned simulations of fluid-structure interaction*, in: Archives of Computational Methods in Engineering, Vol. 20 (2013), pp. 185-238

[12] **D. De Santis and A. Shams:** *Numerical study of flow-induced vibration in a wire wrapped fuel assembly*, 17th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-17) Qujiang Int'l Conference Center Xi'an, September 3-8, 2017

[13] **D. De Santis and A. Shams:**, *Analysis of flow induced vibrations and static deformations of fuel rods considering the effects of wire spacers and working fluids*, under revision 2018

[14] **J. Donea; A. Huerta; J.-Ph. Ponthot and A. Rodríguez-Ferran:** *Arbitrary Lagrangian-Eulerian Methods, Chapter 14*, in: Encyclopedia of Computational Mechanics, John Wiley & Sons, Ltd (2004), pp. 413-437

[15] **D. Echeverría and P. W. Hemker:** *Manifold mapping: a two-level optimization technique*, in: Computing and Visualization in Science, Vol. 11 (2008), pp. 193-206

[16] **M. Gill; F. Livens and A. Peakman:** *Chapter 9 - Nuclear Fission*, in: Future Energy, Elsevier (2014), pp. 181-198

[17] **K. Gramoll:** *Mechanics - theory*, eCourses, retrieved September 20th, 2018, from https://www.ecourses.ou.edu/cgi-bin/ebook.cgi?doc=&topic=me&chap_sec=09.2&page=theory

[18] **T. Holzmann:** *Mathematics, numerics, derivations and OpenFOAM®*, Holzmann CFD (2016), 166 pages

[19] **IAEA:** *Operational & long-term shutdown reactors*, retrieved September 2nd, 2018, from https://pris.iaea.org/PRIS/WorldStatistics/OperationalReactorsByType.aspx

[20] **R. I. Issa:** *Solution of the implicitly discretised fluid flow equations by operator-splitting*, in: Journal of Computational Physics, Vol. 62 (1986), pp. 40-65

[21] **W. P. Jones and B. E. Launder:** *The prediction of laminarization with a two-equation model of turbulence*, in: International Journal of Heat and Mass Transfer, Vol. 15 (1972), pp. 301-314

[22] **S. Kaneko; T. Nakamura; F. Inada; M. Kato and N. W. Mureithi:** *Flow-Induced Vibrations - classifications and lessons from practical experiences*, Elsevier (2008), 310 pages

[23] **M. Kim; P. Hughes and R. A. Ainsworth:** *An overview of IAEA technical guidelines on Fluid-Structure Interaction*, in: Proceedings of the ASME 2014 Pressure Vessels and Piping Conference, Vol. 4 (2014), PVP2014-29109, 4 pages

[24] **Y. Kitsunai; H. Kobayashi:** *Leakage of primary coolant at Mihama Unit 2 due to failure of SG tube*, retrieved August 25th, 2018, from http://www.shippai.org/fkd/en/cfen/CB1061010.html

[25] **S. Kondo:** *Lessons learned for PSA from the SGTR incident at Mihama, unit 2, in 1991*, in: Reliability Engineering and System Safety, Vol. 45 (1994), pp. 57-65

[26] **B. Koren:** *A robust upwind discretization method for advection, diffusion and source terms*, in: Department of Numerical Mathematics [NM], No. R 9308 (1993)

[27] **S. Kottapalli:** *Numerical prediction of flow induced vibrations in nuclear reactors through pressure fluctuation modeling*, Technische Universiteit Delft (2016), 60 pages

[28] **F. Liu:** *A thorough description of how wall functions are implemented in OpenFOAM*, Chalmers University of Technology (2017), retrieved November 2nd, 2018, from http://www.tfd.chalmers.se/~hani/kurser/OS_CFD_2016/FangqingLiu/openfoamFinal.pdf

[29] **W. Marcum and B. I. Spinrad:** *Nuclear reactor*, Encyclopædia Britannica, inc., retrieved September 1st, 2018 from https://www.britannica.com/technology/nuclear-reactor/Introduction (p. 1), https://www.britannica.com/technology/nuclear-reactor/Thermal-intermediate-and-fast-reactors (p. 3) and https://www.britannica.com/technology/nuclear-reactor/Fuel-types (p. 4)

[30] **F. R. Menter:** *Two-equation eddy-viscosity turbulence models for engineering applications*, in AIAA Journal, Vol. 32 (1994), pp. 1598-1605

[31] **J. M. McDonough:** *Introductory lectures on turbulence,* Departments of Mechanical Engineering and Mathematics, University of Kentucky, retrieved November 1st, 2018, from http://web.engr.uky.edu/~acfd/lctr-notes634.pdf

[32] **Mitsubishi Nuclear Fuel:** *Manufacturing process in Tokai plant (PWR Fuel)*, retrieved September 3rd, 2018, from http://www.mnf.co.jp/en/business/process.html

[33] **A. Miyakawa; H. Maeda; Y. Kani and K. Ito:** *Sodium leakage experience at the prototype FBR Monju*, in: Proceedings of a technical committee meeting on unusual occurrences during LMFR operation, IAEA-TECDOC–1180 (2000)

[34] **OpenFOAM:** *User Guide, Version 2.2.0*, OpenFOAM Foundation, 2013

[35] **S. V. Patankar and D. B. Spalding:** *A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows*, in: International Journal of Heat andMass Transfer, Vol. 15 (1972), pp. 1787-1806

[36] **M. J. Pettigrew; C. E. Taylor, N. J. Fisher, M. Yetisir and B. A. W. Smith:** *Flow-induced vibration: recent findings and open questions*, in: Nuclear Engineering and Design, Vol. 185 (1998), pp. 249-276

[37] **Y. Saad:** *Numerical methods for large eigenvalue problems*, 2nd Edition, Society for Industrial and Applied Mathematics (2011), 276 pages

[38] **S. Sachs; M. Streitenberger; D. C. Sternel and M. Schäfer:** *Extrapolation methods for accelerating unsteady partitioned fluid-structure interaction simulations*, in: International Journal of Multiphysics, Vol. 5 (2012), pp. 287-297

[39] **R. Schaback:** *A practical guide to radial basis functions*, Universität Göttingen, teaching notes (2007), retrieved September 2nd, 2018, from http://num.math.uni-goettingen.de/schaback/teaching/sc.pdf

[40] **A. Shams; S. Kottapalli; A. H. van Zuijlen and M. J. B. M. Pourquie:** *Numerical simulation of turbulence induced vibrations with U-RANS using the pressure fluctuation model*, in: Journal of Fluids and Structures, under revision 2018

[41] **S. Sharma:** *Numerical study of turbulence induced vibrations using synthetic fluctuation field modeling in nuclear reactor applications*, Technische Universiteit Delft (2018), 73 pages

[42] **S. Takahashi; K. Hasegawa and T. Nakamura:** *Nuclear power plant issues in IAEA technical guidelines on Fluid-Structure Interactions*, in: Proceedings of the ASME 2014 Pressure Vessels and Piping Conference, Vol. 4 (2014), PVP2014-28476, 7 pages

[43] **E. ter Hofstede:** *Numerical study of fluid structure interaction in nuclear reactor applications*, Technische Universiteit Delft (2015), 64 pages

[44] **S. Turek and J. Hron:** *Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow*, in: Fluid-Structure Interaction, Springer (2006), pp. 371-385

[45] **B. Uekermann:** *Partitioned Fluid-Structure Interaction on Massively Parallel Systems*, PhD Thesis, Technische Universität München (2016)

[46] **D. S. Weaver; S. Ziada; M. K. Au-Yang; S. S. Chen; M. P. Païdoussis and M. J. Pettigrew:** *Flow-induced vibrations in power and process plant components—progress and prospects*, in: Journal of Pressure Vessel Technology, Vol. 122 (2000), 10 pages

[47] **D. C. Wilcox:** *Reassessment of the scale-determining equation for advanced turbulence models*, in: AIAA Journal, Vol. 26 (1988), pp. 1299-1310

Abbreviations

ALE	Arbitrary Lagrangian-Eulerian
AVB	Anti-Vibration Bar
B.C.	Boundary Condition
BWR	Boiling Water Reactor
CFD	Computational Fluid Dynamics
CSM	Computational Solid Mechanics
DN	Dirichlet-Neumann
DNS	Direct Numerical Simulation
DOF	Degree Of Freedom
FBR	Fast Breeder Reactor
FDM	Finite Difference Method
FEM	Finite Element Method
FFT	Fast Fourier Transformation
FIV	Flow-Induced Vibrations
FSI	Fluid-Structure Interaction
FVM	Finite Volume Method
GFR	Gas-cooled Fast Reactor
I.C.	Initial Condition
IQN-ILS	Interface Quasi-Newton algorithm with an approximation for the Inverse of the Jacobian from a Least-Squares model
JAEA	Japan Atomic Energy Agency
JNC	Japan Nuclear Cycle development institute
LES	Large Eddy Simulation
LFR	Lead-cooled Fast Reactor
LMFR	Liquid Metal-cooled Fast Reactor
LWR	Light Water Reactor
MM	Manifold Mapping
MSR	Molten Salt Reactor
NRG	Nuclear Research and consultancy Group
PbBi	Liquid-Bismuth eutectic
PDE	Partial Differential Equation
PFM	Pressure Fluctuation Model
PHWR	Pressurized Heavy-Water Reactor
`PIMPLE`	merged `SIMPLE-PISO`
`PISO`	Pressure-Implicit-Split-Operator

PWR	Pressurized Water Reactor
RBF	Radial Basis Function
SCWR	SuperCritical Water-cooled Reactor
SFR	Sodium-cooled Fast Reactor
SIMPLE	Semi-Implicit-Method-Of-Pressure-Linked-Equations
STA	Science and Technology Agency
SVD	Singular Value Decomposition
TIV	Turbulence-Induced Vibrations
TU/e	Eindhoven University of Technology
U-RANS	Unsteady Reynolds Averaged Navier-Stokes
VHTR	Very High Temperature Reactor