NRG-beamFoam: a simplified numerical FSI solver developed in OpenFOAM

for modelling axial-flow in nuclear reactors

George Ștefan Popi

TU Delft

NRG

NRG-beamFoam: a simplified numerical FSI solver developed in OpenFOAM

for modelling axial-flow in nuclear reactors

by

George Ștefan Popi

to obtain the degree of Master of Science at the Delft University of Technology, to be defended publicly on Monday February 19, 2024 at 2:00 PM.

Student number:	5625564	
Thesis committee:	Dr. ir. M.I. Gerritsma	TU Delft, Chair
	Dr. Ing. S.G.P. Castro	TU Delft, Examiner
	Dr. ir. A.H. van Zuijlen	TU Delft, Supervisor
	Ir. K. Zwijsen	NRG, Supervisor

Cover: Damien Jemison

Preface

On a personal level, my graduation from a Master's in Aerospace Engineering at TU Delft feels like the culmination of my efforts in the last 8 years, ever since I decided to become an engineer while still in high school. Since then, I have always tried to set higher standards for myself and to rise to the technical challenges that lay in front of me. This thesis feels like one of the rare occasions where I feel unapologetically happy with the outcome.

First of all, thanks to Sander and Kevin for their calm guidance and continuous availability throughout the project. I found our meetings to be interesting and reenergizing, and without them, this thesis would probably have less than a quarter of the results that it has. I would also like to send my special thanks to Edo, without whom the OpenFOAM implementation would have simply been impossible. The only way to repay all of the coding help I received from him was to cause some amusement here and there due to my whimsical style of coding, but somehow he never seemed to be bothered about it. I am also grateful for the help that I received from Sina and André in setting up the FSI cases and in using the AniPFM model. Thus, this work should be seen as a communal effort of a small team of people, rather than an individual study.

Apart from the professional help that I received throughout the thesis, I was also lucky enough to meet a bunch of nice, fun, and supportive people in my personal life. I would like to thank Aron and Georgia, who witnessed my ups and downs throughout the year, and never gave up on me. I would also like to thank Avyadhish and Rakshith for offering your companionship (and nice food) throughout the tough first year, and for allowing me to discover a new and rich culture. I would also like to acknowledge Jochem, my main brother-in-arms for group projects and beyond, without whom life in the Netherlands would feel much more foreign. Finally, I would like to thank Isabelle and the rest of the Alkmaar group (Jelle, Dennis, Alex, Aida, Gilliam, Natalie) for all of the nice experiences that we had in the last year.

George Ștefan Popi
Alkmaar, Febuary 2024

Abstract

Nuclear power is increasingly seen as a potential solution to the decarbonization of the energy sector in the coming decades. However, one of the main causes of downtimes for current-generation nuclear reactors is Grid-To-Rod-Fretting (GTRF), which is a multiphysics phenomenon. On one hand, the cooling axial water flow inside Pressurized-Water Reactors (PWRs) causes the Turbulence-Induced Vibration (TIV) of the slender fuel elements found in the nuclear reactor cores. This dynamic Fluid-Structure Interation (FSI) phenomenon, combined with the high operating temperatures, and the irradiation of the different components contribute to the creep behaviour of the fuel rods, and potentially to their cracking. Should this occur, the operation of the reactor is immediately stopped, and expensive maintenance work is carried out.

To quantify the amplitudes of the TIV due to axial flow in nuclear reactors, before this study, NRG developed a two-way coupled FSI workflow called NRG-FSIFOAM, within which the fluid behaviour was predicted using the computationally efficient AniPFM synthetic turbulence model, developed in OpenFOAM. The structural solver had been developed separately in deal.II, and it consisted of using 3D Lagrange elements for a Finite Element Method (FEM) discretization. The two solvers were coupled using preCICE, which handled both the mapping between the two domains, as well as the subiteration that was required for strong coupling. Nevertheless, it was believed that the complexity of the structural model could be decreased, given the slenderness of the fuel elements, the predominant bending imposed by the fluid forcing, and their rigidity to shearing. In turn, using a computationally efficient structural model would accommodate shorter computation times, as well as the possibility to numerically simulate more complex geometries, thereby increasing the accuracy of the results.

To reduce the computational costs of the structural solver, this thesis initially proposes a 1D beam-element FEM solver. This structural representation is further simplified using a Mode Acceleration (MA) Reduced Order Model (ROM). This method is based on determining the nodal FEM displacements using the superposition of a user-inputted number of structural eigenmodes. Thus, the degrees of freedom for the structural problem are the amplitudes that one should apply to the reduced eigenbasis to best represent the current deformation. A quasi-static correction is subsequently applied to the solution, by assuming that the inertial effects are fully contained by the considered eigenmodes.

To take advantage of the low dimensionality of this new formulation, new mapping routines are also proposed, in the form of a load interpolator (mapping the concentrated fluid cell face forces on the FSI boundary to the structural nodes), and a displacement interpolator (transferring the displacements obtained at the structural FEM nodes to the nodes of the fluid mesh on the FSI boundary). The load interpolator is based on a conservative technique of transforming the distributed loads along each beam element to concentrated nodal loads for the FEM grid. It is therefore proposed that each beam element be subdivided into multiple segments. For each of these, the distributed loads are assumed to be continuous. The displacement interpolator makes use of the FEM modelling assumptions, and it uses the values of the shape functions along the beam elements to determine the local deformations. The strong coupling between the fluid and the structural domain is resolved using the Aitken subiteration algorithm. To avoid the computational overhead of using multiple separate libraries that was present in NRG-FSIFOAM, for the newly proposed workflow, the structural solver and the associated mapping routines are implemented directly in OpenFOAM. More specifically, all of the modules are contained within a mesh motion solver boundary condition. The mesh motion solver is then called at each outer corrector of the PIMPLE algorithm. For the remainder of this work, the combination of all of the features described above is called NRG-beamFoam.

Starting from the proposed framework, first, the individual verification of the different components was conducted. It is shown that for a simplified fuel element geometry the FEM solver offers accurate estimations of the eigenmodes and eigenfrequencies, and that the dynamic solution converges to the expected analytical solution when decreasing the timestep size. The load interpolator and the displacement interpolator also

exhibited convergent behaviour with respect to the number of segments and elements, respectively.

Having proven the expected functionality of the FEM solver and the associated mapping routines, the Aitken subiteration algorithm is also verified by using the entire workflow created by the three previous modules. It is found that the FSI convergence is significantly improved when compared to a fixed Under Relaxation Factor (URF) scheme. The NRG-beamFoam methodology is subsequently compared with NRG-FSIFOAM for FSI where a laminar fluid model is used to reproduce the experiment of Chen and Wambsganss [1]. This consists of an annular geometry, where the inner rod is flexible, and the outer one is fully rigid. The inner rod is initially deformed, and subsequently allowed to vibrate as water flows axially at 10 m/s. It is found that the frequency and damping computed for the motion using the two workflows have a relative difference that is lower than 1.5%.

After the laminar FSI simulation, an Unsteady Reynolds-Averaged Navier-Stokes (URANS) FSI simulation is conducted, to observe whether the turbulent stresses are correctly accounted for by the implementation proposed herein. It is found that the relative matching with NRG-FSIFOAM is now within 0.5%. At the same time, the computational cost of using NRG-beamFoam is 6% lower per FSI subiteration. Nevertheless, the total number of FSI subiterations executed by NRG-beamFoam is larger, given the superior IQN-ILS algorithm used by NRG-FSIFOAM, and its capacity to control the number of subiterations using the FSI residual obtained for the current subiteration. Thus, the subiteration algorithm represents one of the main measures that are recommended for future optimization of the computational costs of the methodology in this thesis.

Further, the implementation in this thesis is coupled with the AniPFM fluid model. It is shown that NRG-beamFoam has the advantage of only calling the AniPFM routines once per timestep, without significant changes in the accuracy. This represents an operational advantage over NRG-FSIFOAM, which must use the computationally expensive AniPFM module at every subiteration. Further, it is found that if the segments' nodes are axially non-matching with the fluid nodes on the FSI boundary, the load interpolator produces noticeable variations between the total forcing imposed by the fluid mesh and that which is mapped to the structural nodes. What's more, the load mapping appears to be dependent on round-off errors through the way that the connectivity matrix is computed between the fluid cell face centers and the segments' nodes. Thus, an improvement of the work herein refers to implementing an improved version of the load interpolator, where the redistribution of the loads better accounts for the numerical variation in the spatial location of the cell faces, and the boundaries between adjacent segments. For the remainder of the thesis, these inconveniences were mitigated by only using matching segments, as well as other simplified load interpolator approaches.

Even if the load interpolator issues were mitigated, FSI simulations combining NRG-beamFoam and AniPFM appear to crash prematurely. No clear correlation between the structural discretization and the stability is found. Possible identified causes are the random generator inside the AniPFM model, the incorrect computation of the rotations within NRG-beamFoam, or the suboptimal discretization schemes used by AniPFM. Further research is required to identify the root cause of this behaviour.

Finally, the ROM was applied to the FEM formulation. It is found that for laminar FSI simulations, a single degree of freedom is enough for accurately computing the rod's damped vibration. What's more, the MA exhibits the capacity of spatially and dynamically reconstructing responses that are associated with eigenmodes that are slightly outside of the reduced eigenbasis. When combined with the AniPFM model, the ROM model proposed in this work appears to only require 8 degrees of freedom per bending direction to offer a good estimation of the rod's deformed state at all time instances. Nevertheless, the same stability issues are identified as for the FEM solver. What's more, slight discrepancies are observed between the FEM solution and the ROM one when using the same total number of degrees of freedom. A possible hypothesis for this is the incorrect scaling of the eigenmodes of the structure, and it is recommended that this is assessed in future research.

Contents

List of Figures

List of Tables

Nomenclature

List of Abbreviations

ALE	Arbitrary Lagrange-Euler
AniPFM	Anisotropic Pressure Fluctuation Model
BC	Boundary Condition
BT	Balanced Truncation
CFD	Computational Fluid Dynamics
DNS	Direct Numerical Simulation
FA	Fuel Assembly
FEM	Finite Element Method
FIV	Flow-Induced Vibrations
FSI	Fluid-Structure Interaction
FVM	Finite Volume Method
GCL	Geometrical Conservation Law
GTRF	Grid-To-Rod-Fretting
IQN-ILS	Inverse Quasi-Newton - Inverse Least-Squares
LDRV	Load Dependent Ritz Vectors
LES	Large Eddy Simulation
MA	Mode Acceleration
MD	Mode Displacement
MIMO	Multiple Input Multiple Output
MTA	Modal Truncation Augmentation
OC	Outer Correctors
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PFM	Pressure Fluctuation Model
PISO	Pressure-Implicit-of-Split-Operations
POD	Proper Orthogonal Decomposition
PWR	Pressurised Water Reactor
RANS	Reynolds-Averaged Navier-Stokes
RMS	Root-Mean Square
ROM	Reduced-Order Model
SIMPLE	Semi-Implicit-Method-Of-Pressure-Linked-Equations
SVD	Singular Value Decomposition
TIV	Turbulence-Induced Vibrations
URANS	Unsteady Reynolds-Averaged Navier-Stokes
URF	Under-Relaxation Factor

List of Greek Symbols

α	Displacement constant for the Newmark time integration scheme
$\boldsymbol{\epsilon_s}$	Structural strains
$\boldsymbol{\mu}_\mathrm{t}$	Eddy viscosity
$\boldsymbol{\Phi_s}$	Eigenmode reduction basis
$\boldsymbol{\phi_s}$	Structural eigenmode
$\boldsymbol{\sigma_t}$	Turbulent fluctuation direction vector
$\boldsymbol{\tau_f}$	Effective fluid stress tensor
χ	Error
Δ^k	FSI residual convergence criterion
δ_{ij}	Kronecker delta
γ	Velocity constant for the Newmark time integration scheme
κ	Discretization ratio between the structural and the fluid domain
ω	Specific turbulence dissipation rate
ω_s	Structural circular eigenfrequency
ω_UR	FSI coupling Under-Relaxation Factor
ϕ_t	Phase angle of the turbulent fluctuations
ρ	Density
$\sigma_\mathbf{s}$	Structural stresses
ε_t	Turbulent dissipation
φ_s	Structural rotations
ξ_s	Modal damping ratio

Operators

$\delta \bullet$	Virtual quantity
$\bar{\bullet}$	Averaging operator
$A(\bullet)$	Amplitude of dynamic quantity

List of Roman Symbols

$\boldsymbol{a_s}$	Generalized modal coordinates
A	Area
$\boldsymbol{B}$	Differentiation matrix
$\boldsymbol{d}$	Structural displacements on the FSI boundary
D	Rod diameter
E	Young's Modulus
F	Fluid solver operator
f	Frequency
F	Nodal shear force
$\boldsymbol{f_s}$	Global external forcing vector
f_B	Externally applied nodal force
$\boldsymbol{H}$	Transfer function matrix
I	Bending moment of inertia
k	Turbulent kinetic energy
$\boldsymbol{K_s}$	Global stiffness matrix
$\boldsymbol{k_t}$	Turbulent fluctuation wavenumber vector
k_t	Turbulent fluctuation wavenumber
$\boldsymbol{L}$	The direction vector of the rod's axis
L	Rod length
l_t	Turbulent length scale
M	(Nodal) Bending moment
$\boldsymbol{M_s}$	Global mass matrix
$\boldsymbol{N}$	Shape function
$\boldsymbol{n}$	Normal
N_{FEM}	Number of degrees of freedom of the FEM system
N_{loops}	Number of FSI loops executed at each timestep
N_{modes}	Number of modes
N_{ROM}	Number of degrees of freedom of the reduced system
p	Fluid pressure
Q	Shear force
q_s	External structural distributed load
q_t	Scaled turbulent energy
$\boldsymbol{R}$	Reynolds Stress Tensor
$\boldsymbol{r}$	FSI residual vector
$\boldsymbol{r_{beam}}$	Position vector with respect to the axis of the rod in the beam reference system
$\boldsymbol{R_{rot}}$	Rotation matrix between the OpenFOAM and the beam reference system
$\boldsymbol{R_{rot,y}}$	Rotation matrix around the y axis
$\boldsymbol{R_{rot,z}}$	Rotation matrix around the z axis
S	Structural solver operator
$\boldsymbol{S}$	Fluid strain rate
T	Vibration period
T_{pass}	Average flow pass-time
$\boldsymbol{U_f}$	Fluid velocities
U_{axial}	Axial flow velocity
$\boldsymbol{u_s}$	Structural displacements
$\mathbf{u_t}$	Anisotropic velocity fluctuations
V	Volume
$\boldsymbol{V_m}$	Mesh motion velocity
$\boldsymbol{V_{ROM}}$	ROM projection matrix
$\mathbf{v_t}$	Dimensional velocity fluctuations
$\boldsymbol{w_t}$	Non-dimensional velocity fluctuations
$\boldsymbol{X_f}$	Fluid displacements
$\boldsymbol{z}$	Reduced space coordinates

Subscripts

beam	Referring or belonging to the beam reference system
cont	Referring or belonging to the body volume of the solid continuum
el	Referring or belonging to a beam FEM element
f	Referring or belonging to the fluid cell faces
f	Referring or belonging to the fluid domain

fit	Referring or belonging to a curve fitting procedure
IS	Referring or belonging to the IQN-ILS algorithm
MA	Referring or belonging to the Mode Acceleration ROM technique
MTA	Referring or belonging to the Mode Truncation Augmentation ROM technique
OF	Referring or belonging to the OpenFOAM reference system
red	Referring or belonging to a reduced Singular Value Decomposition representation of a matrix
Ritz	Referring or belonging to the Ritz method
ROM	Referring or belonging to the Reduced Order Modelling technique
s	Referring or belonging to the structural domain

Superscripts

$\bullet'$	Fluctuating part
XY	Referring to the XY bending plane
XZ	Referring to the XZ bending plane

1

Introduction

The generation of energy currently represents one of the main contributors to the greenhouse emissions done by human activity. For example, in 2010, 41% of man-related carbon dioxide (CO_2) emissions were due to the energy industry [2, 3]. At the same time, the total energy demand around the globe is expected to increase in 2040 by half when compared to the 2021 levels [4]. Despite this, to limit the increase in the average global temperature to a sustainable value for long term thriving, the Paris Agreement initially set the objective of zero carbon emissions for the energy industry by 2050 [5]. Though this initial target was softened, it currently still stipulates that the energy sector should reduce its current carbon footprint to a third by then [6], while still meeting a demand that will be more than 1.5 times higher. This suggests the need to develop and promote the use of energy-generating techniques that are more environmentally sustainable than the ones most commonly used today, such as coal and gas.

Nuclear energy represents a strong candidate to meet the green energy goals of the planet. This technology has lower or comparable lifecycle CO_2 emissions to alternatives such as solar, hydropower, or wind energy [7], while also offering the most stable energy output [8], and incurring the least human losses per generated energy unit [9, 10]. Despite those benefits, one of the main reasons for which nuclear energy is not used more intensively is the heavily polarized and fluctuating public popularity. The main concerns are related to potential nuclear accidents such as Fukushima or Chernobyl [11], and to the improper waste disposal of radioactive material [12]. Hence, the safety of nuclear reactors is of paramount importance for the popular adoption of the technology.

The main malfunction of currently operating reactors is caused by the phenomenon of Grid-To-Rod-Fretting (GTRF). 58% of the outages of Pressurized Water Reactors (PWRs) between 2011 and 2015 were attributed to GTRF, with research for its mitigation still being carried out, for example within the GO-VIKING joint European project [13]. This phenomenon is predominantly observed for nuclear reactors using PWR technology, wherein the nuclear fuel is stored in long, slender beams, which are cooled during the fission reaction using axially-flowing water at high pressures (approximately 150 bar [14]). In every reactor, hundreds of such fuel elements can be found, closely bundled next to one another into Fuel Assemblies (FAs). Multiple such fuel assemblies are routinely used within PWRs. A simplified representation of a conventional PWR (left) and of the components of a FA (right) is given in Figure 1.1. GTRF represents a multi-physics phenomenon. On one hand, the turbulent water flow excites the rods within the FA, a phenomenon that is known as Turbulence-Induced Vibration (TIV). In turn, this leads to mechanical contact between the fuel elements and the spacer grids, where the purpose of the latter is to prevent neighboring rods from touching each other, and to increase the rods' rigidity. Combined with the high operating temperatures, usually of about 300°C [15], the fuel rods become prone to creep, and eventually to cracking. The propagation of the crack inside the fuel rods can lead to the escape of radioactive material inside the cooling water flow. At this point, the operation of PWRs is automatically stopped by sensors, and the damaged fuel rods need to be replaced. A visual representation of the creep effects (left) and of the onset of GTRF (right) can be seen in Figure 1.2. The removal of the damaged fuel elements leads to high maintenance costs, and to lower energy yields of the nuclear reactors, as the entire powerplant needs to be shut down.

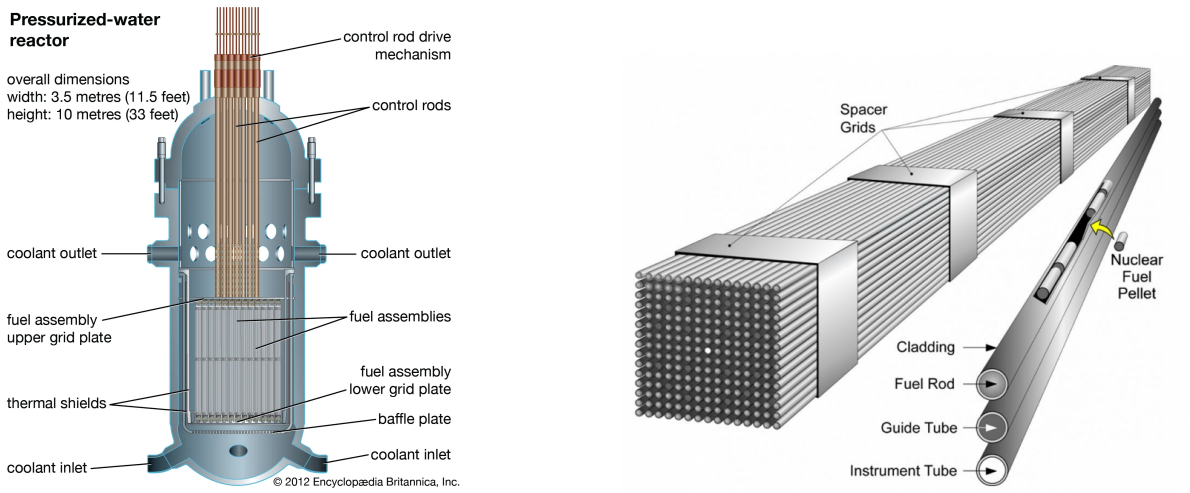


Figure 1.1: Left: Schematic sketch of a PWR (taken from [30]); right: isolated view of a fuel assembly within a PWR (taken from [31])

Numerous studies have been conducted to create designs that are resistant to GTRF. Many of the methodologies to do so were either experimental [16–19], or were centered around using simplified analytical models [20–22]. However, the former are limited by the difficulty in representing the operational conditions (high temperature, irradiation) and the dimensions of a PWR chamber (approximately 4 meters in length), as well as the influence of the operational time over, for example, the creep behaviour. The analytical models are limited by the need to use a fluid model that can accurately represent the excitation imposed on the rods, as well as the complex phenomena that occur at bundle level, such as hydrodynamic coupling [23–27], or added stiffness [28, 29] that will be detailed in Subsection 1.1.2.

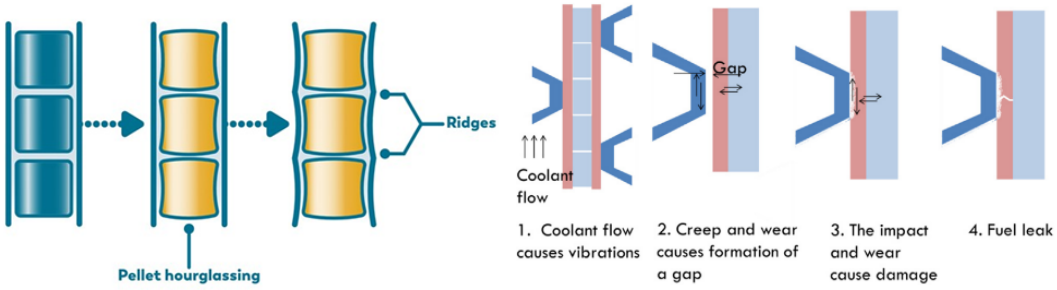


Figure 1.2: Left: displacement evolution of the fuel rods due to creep (taken from [32]); right: schematic of the fretting mechanism (taken from [33])

A possible alternative to these approaches is represented by the numerical simulation of bundled configurations. The main limitation of doing so, however, is the the prohibitively large computational cost. It is commonly known that in computing fluid flows, one of three approaches can be used. These are differentiated based on how wide the computed band within the turbulence spectrum is. The most accurate approach, Direct Numerical Simulation (DNS) computes the effect of all turbulent scales, including the smallest ones, known as the Kolmogorov eddies [34]. Due to its fidelity, it is also the most expensive fluid model, and in practice, it is currently used extremely rarely for industrial applications. The second option is that of using Large Eddy Simulation (LES), where only the effect of the largest and the most energetic scales is computed, while that of the smaller eddies is modelled. Though LES workflows were found to be able to accurately reproduce TIV for nuclear rods [35, 36], their application on complex bundled geometries would require grids of billions of cells. For example, a recent study by Matozinhos and Hassan [37] required 309 million cells to simulate a 5×5 bundle with a length of approximately 0.3 m. Such processing demands are hence currently limiting private companies from using numerical simulations of bundled configurations to design against GTRF. The last class of fluid models is that of Reynolds-Averaged Navier-Stokes (RANS) methods, where none of the scales are directly solved. Instead, one focuses on computing the total velocity fluctuations that are obtained by modelling the effect of all turbulent scales combined [34]. Though traditional RANS methods

were found to be able to accurately reproduce properties such as the vibration frequency of structures in axial TIV conditions, they appear to underestimate the amplitude of the fluid excitation, due to their incapacity to model the localized and correlated effect of turbulence over the rods [38].

To avoid the overhead of using an LES model, Nuclear Research Group (NRG) has developed a synthetic turbulence model, called the Anisotropic Pressure Fluctuation Model (AniPFM) [39–41]. This is based on a computationally cheap Unsteady Reynolds-Averaged Navier-Stokes (URANS) framework, and is suitable for incompressible calculations. It focuses on generating anisotropic spatial velocity fluctuations, using time correlation methods to propagate them in time. Once the velocity perturbations are obtained at a given timestep, they are used to solve for the pressure fluctuations. This model has been developed in-house using the open-source OpenFOAM library [42]. Before the start of this thesis, the AniPFM fluid model was successfully integrated within a Fluid-Structure Interaction (FSI) computational workflow, called NRG-FSIFOAM [38]. This used a 3D Finite Element Method (FEM) structural solver developed in deal.II [43], while the transfer of information between the two, as well as the coupling numerics, were handled by a third software, preCICE [44]. In parallel with the research herein, Freitas [41] conducted research that better determined the settings that one needs to use to increase the certainty of results when using the NRG-FSIFOAM workflow. The most elaborate validation case that had been used for NRG-FSIFOAM was the experiment published by Chen and Wambsganss [1], consisting of a flexible single brass beam subjected to turbulent axial water flow in an annular channel, as will be detailed in Subsection 3.5.1. In this work, this validation case will be referred to as the brass beam case.

The long-term objective of NRG for its NRG-FSIFOAM workflow is its use in simulating the FSI phenomena for configurations containing multiple rods. However, before doing so, optimizations in terms of the computational costs need to be made. Despite the progress made in terms of the fluid modelling costs thanks to the AniPFM, it was believed that improvements could still be made about the costs related to the structural modelling and the communication between the two domains. For one, the use of a 3D FEM grid to compute the structural response was seen as potentially superfluous, since the cylindrical, rigid fuel rods are predominantly subjected to lateral bending. Furthermore, it was identified that the computational overhead caused by using preCICE was significant. For example, van den Bos [38] found that approximately 6% of the computational time was spent on communicating and mapping the data between the separate structural and fluid solvers for the FSI case of Chen and Wambsganss [1].

Within this *status quo*, the main purpose of the work here is to adapt the NRG-FSIFOAM workflow so that it can be employed efficiently for simulating physics containing multiple rods subjected to axial water flow in PWR conditions. Amongst these physical phenomena, GTRF plays a central role, as was explained at the beginning of this Chapter. The proposed method to help the scaling of the NRG-FSIFOAM workflow to more complex geometries is to optimize the structural modelling and the way that this is coupled to the fluid solver. Before a research outlook is formulated in Section 1.2, first, the feasibility of using the AniPFM fluid model for simulating bundled configuration phenomena is assessed. This question had never been answered before this study, although without it, the further development of NRG-FSIFOAM would be redundant. Hence, in Subsection 1.1.2, the specific phenomena associated with bundled configurations are discussed. Additionally, the optimized workflow that will be developed for this thesis needs to be able to also reproduce other fundamental FSI phenomena whose design require high-fidelity simulations. Consequently, in Subsection 1.1.1 a short classification of the general family of phenomena within the family of Flow-Induced Vibrations (FIV) is given. In Section 1.2, the research objective and questions are formulated. In Section 1.3, a summary of the contents of the main Chapters in this work is offered.

1.1. FSI of isolated and bundled fuel rods

As mentioned at the beginning of this Chapter, in most of the currently operating nuclear reactors, that is, in PWRs, the rods in which the nuclear fuel is stored are cooled using axial water flow. This continues to flow downstream, where it acts onto turbines driving an electrical generator, before it is recirculated into the reaction chamber. This cycle inevitably leads to energy changes between the liquid (coolant water flow) and the solid (the reactor's components). Though this general pattern occurs in other components of the nuclear reactor, for example, in the heat exchanger [45], the scope of this thesis is narrower: it only treats phenomena associated with the fuel elements subjected to turbulent axial water flow. As a consequence, only these will be

elaborated upon in this Section. Particular attention is offered to the phenomena in bundled configurations of fuel rods, given that their analysis is the major long-term goal of which this research is a component of. It is of interest to not only present the physics that occur inside the reaction chamber, but also which of those need to be captured by using the relatively high-fidelity updated version of the previous NRG-FSIFOAM workflow. Another important consideration is whether the given workflow can be used to simulate the behaviour of multiple fuel rods, since previous to this research only validations using single rods had been conducted. First, a brief introduction to FIV is provided in Subsection 1.1.1, while in Subsection 1.1.2 a description of phenomena specific to clustered configurations is formulated.

***Most important, **Should be considered, *Less likely

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	***	*	**	-

Table 1.1: the relative importance of different excitation mechanisms with respect to the flow direction, the orientation of the flow and the type of phases present, as classified in [46]

1.1.1. FIV in axial flow and modelling approaches

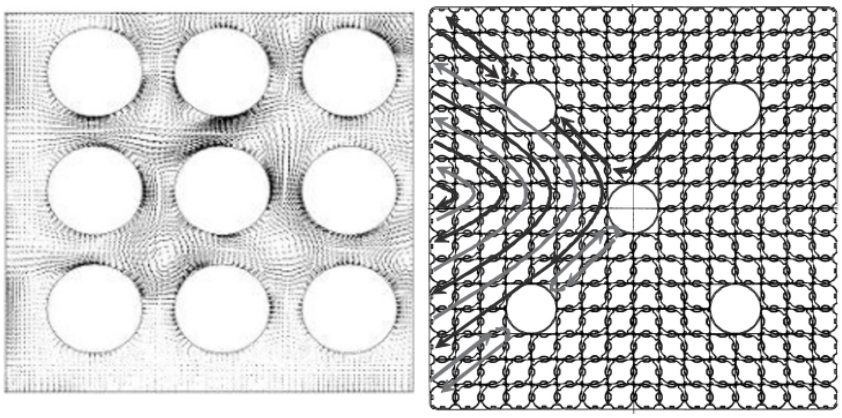
FIV is the name of the discipline that studies the exchange of energy between the fluid and the structure. It has emerged as a formal area of research in the first half of the 1970s [47]. In the literature, numerous ways to classify the encountered phenomena are offered, based on their phase and unsteadiness [45], the fluid velocity at which they occur [48], or the source of excitation [49]. In this work, the classification of Pettigrew [46] is used, as presented in Table 1.1. One can observe that for axial, internal, liquid flow, the most important mechanisms are acoustic resonance, turbulence excitation, and fluidelastic instability. Each of these is presented separately in the following.

Acoustic resonance occurs when fluid-dynamic periodic phenomena such as vortex shedding occur inside the reactor at frequencies that are close to the eigenfrequencies of the structure [45]. This can lead to resonance and therefore to potential damage of the structure. Despite the risk, design against acoustic resonance can be managed with simple analytical models [45], and therefore it is not required to be included in the NRG-FSIFOAM workflow.

Turbulence excitation refers to the excitation of the structure by the turbulence generated inside the axial water flow. It represents the main motivation for using the AniPFM model, and it has been extensively presented by van den Bos [38] and Kottapalli [50] for single rods. For such configurations, the two obtained promising prediction results for turbulence excitation. For this type of phenomenon, experimental data can only be reliably used if the laboratory model matches the geometry and flow conditions of the cluster very well [51–53]. If not, the accuracy of the results is within one to two orders of magnitude [51], making them too inaccurate

for GTRF predictions. In terms of numerical approaches for TIV of bundled configurations, this appears to have only been done by Bertocchi et al. [54], where a two-way coupled URANS-shell FEM approach was used to compute a small-scale 7-tube bundle. In this study, it was found that the amplitude of the excitations is underestimated. Thus, for TIV, the AniPFM model is expected to be adequate, and capable of predicting the vibration amplitudes better than a URANS fluid model.

Fluidelastic instability represents a two-way coupled phenomenon between the fluid and the structure. It occurs if the structural deformations due to the hydrodynamic loads modify the fluid flow in a way that encourages larger displacements. If this positive feedback loop is reinforced in time, the stability limit of the structure may be reached, leading to catastrophic failure. Though the phenomenon is of importance for heat exchangers in cross-flow [55], the velocities at which it occurs for axial flow are much larger than those found in operating nuclear reactors [56]. However, in practice, for bundles of fuel rods, one needs to consider that mixing vanes are placed on the spacer grids. Their role is to increase the heat transfer that occurs at the interface with the rods by generating turbulence [45]. In doing so, cross-flow locally occurs. This can become quite significant, as shown, for example, by the experimental results of Han et al. [57], as seen on the left of Figure 1.3.



(a) (b)

Figure 1.3: Left: cross-flow velocity vectors generated by a bundled rod-mixing vane assembly, 10 hydraulic diameters downstream of the mixing vanes, measured using Laser Doppler Anemometry (taken from [57]); right: idealised hydrodynamically balanced cross-flow generated by a suitable symmetric mixing vane design (taken from [16])

In the context of mixing vane-induced cross-flow, it was found that fluidelastic instability due to cross-coupling may occur [45]. This occurs if the fluid forcing over given degrees of freedom of the structure are heavily dependent on the others, and if this dependence is such that small perturbations can be destabilizing. To mitigate this, it is recommended that the mixing vans are placed so that they generate eddies of opposite vorticities, therefore balancing their excitation effect. This concept is seen on the right of Figure 1.3. As this phenomenon is caused by strongly coupled FSI, it can only be captured using high-fidelity FSI workflows. A strongly coupled FSI algorithm is one where information is exchanged multiple times between the fluid and the structure for a given timestep, until there is good matching between the fluid loads and the structural displacements on the FSI boundary [58]. The current NRG-FSIFOAM workflow accommodates for two-way coupling, and to be able to predict structural instabilities, this capability will also be included in the newly developed workflow.

1.1.2. Phenomena specific to clustered cylinders in axial flow

Within the literature, two phenomena specific to bundled configurations of nuclear rods were identified: hydrodynamic coupling and added stiffness. The physics of the two and adequate modelling methods are discussed in this Subsection.

Hydrodynamic coupling refers to the phenomena where the movement of a given cylinder is propagated to the others through the fluid medium. Hence, the movement of one cylinder is dependent on not only the effect of the axial flow, but also on the movement of all other cylinders. This was first studied numerically

by Païdoussis [24] using a 2D potential flow model for a stationary and relatively heavy flow. His results can be vizualized in Figure 1.4: an initial lateral displacement in both directions of the right cylinder (number 1, upper part of (c) in Figure 1.4) leads to the excitation of the second cylinders' movement (number 2, lower part of (c) in Figure 1.4), though the latter was initially at rest. It is important to mention that due to hydrodynamic coupling, groups, rather than single eigenmodes, can be found within a narrow frequency range for the structure, where the shapes of the eigenmodes for the given configuration are seen in Figure 1.4, (b). Later, the same behaviour was also observed experimentally [26, 27]. The hydrodynamic coupling is heavily dependent on the distance between the rods in the bundle (the closer they are, the stronger the coupling), as well as the distance to the bounding walls of the bundle [25]. Though viscous coupling may also occur [59], the spacing at which this happens is much smaller than those found in nuclear applications.

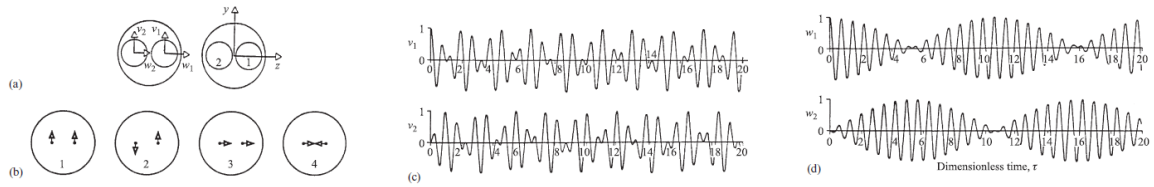


Figure 1.4: (a) Layout of the numerical experiment and of the considered DOFs; (b) the first group of eigenmodes; (c) the time response of the system to an initial unitary displacement in the v_1; (d) the w_1 degree of freedom (taken from [23])

Next to the hydrodynamic coupling, within the literature a second phenomenon specific to bundled configurations was identified, referred to as the added stiffness effect. It was first observed experimentally by Ricciardi and Boccaccio [28], who used a full-scale fuel assembly for their study, over which axial water flow was imposed. The size of the experimental chamber was only slightly larger than the size of the fuel assembly, thereby simulating the confinement of the axial water flow within a reactor chamber. The fuel assembly was deformed laterally using a hydraulic jack, and subsequently allowed to vibrate freely. A simplified scheme of the experiment can be seen on the left of Figure 1.5. It was found that the lateral displacements of the fuel assembly decreased with the axial velocity flow, that is, the fuel assembly became stiffer as the fluid velocity increased, as can be seen in the middle of Figure 1.5. This appears to be caused by the effect of the static deformation of the fuel assembly over the water flow: as the bundle deforms laterally, the change in the relative flow area between the opposite boundaries of the fuel assembly leads to a static pressure gradient that tends to stabilize the structural deformation. This hypothesis is confirmed by the measurements in axial velocity, as can be seen in Figure 1.5.

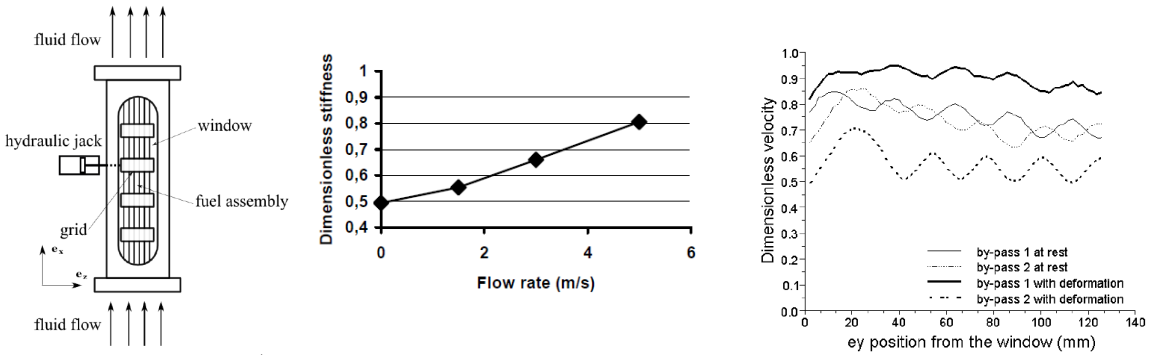


Figure 1.5: Left: experimental setup; middle: change of the effective stiffness of the FA with respect to the flow velocity; right: streamwise velocity component external at the bypass between the fuel assembly and the test section (taken from [28])

For this Subsection, it is important to assess whether the current NRG-FSIFOAM workflow is suitable for the simulation of bundled configurations. Hence, it is necessary to determine the complexity of the fluid model that needs to be used to correctly capture the hydrodynamic coupling and the added stiffness effect. For both, it can be concluded that a URANS model, and in particular, the AniPFM, is suitable. For one, the hydrodynamic coupling was accurately modelled using potential flow [24], and was found to be captured

qualitatively by URANS [60]. At the same time, the added stiffness effect was also observed numerically by Dressel et al. [29] when using URANS. Given that the AniPFM has a higher fidelity than the URANS, it is expected that the current fluid model will be capable of capturing both the hydrodynamic coupling and the added stiffness effects accurately.

1.2. Research questions and objectives

Based on the information presented in Section 1.1, it can be concluded that a two-way coupled FSI workflow that uses the AniPFM model can be expected to accurately model all relevant FSI phenomena for axial flows inside nuclear reactors that cannot be solved using simpler engineering approaches, such as TIV, self-induced excitations due to the crossflow induced by mixing vanes, or the hydrodynamic coupling and added stiffness for bundled configurations. Nevertheless, in its current version, the NRG-FSIFOAM's structural model remains cumbersome, using a potentially overcomplex structural solver, and three standalone codes to obtain the FSI computations. On the other hand, in current literature FEM solvers using 1D beam-element FEM solvers were successfully used [29, 61], along with eigenmode-based Reduced-Order Models (ROMs) [36, 62]. Next, the identified research gap is presented. Although these structural methods were found to work well with URANS [29, 62, 63] and with LES [36, 64, 65], to the knowledge of the author, their feasibility when combined with a synthetic turbulence fluid model such as the AniPFM was not tested prior to this thesis. What's more, the discretization settings that need to be used, such as the number of degrees of freedom that need to be utilized to describe the problem, are also not clear from the available literature. Thus, the identified knowledge gap for this thesis refers to the compatibility of the said structural models with the AniPFM, as well as the optimal settings to be used in running them. Thus, the following main question is formulated:

Is it feasible to compute the structural response of nuclear fuel rod-like structures using ROM methods that simplify a 1D beam discretization, for two-way coupled FSI applications in axial flow, wherein fluid forces are computed using a URANS Anisotropic Pressure-Fluctuation Model (AniPFM) ? If yes, what are the general discretization and setup guidelines to do so?

Before answering the main question of the research, it is necessary to verify that the newly developed structural solvers work correctly. An additional layer of complexity is added by the communication of the structural solver with the mapping and the subiteration modules that are required for its integration in an FSI workflow. In turn, these modules are themselves linked to the OpenFOAM fluid solver, as will be explained in Chapter 3, leading to additional question marks regarding the combined working of the methodology. Furthermore, although the AniPFM fluid model is the ultimate fluid model with which NRG-beamFoam is desired to be coupled, it is nevertheless a relatively complex solver, containing on top of a URANS solver a stochastic generator of non-dimensional velocity fluctuations, and a model of their transient transport that is based on a Partial Differential Equation (PDE) [38]. The AniPFM is hence not only more expensive than a standard URANS model, but it also leads to a higher difficulty in debugging and drawing conclusions about the specific characteristics of NRG-beamFoam if the two are coupled from the start of the thesis, with no other intermediate FSI results obtained from simpler fluid models. It is therefore preferable that the newly developed NRG-beamFoam methodology is developed sequentially, to ensure its correct functioning for different levels of fluid modelling complexity. Hence, to formulate a definitive and convincing answer to the main research question, the following subquestions are formulated:

1. Can the newly implemented structural solvers be used along with the newly developed mapping routines and subiteration algorithm for two-way FSI simulations? What is the accuracy of the ensuing workflow, compared to NRG-FSIFOAM for such a scenario?

2. What is the accuracy of the new FSI solver in computing axial-flow FSI when using a URANS fluid model? Are there any computational benefits compared to the NRG-FSIFOAM workflow?

3. What are the preferential settings that should be used for the novel FSI solver when combined with the AniPFM model?

Consequently, the main objective of the current thesis is:

To simplify and improve the computational costs of the NRG-FSIFOAM FSI workflow and to allow the higher-fidelity simulation of more complex configurations by using a simplified and cost-effective structural solver.

Based on the available literature, other studies dealing with the topic of axial-flow FSI have successfully used either 1D beam FEM elements [29, 61], or eigenmode-based Reduced-Order Models (ROMs) [36, 62] methodologies to reduce the structural computational costs, by using a reduced number of degrees of freedom to describe the solid response. Those structural solver possibilities will be further explored in Chapter 2. Later in the thesis, the two will be combined to maximise computational efficiency. Assuming that the right loads are provided at the interface between the water and the rods, there is no indication in the literature that these structural models wouldn't be compatible for the physics presented in Section 1.1. These are intended to supersede the current Lagrange-element 3D FEM solver that NRG-FSIFOAM uses. However, none of these methods are readily available in deal.II. To avoid the overhead of using a standalone structural code and a coupling software, *it was decided to implement the new structural solver directly in OpenFOAM, to ease communication with the AniPFM routines*. The development of an OpenFOAM-only coupled workflow within NRG that is tailored specifically for axial-flow FSI also implies that the overall usability increases when compared to the current preCICE approach, by specializing the solver and therefore reducing the number of user-inputted settings required. This helps in fulfilling both the cost reduction and the simplification objective stated above.

Although the main objective that was stated for the thesis is to develop a simplified structural solver in OpenFOAM that can be used in combination with the AniPFM fluid model, in practice a secondary objective also arose. In renouncing the use of preCICE, all its coupling functions also need to be fulfilled by the newly developed methodology. First, the transfer of information between the structural and fluid grid needs to be facilitated. In FSI problems, this mapping is bidirectional, where the structural displacements are sent to the fluid grid, and the fluid loads to the structural mesh. What's more, a strategy dealing with the potentially strong coupling between the structure and the fluid needs to be implemented. In practice, the strong FSI coupling is solved by using a subiterative approach, where the fluid and the structural solvers are run multiple times at a given timestep until convergence between the fluid loads and the structural displacements on the FSI boundary is obtained [58]. All of these new features represent an opportunity to again reduce the computational costs that were associated with the NRG-FSIFOAM workflow because of the non-specialized mapping and subiteration routines used in preCICE. In this thesis, the improvement of the mapping and subiteration costs is achieved by using specific routines that apply to nuclear fuel elements in axial flow, characterized by their slenderness, and small relative displacements. Thus, the second objective of the research is:

To simplify and improve the computational costs of the NRG-FSIFOAM FSI workflow and to allow the higher-fidelity simulation of more complex configurations by using simplified and cost-effective mapping and FSI subiteration techniques.

For this thesis, the newly proposed FSI solver that will be developed to tackle the two main objectives will be called NRG-beamFoam. The name originated in the use of the OpenFOAM library, and the 1D beam element FEM solver that will be implemented as the backbone of the structural solver.

1.3. Report outline

The rest of the report is dedicated to developing the theoretical knowledge, the methodology, and to presenting results that can answer the questions put forward in Section 1.2. Thus, Chapter 2 lays out the concepts from the fields of FSI, fluid dynamics, FEM, and ROM that are necessary to build the structural solver and the associated mapping and subiteration routines, and to understand requirements imposed by the fluid solver. In Chapter 3, specific details are provided of how the different modules of the FSI loop of this thesis are implemented, along with the test cases that will be used for the verification in this thesis. In Chapter 4, an isolated verification of the main modules of the NRG-beamFoam implementation is presented, that is, the beam-element FEM solver, the load interpolator (mapping the fluid loads on the FSI boundary to the structure), and the displacement interpolator (transferring the structural displacements to the fluid nodes on the FSI boundary). This isolated verification of the components is necessary before the use in FSI simulations to be better able to pinpoint potential issues that arose in the development of any of the modules. Once these individual tests are passed, in Chapter 5 the subiteration algorithm is verified, before laminar and URANS simulations with NRG-beamFoam are run, and the outputs are compared with the NRG-FSIFOAM workflow. For the latter, a computational cost comparison is also provided. In Chapter 6, the findings that are obtained when running the NRG-beamFoam model with the AniPFM are provided. Chapter 7 deals with the individual verification of the ROM structural solver, as well as its behaviour when used with the AniPFM fluid model for

FSI simulations. Finally, in Chapter 8, the conclusions to the thesis are presented, along with the answers to the research questions, and the recommendations for future work.

2

Theoretical background

As explained in Chapter 1, the objective of this research is to implement an accurate and efficient structural model for FSI simulations that is suitable for axial-flow TIV. To this end, an FEM approach will be used, along with additional ROMs that are meant to reduce the size of the structural problem that needs to be solved. In this Chapter, aspects referring to the FSI are first presented in Section 2.1. Section 2.2 presents introductory aspects about the fluid modelling. Next, the theory of the FEM model is presented in Section 2.3. Finally, different ROM methods are presented in Section 2.4, and their feasibility for this study is assessed.

2.1. FSI

In Subsection 1.1.1, it was pointed out that given the coupling between the structural movement and the resulting load forces, fluidelastic instabilities may occur for the fuel elements inside the nuclear reactor chamber. This represents a situation where the structural solution dynamically influences the fluid one, and vice versa. Thus, it is required that the fluid problem is formulated by taking into account the current state of the structure, and analogously, the structural solution should consider the current configuration of the flowfield. The field of research that deals with accommodating the coupling of the structural and the fluid solvers in a numerical setting is known as FSI.

The purpose of this Section is to present general aspects about what exactly the coupling between the fluid and the structural solvers entails. How these challenges are dealt with in this thesis is predominantly explained in Chapter 3, with specific references made throughout this Section. Thus, Subsection 2.1.1 presents the physical conditions that need to be satisfied in a pointwise manner at the interface between the fluid and the structure. Subsection 2.1.2 then presents the specific requirements that arise when one starts considering how to enforce the said physical conditions onto the discretized domains that are usually used to solve the two continua problems. Given the strong coupling between the flow and the structure for nuclear reactor applications, particular attention is allocated to implicit partitioned FSI coupling algorithms in Subsection 2.1.3.

For this Section, it is assumed that the timestep used for the two solvers is equal. Consequently, no details are offered, for example, about subcycling partitioning methods [66], that use different timestep values for the two domains.

2.1.1. Interface conditions

When FSI is taken into account for a physical problem, specific boundary conditions need to be satisfied at the interface between the fluid and the structure. In literature (for example, [67]), two different sets of boundary conditions are distinguished: the kinematic and the dynamic boundary conditions.

The kinematic conditions are based on the physical observation that at the wall where the FSI occurs, the fluid must locally have the same velocity as the wall itself. This corresponds to the well-known no-slip condition from fluid mechanics, written for a moving wall:

$$\boldsymbol{U}_{\mathbf{f}}(\boldsymbol{x}) = \frac{\partial \boldsymbol{u}_{\mathbf{s}}(\boldsymbol{x})}{\partial t}, \tag{2.1}$$

where $\boldsymbol{U_f}$ represents the velocities for the fluid domain, $\boldsymbol{u_s}$ the displacements for the structural one, and $\boldsymbol{x}$ an arbitrary location on the FSI boundary. What's more, at a given time instance, it must also be that the local displacements are equal between the two domains everywhere across their interface:

$$\boldsymbol{X_f}(\boldsymbol{x}) = \boldsymbol{u_s}(\boldsymbol{x}), \tag{2.2}$$

where $\boldsymbol{X_f}$ represents the displacements of the fluid domain.

Apart from the compatibility conditions of the displacements and velocities on the FSI boundary, an equilibrium between the stresses must also occur in the direction normal to the interface between the two domains:

$$\boldsymbol{\sigma_f}(\boldsymbol{x}) \cdot \boldsymbol{n_f}(\boldsymbol{x}) = -\boldsymbol{\sigma_s}(\boldsymbol{x}) \cdot \boldsymbol{n_s}(\boldsymbol{x}), \forall \boldsymbol{x}, \tag{2.3}$$

with $\boldsymbol{\sigma}$ denoting the stresses exerted onto the FSI boundary, and $\boldsymbol{n}$ the direction that is locally normal to the FSI boundary. Condition (2.3) is known as the dynamic interface condition.

2.1.2. Specific challenges in discretized FSI

The observations made in Subsection 2.1.1 refer to the physical requirements that are valid for a *continuous* formulation of the fluid and the structural problems. However, for industrial applications, both of these are often solved using discretized approaches. What's more, two different families of numerical methods are often used, where a Finite-Volume formulation is used for the fluid, and a Finite Element one for the solid. Thus, different grids are used between the two, where the fluid mesh is often finer. This will also be the case for the study herein. Hence, the non-matching, independent meshes lead to two specific problems: the order in which the two solvers are executed at a given timestep, and the transfer of information between the discretized fluid and structure representations. These two aspects will be discussed in the following.

From an FSI perspective, it is required that the solvers are run such that at the end of a given timestep, the conditions (2.1)-(2.3) are all met simultaneously for both solvers. This would ideally be obtained by solving the unified system of equations that is formed by combining the structural and the fluid unknowns. This approach is known as *monolithic*, and it represents a stable and accurate solution to the FSI problem. However, this is hindered by the different requirements that are specific to solving each of the two media. For one, as already mentioned, the discretization techniques are different for the two. Thus, due to the specific techniques required to solve each domain, separate, standalone codes are developed for each, and what is commonly preferred over a monolithic approach is a *partitioned* one. This enables the use of separate, pre-existing solvers for the fluid and the structure, and the exchange of information that is required to satisfy conditions (2.1)-(2.3) is executed before a standalone solver is run. Based on how the information is transferred between the two solvers, parallel and serial partitioned schemes can be distinguished, as visualized in Figure 2.1. For the former, the exchange of information occurs once, at the beginning of each timestep, in a bidirectional manner, before the fluid and the structural solvers are run at the same time. For the latter type of partitioning schemes, at the beginning of timestep n, solver A transfers its coupling information to solver B. Subsequently, solver B advances to time instance $n+1$, and it subsequently sends the predictor information from timestep $n+1$ back to the solver A. This predictor information is then used as input to march the solver A from timestep n to $n+1$.

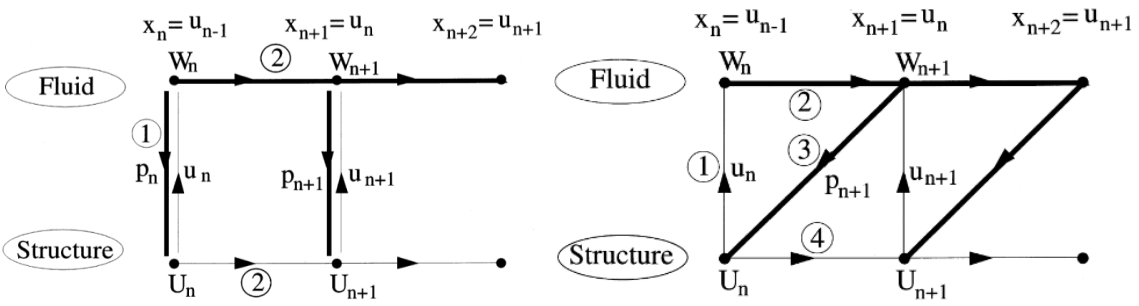


Figure 2.1: Visual representation of the parallel (left) and the serial (right) partitioned coupling schemes; taken from [66]

For any of these two approaches, a partitioning error is introduced, that wouldn't exist if one used a monolithic formulation. This is caused by the time-lagged transfer of information between the domains, and thus the mere approximate satisfaction of conditions (2.1)-(2.3). For example, if one used the serial scheme on the right of Figure 2.1, the fluid solution obtained for timestep $n+1$ would be computed by considering the structural state at timestep n. However, the conditions (2.1)-(2.3) ought to be fulfilled at every instance in time, that is, they would be rigorously fulfilled if the structural solution at timestep $n+1$ were taken into consideration. An error therefore ensues, since for the kinematic interface conditions the displacements at the previous timestep are used.

To mitigate the partitioning error, different approaches can be taken. One of them is to reduce the timestep that is used to advance the solution, and thus to decrease the differences that occur between the states at timestep n and $n+1$. For the considered application this can lead to a very rapid increase in the computational cost. Another option is to use high-order time integration schemes that are specifically designed to reduce partitioning error [68]. However, for strong coupling, this method also reduces the computational efficiency significantly, since the number of timesteps that is required can quickly increase. The other alternative is to execute the partitioned scheme multiple times at each timestep. This is known as an implicit partitioning scheme, or a subiteration coupling algorithm. For this thesis, this approach is preferred to ensure the coupling between the two domains. Hence, different subiteration schemes will be presented in Subsection 2.1.3.

The second aspect that needs to be considered by discretized FSI workflows is the mapping of information between the grids. Given the numerical methods used to solve the solid and fluid problems, the stresses and the displacements present in (2.1)-(2.3) are only known at the elements' boundary nodes or cell face centers. However, as already mentioned, it is often the case that the structural and the fluid grids are non-conformal on the FSI boundary. Thus, to enforce the interface conditions, at least one of the displacements or stress fields needs to be interpolated to the other grid. The exchange of information is said to be consistent if a uniform field is correctly transferred to the other grid, or conservative, if the work done on the FSI boundary by the solution mapped onto the target mesh is equal to the work done on the source one. Multiple interpolation techniques have been established in literature, such as nearest neighbour, weighted residual methods, or Radial-Basis Function Interpolation (RBFI) [69]. All of these represent general approaches for mapping the information between two arbitrary grids. For this thesis, mapping routines specific to a 1D structural solver will be used, and they will be elaborated upon in Subsections 3.4.1 (for transferring the fluid stresses to the structural nodes) and 3.4.4 (for mapping the structural displacements to the fluid grid), respectively. Those are suitable for the main structural model that will be used for this thesis, and which will be elaborated in Section 2.3.

2.1.3. Implicit partitioned coupling schemes

For PWR reactors, the water flow over the fuel elements is characterized by a structure-to-fluid density ratio $\frac{\rho_s}{\rho_f}$ equal to approximately 8. This low value is conducive to strong coupling, and it justifies the need to implement a subiteration algorithm. In this Subsection, it is assumed that the structural displacements are updated at each subiteration, since this corresponds to the choice made for the subiterator module developed in this thesis, as will be later presented in Subsection 3.4.3. Nevertheless, the same principles would apply if one updated the stresses on the FSI boundary.

Updating the FSI boundary displacements at an arbitrary i^{th} subiteration consists of taking as input the previously obtained displacements on the FSI boundary, $\boldsymbol{d}^i$, and using these to successively compute the new fluid loads, and the new prediction of the displacements on the FSI boundary, $\tilde{\boldsymbol{d}}^i$:

$$\tilde{\boldsymbol{d}}^i = \mathtt{S} \circ \mathtt{F}(\boldsymbol{d}^i), \tag{2.4}$$

where the S and F operators signify the structural and the fluid solvers, respectively. The goal of the subiteration process can be summarized as:

$$\text{At each timestep, continue implicit coupling algorithm until } \tilde{\boldsymbol{d}}^i = \boldsymbol{d}^i.$$

When this condition is reached, the monolithic solution is obtained. This formulation of the subiteration problem is known as the fixed-point equation. On the other hand, if one defines the residual between the

new prediction and the provided input, $\boldsymbol{r}_{\mathrm{i}}$, as:

$$\boldsymbol{r}_{\mathrm{i}} = \tilde{\boldsymbol{d}}^{\mathrm{i}} - \boldsymbol{d}^{\mathrm{i}}, \tag{2.5}$$

then the objective of the subiteration algorithm becomes:

$$\text{At each timestep, continue implicit coupling algorithm until } \boldsymbol{r}_{\mathrm{i}} = \boldsymbol{0},$$

which represents a minimization problem. The most straightforward way to update the fluid loads on the FSI boundary is to use the fixed-point iteration update, that is, to simply update the load inputs with the residuals obtained:

$$\text{Fixed-point iteration: } \boldsymbol{d}^{\mathrm{i+1}} = \boldsymbol{d}^{\mathrm{i}} + \boldsymbol{r}_{\mathrm{i}}. \tag{2.6}$$

However, this is a sound approach only if the composite $\mathtt{S} \circ \mathtt{F}$ operator in (2.4) is a contraction [58], which is not always the case. If the operator is a dilatation, then the fixed-point iteration can become unstable. Alternatively, a faster convergence could also be obtained compared to the fixed-point iteration approach (2.6) in the cases where the operator is a strong contraction. Thus, what is often done in practice, is to generalize the fixed-point iteration by multiplying the residual with a scalar $\omega_{\mathrm{UR,i}}$ before updating the solution:

$$\boldsymbol{d}^{\mathrm{i+1}} = \boldsymbol{d}^{\mathrm{i}} + \omega_{\mathrm{UR,i}} \boldsymbol{r}_{\mathrm{i}}. \tag{2.7}$$

This approach enables stability for the cases where the operator $\mathtt{S} \circ \mathtt{F}$ is a dilatation by using a value of $\omega_{\mathrm{UR,i}}$ that is smaller than one. For strong contracting behaviour, the value of $\omega_{\mathrm{UR,i}}$ can be increased to values above unity. The scalar $\omega_{\mathrm{UR,i}}$ will also be called the Under-Relaxation Factor (URF) in the remainder of this thesis.

Having introduced the concept of the URF, the simplest implicit scheme that can be used is one where a fixed value is used for all subiterations:

$$\text{Fixed URF: } \boldsymbol{d}^{\mathrm{i+1}} = \boldsymbol{d}^{\mathrm{i}} + \omega_{\mathrm{UR,0}} \boldsymbol{r}_{\mathrm{i}}. \tag{2.8}$$

Though the update in (2.8) is very easy to implement, it presents specific disadvantages. For one, the value of the URF must be set at the beginning of the FSI simulation, when the behaviour of the $\mathtt{S} \circ \mathtt{F}$ operator cannot be accurately predicted, and thus the optimal value of $\omega_{\mathrm{UR,0}}$ cannot be easily determined. Furthermore, the value of $\omega_{\mathrm{UR,0}}$ is usually kept constant for all timesteps, though the change in the residual $\boldsymbol{r}_{\mathrm{i}}$ with respect to the displacements' updates $\boldsymbol{d}^{\mathrm{i}}$ may vary during the transient. To combat these limitations, alternative methods have been put forward. These use the values of the residuals at the previous iterations to try and estimate the changes in the current residuals $\boldsymbol{r}_{\mathrm{i}}$ with respect to the structural displacements $\boldsymbol{d}^{\mathrm{i}}$. Subsequently, the displacements are updated by considering the approximated search direction along which the residual is minimized. The Aitken method does so by using the residuals at the two latest subiterations, $\boldsymbol{r}_{\mathrm{i\text{-}1}}$ and $\boldsymbol{r}_{\mathrm{i}}$ [70]:

$$\text{Aitken: } \boldsymbol{d}^{\mathrm{i}} = \boldsymbol{d}^{\mathrm{i\text{-}1}} + \omega_{\mathrm{UR,i\text{-}1}} \boldsymbol{r}_{\mathrm{i\text{-}1}};$$

$$\text{prescribe value of } \omega_{\mathrm{UR,0}},\ \omega_{\mathrm{UR,i}} = -\omega_{\mathrm{UR,i\text{-}1}} \frac{\boldsymbol{r}_{\mathrm{i\text{-}1}}^{T} (\boldsymbol{r}_{\mathrm{i}} - \boldsymbol{r}_{\mathrm{i\text{-}1}})}{\|\boldsymbol{r}_{\mathrm{i}} - \boldsymbol{r}_{\mathrm{i\text{-}1}}\|_2^2},\ i \geq 1. \tag{2.9}$$

Apart from the Aitken method, the Inverse Quasi-Newton - Inverse Least-Squares (IQN-ILS) method is also based on the idea of using the previous values of the residuals and the fluid loads to obtain an estimation of the direction along which the residual is minimized. To this end, the difference between the current solution after the i^{th} subiteration, and all of the previous ones at the current timestep are used. For example, the difference in residuals between the current subiteration and the previous one is given by [58]:

$$\Delta \boldsymbol{r}_{\mathrm{i\text{-}1}} = \boldsymbol{r}_{\mathrm{i}} - \boldsymbol{r}_{\mathrm{i\text{-}1}}, \tag{2.10}$$

$$\Delta \tilde{\boldsymbol{d}}_{\mathrm{i\text{-}1}} = \tilde{\boldsymbol{d}}^{\mathrm{i}} - \tilde{\boldsymbol{d}}^{\mathrm{i\text{-}1}}. \tag{2.11}$$

These are then gathered into two matrices, $\boldsymbol{W}_{\mathbf{IS}}$ and $\boldsymbol{V}_{\mathbf{IS}}$, along whose columns the values of $\Delta \boldsymbol{r}$ and $\Delta \tilde{\boldsymbol{d}}$ are stored for all previous subiterations at the current timestep:

$$\boldsymbol{W}_{\mathbf{IS}} = (\Delta\boldsymbol{r}_0\ \Delta\boldsymbol{r}_1\ \ldots\ \Delta\boldsymbol{r}_{\text{i-1}}), \tag{2.12}$$

$$\boldsymbol{V}_{\mathbf{IS}} = \left(\Delta\tilde{\boldsymbol{d}}_0\ \Delta\tilde{\boldsymbol{d}}_1\ \ldots\ \Delta\tilde{\boldsymbol{d}}_{\text{i-1}}\right). \tag{2.13}$$

Using the information stored in $\boldsymbol{W}_{\mathbf{IS}}$, the objective is to find the linear combination of the $\Delta\boldsymbol{r}$ values that is closest to the value of $-\boldsymbol{r}_{\text{i}}$ in a least-square sense:

$$\text{determine } \boldsymbol{\alpha}_{\mathbf{IS}} \text{ s.t. } \boldsymbol{\alpha}_{\mathbf{IS}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^i}{\text{argmin}} \left\| \boldsymbol{W}_{\mathbf{IS}}\boldsymbol{\beta} + \boldsymbol{r}_{\text{i}} \right\|_2. \tag{2.14}$$

The update of the fluid loads is then done using:

$$\text{IQN-ILS: } \boldsymbol{d}^{\text{i+1}} = \tilde{\boldsymbol{d}}^{\text{i}} + \boldsymbol{V}\boldsymbol{\alpha}_{\mathbf{IS}}. \tag{2.15}$$

Other variations of the IQN-ILS method were also presented in the literature, for example, by Haelterman et al. [71], who used residuals from the previous timesteps, and an adequate filtering technique to prevent ill-conditioning, obtaining improved performance over the version of IQN-ILS that is presented herein.

The IQN-ILS method is known to be the best performing subiteration algorithm of all methods presented in this Subsection [58], and it is also the implicit coupling scheme that NRG-FSIFOAM uses. Nevertheless, it is also the most complicated, as it requires the storage and continuous update of the matrices $\boldsymbol{V}_{\mathbf{IS}}$ and $\boldsymbol{W}_{\mathbf{IS}}$. By contrast, the Aitken method only demands that the values of $\boldsymbol{r}_{\text{i-1}}$ and $\omega_{\text{UR,i-1}}$ are stored from the previous subiteration. Since this thesis represents the first instance where an implicit partitioning scheme is implemented for the proposed NRG-beamFoam methodology, it was decided to use the simpler Aitken method. Nevertheless, it is recommended that in the future an IQN-ILS algorithm is also attempted, to further increase the computational efficiency of the workflow.

2.2. CFD

Computational Fluid Dynamics (CFD) refers to the field of research that uses computational methods to analyze fluid flows. In this Section, the CFD theory will only be presented for the general forms of the Navier-Stokes equations when using a Finite-Volume approach, Subsection 2.2.1, the turbulence model used for this thesis, Subsection 2.2.2, and the novel AniPFM fluid model that will be used for the thesis, Subsection 2.2.3.

2.2.1. Governing equations

For this thesis, only the conservation of mass and momentum are built into the model. In other words, the effect of the energy exchange is neglected. The fluid flow is thus governed by the Navier-Stokes equations for mass and momentum, which can be written in block form as [72]:

$$\frac{\partial \boldsymbol{W}}{\partial t} + \underbrace{\frac{\partial \boldsymbol{E}}{\partial x} + \frac{\partial \boldsymbol{F}}{\partial y} + \frac{\partial \boldsymbol{G}}{\partial z}}_{\text{convection}} = \underbrace{\boldsymbol{J}}_{\text{source terms}}, \text{where} \tag{2.16}$$

$$\boldsymbol{W} = \begin{bmatrix} 0 \\ u_{\text{f}} \\ v_{\text{f}} \\ w_{\text{f}} \end{bmatrix}, \boldsymbol{E} = \begin{bmatrix} u_{\text{f}} \\ u_{\text{f}}^2 + \frac{p}{\rho_{\text{f}}} - \frac{1}{\rho_{\text{f}}}\boldsymbol{\tau}_{\text{f,xx}} \\ u_{\text{f}}v_{\text{f}} - \frac{1}{\rho_{\text{f}}}\boldsymbol{\tau}_{\text{f,xy}} \\ u_{\text{f}}w_{\text{f}} - \frac{1}{\rho_{\text{f}}}\boldsymbol{\tau}_{\text{f,xz}} \end{bmatrix}, \boldsymbol{F} = \begin{bmatrix} v_{\text{f}} \\ u_{\text{f}}v_{\text{f}} - \frac{1}{\rho_{\text{f}}}\boldsymbol{\tau}_{\text{f,xy}} \\ v_{\text{f}}^2 + \frac{p}{\rho_{\text{f}}} - \frac{1}{\rho_{\text{f}}}\boldsymbol{\tau}_{\text{f,yy}} \\ u_{\text{f}}w_{\text{f}} - \frac{1}{\rho_{\text{f}}}\boldsymbol{\tau}_{\text{f,yz}} \end{bmatrix}, \boldsymbol{G} = \begin{bmatrix} w_{\text{f}} \\ u_{\text{f}}w_{\text{f}} - \frac{1}{\rho_{\text{f}}}\boldsymbol{\tau}_{\text{f,xz}} \\ u_{\text{f}}w_{\text{f}} - \frac{1}{\rho_{\text{f}}}\boldsymbol{\tau}_{\text{f,yz}} \\ w_{\text{f}}^2 + \frac{p}{\rho_{\text{f}}} - \frac{1}{\rho_{\text{f}}}\boldsymbol{\tau}_{\text{f,zz}} \end{bmatrix}. \tag{2.17}$$

In equations (2.16) and (2.17), u_{f}, v_{f}, and w_{f} refer to the fluid velocity components in the x, y, and z directions, p and ρ_{f} to the fluid pressure and density, and $\boldsymbol{\tau}$ to the viscous stress tensor. The term on the right-hand side of (2.16), $\boldsymbol{J}$, refers to the source terms added to each of these equations. For this particular case, no source terms are considered, that is, $\boldsymbol{J}^T = [0\ 0\ 0\ 0]$. For simplicity, in the following, the right-hand side of (2.16) will be considered equal to $\mathbf{0}$. Equation (2.16) assumes an Eulerian frame of reference, and it therefore shows that at each fixed point within the fluid domain, the balance of the mass and momentum is modified by the locally occurring convection, and by the local generation of mass or momentum.

Equation (2.16) is formulated for a continuum. In practice, the fluid solution is subdivided into multiple finite volumes, whose unification forms the exact fluid control volume at all time instances. It is therefore of interest to obtain a spatially discretized system starting from (2.16), such that the fluid state at a given time instance can be fully described by a finite number of degrees of freedom. This is done by applying the Gauss

theorem for every element of the fluid discretization. For simplicity, it is first assumed that the mesh is static throughout the analysis, such that for the i$^{\text{th}}$ finite volume (2.16) becomes:

$$\frac{\partial}{\partial t}\int_{V_{\text{fix, i}}} \boldsymbol{W}\,dV + \int_{S_{\text{fix, i}}} [\boldsymbol{E},\boldsymbol{F},\boldsymbol{G}]\cdot\boldsymbol{n}\,dS = \boldsymbol{0}, \tag{2.18}$$

where $V_{\text{fix,i}}$ and $S_{\text{fix,i}}$ refer to the volume and the outer surface of the i$^{\text{th}}$ element, and $\boldsymbol{n}$ represents the local normal of the bounding cell faces. Equation (2.18) shows that the time variation of the total momentum within a fluid cell is equal to the flux of momentum that goes inside each finite volume through its boundary faces.

Having presented the general finite volume philosophy for fixed grids, moving meshes are now discussed, as this will be used for the two-way coupling functionality of NRG-beamFoam. In contrast to (2.18), when one is deforming the mesh, the momentum balance also depends on the relative change in the finite volume cells themselves. One can assimilate the effect in the changing finite volumes with an additional convective term caused by the local movement of the grid relative to the local flowfield. This additional flux due to the mesh movement is proportional to the average local mesh velocity computed between consecutive timesteps, $\boldsymbol{V_m}$. Thus, by including the mesh motion, one obtains the Arbitrary-Lagrange Eulerian (ALE) formulation of the Navier-Stokes equations [73]:

$$\frac{\partial}{\partial t}\int_{V_{\text{mov, i}}} \boldsymbol{W}\,dV + \int_{S_{\text{mov, i}}} ([\boldsymbol{E},\boldsymbol{F},\boldsymbol{G}]\cdot\boldsymbol{n} - \boldsymbol{W}\boldsymbol{V_m}\cdot\boldsymbol{n})\,dS = \boldsymbol{0}. \tag{2.19}$$

The ALE nomenclature stems from the fact that the convective term of the mesh motion is computed by considering a Lagrangian approach, while the fluid solution is computed using a finite-volume methodology, and hence an inherently Eulerian approach. To compute the average mesh motion velocity $\boldsymbol{V_m}$, one uses the Geometrical Conservation Law (GCL), which relates the change in the volume of the cells to the volume "swept" by its boundaries [73]:

$$\frac{d}{dt}\int_{V_{\text{mov, i}}} dV = \int_{S_{\text{mov, i}}} \boldsymbol{V_m}\cdot\boldsymbol{n}\,dS. \tag{2.20}$$

Having presented the application of the Gauss' theorem for moving meshes, the resulting system of equations is discussed. In the Finite-Volume Method (FVM), the quantities in $\boldsymbol{W}$ are assumed to be the unknowns, and their solution is stored at the cell center. Furthermore, it hypothesized that the unknowns are constant throughout each finite volume, such that the ALE formulation of the Navier-Stokes equations, (2.19), becomes:

$$\frac{d}{dt}\left(V_{\text{i}}\boldsymbol{W}_{\text{i}}\right) + \boldsymbol{\Phi}_{\text{i}}\left(\boldsymbol{W}_{\text{j}}, \boldsymbol{V_m}\right) = \boldsymbol{0}, \tag{2.21}$$

where V_{i} represents the volume of the currently considered cell, and $\boldsymbol{\Phi}$ the convective terms obtained using the information from the neighbouring cells, and the deformation of the cell itself. It is worth noting that in equation (2.21) j is the index used for the neighbouring cells. Thus, (2.21) represents the semi-discretized fluid equations that need to be solved to obtain the unknown values of the velocity components. Due to the flux between neighbouring cells, the values obtained for the different finite volumes are coupled, and therefore a system containing all fluid unknowns needs to be solved at a given timestep. To obtain a fully discretized system, a time discretization scheme should also be applied for the velocity terms. This aspect is not presented in this report, but the interested reader is referred to, for example, [74], where specific information about the schemes used in OpenFOAM is offered.

2.2.2. Turbulence modelling

A particular challenge that is posed by the Navier-Stokes equations (2.16) is the non-linear velocity terms, which lead to the generation of turbulence. This requires that the effect of the turbulent scales be accounted for in the fluid model. As already mentioned in Chapter 1, this can be done using formulations of different complexities (DNS, LES, RANS). For this work, similar to the previous research using the AniPFM, a URANS approach is chosen. In this Subsection, only the specific model that is directly used for the outcome of the thesis are discussed. More ample reviews were presented by, for example, van den Bos [38]. For conciseness,

Einstein notation is used. Thus, in particular for this Subsection, u_i refers to the different velocity components.

The RANS methods are based on the principle of the Reynolds decomposition, where the instantaneous values of the velocity and of the pressure in (2.16) are decomposed into their mean and their instantly fluctuating component:

$$u_\mathrm{i} = \bar{u}_\mathrm{i} + u'_\mathrm{i}, p = \bar{p} + p', \text{ where} \tag{2.22}$$
$$\bar{u} = \lim_{M\to\infty} \frac{1}{M}\sum_M u, \text{ and } \bar{p} = \lim_{M\to\infty} \frac{1}{M}\sum_M p$$

represent the velocity and pressure averages. Replacing (2.22) in (2.16), and using the different statistical properties of the mean and fluctuating fields, one obtains [38]:

$$\begin{aligned} \frac{\partial \bar{u}_\mathrm{i}}{\partial x_i} &= 0 \\ \frac{\partial \bar{u}_\mathrm{j}}{\partial t} + \frac{\partial \bar{u}_\mathrm{i}\bar{u}_\mathrm{j}}{\partial x_i} &= -\frac{1}{\rho_\mathrm{f}}\frac{\partial \bar{p}}{\partial x_j} + \frac{1}{\rho_\mathrm{f}}\frac{\partial}{\partial x_i}\left(\bar{\tau}_{\mathrm{f,ij}} - \boxed{\rho_\mathrm{f}\overline{u'_\mathrm{i}u'_\mathrm{j}}}\right) \end{aligned} \tag{2.23}$$

The equations in (2.23) are written only with respect to the mean components. Thus, it can be deduced that for a RANS approach, only the mean values of the fluid fields are calculated. Comparing the momentum equations in (2.23) to the ones in (2.16), it can be observed that for the former formulation an additional term appears. This term is boxed at the bottom of (2.23), and it is called the Reynolds stress tensor, $\boldsymbol{R}$. Within $\boldsymbol{R}$, 6 independent terms are found, which constitute additional unknowns to the fluid problem. Thus, to model them, additional conditions need to be added to the problem. These new mathematical equations form the turbulence model that is used for the RANS approach.

For this thesis, the SST $k-\omega$ turbulence model is used [75], based on the recommendations of van den Bos [38]. This is a member of the larger family of eddy viscosity models, which are based on the Boussinesq hypothesis, that is, modelling the Reynolds stress tensor using a similar approach to that of the viscous stresses:

$$-\rho_\mathrm{f}\overline{u'_\mathrm{i}u'_j} = 2\mu_\mathrm{t}\bar{\boldsymbol{S}}_{ij} - \frac{2}{3}\delta_{ij}\rho_\mathrm{f}k, \tag{2.24}$$

where μ_t represents the scalar fictitious eddy viscosity, $\bar{\boldsymbol{S}}_{ij} = \frac{1}{2}\left(\frac{\partial \bar{u}_\mathrm{i}}{\partial x_j} + \frac{\partial \bar{u}_\mathrm{j}}{\partial x_i}\right) - \frac{1}{3}\delta_{ij}\frac{\partial \bar{u}_k}{\partial x_k}$ is the strain rate, k is the turbulent kinetic energy, and δ_{ij} is the Kronecker delta operator. To obtain the unknowns k and μ_t in (2.24), two additional transport equations are solved by the SST $k-\omega$ model, as presented in (2.25) and (2.26). Using the additional relation (2.27), both k and μ_t can be determined. For clarity, the constants in (2.25)-(2.27) are also discussed. $C_\mu, \sigma_\mathrm{k}, \alpha_\mathrm{t}, \beta_\mathrm{t}, \sigma_\mathrm{t}, \sigma_{\omega 2}$, and a_1 represent model constants, that are not dependent on the spatial location of the cell that is considered in the fluid domain. F_1 and F_2 represent blending functions that are dependent, amongst others, on the wall distance [75]. In particular for F_1, near the wall, its value is equal to 1, such that (2.26) corresponds to the $k-\omega$ turbulence model [76]. Far from the wall, however, $F_1 = 0$, and the formulation of the transport equation becomes analogous to that of the $k-\epsilon$ model [77].

$$\rho_\mathrm{f}\frac{\partial k}{\partial t} + \rho_\mathrm{f}\bar{u}_\mathrm{i}\frac{\partial k}{\partial x_i} = -\rho_\mathrm{f}\overline{u'_\mathrm{i}u'_j}\frac{\partial \bar{u}_\mathrm{j}}{\partial x_i} - \rho_\mathrm{f}C_\mu\omega k + \frac{\partial}{\partial x_i}\left[\left(\mu + \frac{\mu_\mathrm{t}}{\sigma_\mathrm{k}}\right)\frac{\partial k}{\partial x_i}\right] \tag{2.25}$$

$$\frac{\partial \omega}{\partial t} + \rho_\mathrm{f}\bar{u}_\mathrm{i}\frac{\partial \omega}{\partial x_i} = -\rho_\mathrm{f}\overline{u'_\mathrm{i}u'_j}\alpha_\mathrm{t}\frac{\omega}{k}\frac{\partial \bar{u}_\mathrm{j}}{\partial x_i} - \beta_\mathrm{t}\rho_\mathrm{f}\omega k + \frac{\partial}{\partial x_i}\left[\left(\mu + \sigma_\mathrm{t}\mu_\mathrm{t}\right)\frac{\partial \omega}{\partial x_i}\right] + 2(1-F_1)\frac{\rho_\mathrm{f}\sigma_{\omega 2}}{\omega}\frac{\partial \omega}{\partial x_j}\frac{\partial k}{\partial x_j} \tag{2.26}$$

$$\mu_\mathrm{t} = \frac{a_1\rho_\mathrm{f}k}{\max\left(a_1\omega, S_{ij}F_2\right)} \tag{2.27}$$

2.2.3. AniPFM

Given the limitations of a URANS fluid model in accurately simulating TIV phenomena, the AniPFM synthetic turbulence model was developed by NRG. This was created such that it maintains the properties of turbulence (chaotic, multiple-scale phenomenon, that is statistically stable, and which is characterized by rotationality and tridimensionality, but also by its dissipative and diffusive nature) [38]. The method consists of multiple steps: generating the non-dimensional velocity fluctuations, correlating them with the previous fluctuations, and finally modifying the outcome to obtain the anisotropic velocity fluctuations. These are presented separately in Subsections 2.2.3.1-2.2.3.3. Once the velocity fluctuations are known, they can be used to compute the pressure fluctuations using a Poisson equation [38].

In this Subsection, a brief overview of the mathematical model of AniPFM is offered, without presenting all of the justifications regarding the choice made for the empirical relations presented. The reader interested in such details is referred to the works of van den Bos [38], and of Freitas [41].

2.2.3.1. Generation of the isotropic non-dimensional velocity fluctuations

The first step of the AniPFM model is to generate the non-dimensional velocity fluctuations, $\boldsymbol{w_t}$, for the current time instance. Starting from the turbulence spectrum that is assumed for the flow, the fluctuations are generated by using a Fourier decomposition, that is, the turbulent perturbations are modelled as a superposition of sinusoids along random directions inside the domain. Thus, the non-dimensional velocity fluctuations can be written as [38]:

$$\boldsymbol{w_t}(\boldsymbol{x}) = \sqrt{6} \sum_{n}^{N_{\text{modes}}} \sqrt{q_{\text{t,n}}} \left[\boldsymbol{\sigma}_{\mathbf{t},\text{n}} \cos\left(\boldsymbol{k}_{\mathbf{t},\text{n}} \cdot \boldsymbol{x} + \phi_{\text{t,n}} \right) \right], \tag{2.28}$$

where $\boldsymbol{x}$ is a random location inside the fluid domain, q represents the scaled energy that is allocated to the n$^{\text{th}}$ wavenumber inside the turbulence spectrum, $\boldsymbol{\sigma_t}$ is the direction vector along which the perturbation is generated, $\boldsymbol{k}$ is the wavenumber vector, and ϕ a uniformly sampled random value of the phase angle with values between 0 and 2π. At this point, to fully define how $\boldsymbol{w_t}$ is computed, it is required to describe how $q_{\text{t,n}}, \boldsymbol{\sigma_{t,n}}$ and $\boldsymbol{k}_{\mathbf{t},\text{n}}$ are computed. This is done in the remainder of this Subsection.

To obtain the wavenumber vectors $\boldsymbol{k}_{\mathbf{t},\text{n}}$, one must first understand how the wavenumbers are obtained, and how they are subsequently transformed into vectors. First, the generation of the wavenumber interval is presented. The smallest considered wavenumber, $k_{\text{t,start}}$, corresponds to the largest scale in the flow. It is obtained by taking three parameters into account: a conservative estimation of the largest eddy length scales based on the turbulent dissipation rate ε_{t} and the largest velocity component, equation (2.29), the requirement that the peak of the energy spectrum $k_{\text{t,e}}$ is higher than $k_{\text{t,start}}$, and a user-inputted length-scale l_{user} that is a consequence of the flow geometry. Based on these, $k_{\text{t,start}}$ is computed using equation (2.30). The largest wavenumber value, $k_{\text{t,end}}$, could be immediately associated with the Kolmogorov length scales. However, in practice, given the low resolution of the mesh and the ensuing risk of aliasing for high values of the wavenumber [38], $k_{\text{t,end}}$ is obtained by considering a lower cut-off wavenumber, $k_{\text{t,cut}}$, where the latter is defined in (2.31), and the relation between the two is given in (2.32). For computing $k_{\text{t,cut}}$ using (2.31), the AniPFM user can choose to compute the cut-off length $l_{\text{t,cut}}$ using either the Shur criterion, (2.33), or the cubic one, (2.34) [41]. In (2.33), h_y and h_z refer to the coordinates of the cell in the transverse directions of the flow, $h_{\max}$ to the maximum height of the channel flow, and d_{w} to the smallest distance to a wall. In (2.34), V_{cell} corresponds to the volume of the discretization cell for which the cut-off length is calculated.

$$k_{\text{t,start}} = \frac{\varepsilon_{\text{t}}}{\max\left(\|\boldsymbol{U_f}\|^3 \right)}, \tag{2.29}$$

$$k_{\text{t,start}} = \max\left[\min\left(k_{\text{t,start}}, \frac{1}{2} k_{\text{t,e}} \right), \frac{2\pi}{l_{\text{user}}} \right]. \tag{2.30}$$

$$k_{\text{t,cut}} = \frac{2\pi}{l_{\text{t,cut}}} \tag{2.31}$$

$$k_{\text{t,end}} = \frac{3}{2} k_{\text{t,cut}}. \tag{2.32}$$

$$\text{Shur cut-off: } l_{\text{t,cut}} = 2\min\left\{ \left[\max\left(h_y, h_z, 0.3 h_{\max} \right) + 0.1 d_{\text{w}} \right], h_{\max} \right\} \tag{2.33}$$

$$\text{Cubic cut-off: } l_{\text{t,cut}} = 2 V_{\text{cell}}^{\frac{1}{3}} \tag{2.34}$$

Starting from the definitions of $k_{\text{t,start}}$ and $k_{\text{t,end}}$, the ensuing interval is split using $N+1$ logarithmically distributed edge-wavenumbers. Those represent the ends of the discretized wavenumber bands being considered. The wavenumber associated with $\boldsymbol{k}_{\mathbf{t},\text{n}}$ in equation (2.28) is equal to the mean of the said $n-1$ and n edge-wavenumbers. To obtain the wavenumber vector, equation (2.35) is used, where random values of the direction angles θ_n and ψ_n are used, with probabilities defined in (2.36). It is worth noting that the same sequence of random numbers that are generated at different timesteps can be reproduced if one uses the same settings for the numerical model, including a random seed number within the AniPFM model.

$$\mathbf{k}_{\mathbf{t},\text{n}} = k_{\text{t,n}} \left[\sin\theta_{\text{n}}\cos\psi_{\text{n}}, \sin\theta_{\text{n}}\sin\psi_{\text{n}}, \cos\theta_{\text{n}}\right]^T, \tag{2.35}$$

$$P\left(\psi_{\text{n}}\right) = \frac{1}{\pi}, \quad P\left(\theta_{\text{n}}\right) = \frac{1}{2}\sin\left(\theta_{\text{n}}\right). \tag{2.36}$$

The value of the scaled energy $q_{\text{t,n}}$ associated with the wavenumber $k_{\text{t,n}}$ is obtained by using a scaled Von-Kármán spectrum:

$$q_{\text{t,n}} = \frac{E_k\left(k_{\text{t,n}}\right)\Delta k_{\text{t,n}}}{\sum_n^N E_k\left(k_{\text{t,n}}\right)\Delta k_{\text{t,n}}}, \text{ where } E(k) = \frac{\left(k/k_{\text{t,e}}\right)^4}{\left[1+2.4\left(k/k_{\text{t,e}}\right)^2\right]^{(17/6)}} \exp\left(-\left(12\frac{k}{k_\eta}\right)^2\right) f_{\text{cut}}(k). \tag{2.37}$$

In (2.37), $\Delta k_{\text{t,n}}$ is equal to the length of the wavenumber bands defined by the logarithmically distributed edge-wavenumbers, and it is therefore already known. Thus, to compute $q_{\text{t,n}}$, it is required to determine the peak wavenumber $k_{\text{t,e}}$, the Kolmogorov wavenumber k_η, and the cut-off frequency f_{cut}. The Kolmogorov wavenumber is calculated using dimensional analysis and the assumption that the size of the smallest eddies is dependent on only the turbulent dissipation ε_{t}, and the fluid's kinematic viscosity, ν_f, thus obtaining (2.38). The computation of the cut-off frequency f_{cut} is based on the local value of the cut-off wavenumber $k_{\text{t,cut}}$ and the considered wavenumber, where the link between the two is provided in (2.39). $k_{\text{t,e}}$ is computed using an empirical relationship that is based on multiplying the local turbulent lengthscale l_{t} with a constant C_{l}, equation (2.40). Though initial studies proposed that the value of C_{l} is equal to 3.0 [38], Freitas [41] found that if one uses a Shur cut-off length, (2.33), better accuracy is obtained for the brass beam case if using $C_{\text{l}} = 2.0$, and $C_{\text{l}} = 0.6$ for the cubic cut- off length given in (2.34).

$$k_\eta = 2\pi\left(\frac{\varepsilon_{\text{t}}}{\nu_f^3}\right)^{\frac{1}{4}}, \tag{2.38}$$

$$f_{\text{cut}}(k) = \exp\left(-\left[\frac{4\max\left(k-0.9k_{\text{t,cut}},0\right)}{k_{\text{t,cut}}}\right]^3\right), \tag{2.39}$$

$$k_{\text{t,e}} = \frac{2\pi}{l_e} = \frac{2\pi}{C_{\text{l}}l_{\text{t}}}. \tag{2.40}$$

Finally, the direction vector $\boldsymbol{\sigma}_{\mathbf{t},\text{n}}$ is discussed. To ensure a divergence-free velocity fluctuation field, it is required that $\boldsymbol{k}_{\mathbf{n}} \cdot \boldsymbol{\sigma}_{\mathbf{t},\text{n}} = 0$, for all modes considered for the velocity fluctuations. To satisfy this condition, $\boldsymbol{\sigma}_{\mathbf{t},\text{n}}$ is obtained as the cross product between $\boldsymbol{k}_{\mathbf{n}}$, as given by (2.35), and a random direction $\boldsymbol{\zeta}_{\text{n}}$, as defined in (2.41), where the angles θ_ζ and ψ_ζ represent independent realizations of the same probability distributions that were also used for ψ and θ in (2.36).

$$\boldsymbol{\zeta}_{\text{n}} = \left[\sin\theta_\zeta\cos\psi_\zeta, \sin\theta_\zeta\sin\psi_\zeta, \cos\theta_\zeta\right]^T, \tag{2.41}$$

$$\boldsymbol{\sigma}_{\mathbf{t},\text{n}} = \frac{\boldsymbol{\zeta}_{\text{n}} \times \boldsymbol{k}_{\text{n}}}{\left|\boldsymbol{\zeta}_{\text{n}} \times \boldsymbol{k}_{\text{n}}\right|}. \tag{2.42}$$

2.2.3.2. Time correlation

The information presented in Subsection 2.2.3.1 fully defines the computation of the non-dimensional velocity fluctuations $\boldsymbol{w}_{\mathbf{t}}$. These are generated at each timestep, and they encapsulate most of the physical characteristics associated with turbulence. Nevertheless, there is also a dynamic component associated with turbulent phenomena, that cannot be captured by only considering the fluctuations as independent occurences at given time instances. More specifically, the turbulent behaviour is influenced by both the transport of eddies inside the fluid, and by the continuous dissipation and production of turbulent kinetic energy [38]. To model the influence of these factors, four different time propagation methods were put forward by Freitas [41] and by van den Bos [38]. For this thesis, only the Convection & Exponential Correlation (C&EC) method was used, since this was found by Freitas [41] to offer a vibratory behaviour that is closer to experimental observed.

More specifically, he found that when using C&EC, similar amplitudes of the vibrations in both bending directions were predicted when considering the NRG-FSIFOAM workflow.

The C&EC method consists of two steps. First, the convection of the velocity fluctuations obtained at the previous timestep, $\mathbf{v_t}^{\text{m-1}}$, is computed using equation (2.43), where U_x, U_y, U_z refer to the local mean-velocity components that were computed using URANS. The second step is to linearly combine the convected values of $\mathbf{v_t}^{\text{m-1}}$ with the newly generated non-dimensional velocity fluctuations, as is done in equation (2.44). The scalar constants a and b in (2.44) are computed using (2.45) and (2.46), where Δt is equal to the fluid timestep, τ_t represents the characteristic timescale of the flow, and f_{τ_t} is a user-inputted coefficient.

$$\frac{\partial \mathbf{v_t}^{\text{m-1}}}{\partial t} + U_x \frac{\partial \mathbf{v_t}^{\text{m-1}}}{\partial x} + U_y \frac{\partial \mathbf{v_t}^{\text{m-1}}}{\partial y} + U_z \frac{\partial \mathbf{v_t}^{\text{m-1}}}{\partial z} = 0 \tag{2.43}$$

$$\mathbf{v}_t^m(\boldsymbol{x}, t) = a_\text{t} \mathbf{v_t}^{\text{m-1}}(\boldsymbol{x}) + b_\text{t} \boldsymbol{w_t}^{\text{m}}(\boldsymbol{x}) \tag{2.44}$$

$$a_\text{t} = e^{-f_\tau \Delta t / \tau_\text{t}} \tag{2.45}$$

$$b_\text{t} = \sqrt{1 - a_\text{t}^2} \tag{2.46}$$

2.2.3.3. Generation of the anisotropic velocity fluctuations

Using the information presented in Subsection 2.2.3.3, one can obtain the *isotropic* velocity fluctuations at a given time instance. However, this assumption is not valid for regions that are adjacent to solid walls, and which are actually responsible for the excitation of the structures in TIV applications [38]. Thus, it was proposed by van den Bos [38] to consider anisotropy for the velocity fluctuations by using the Reynolds stress tensor $\boldsymbol{R}$. More specifically, the Cholesky decomposition of $\boldsymbol{R}$, defined in (2.47), is used. Thus, the anisotropic velocity fluctuations, $\mathbf{u_t}$, are computed using (2.48).

$$\boldsymbol{a_t}^T \boldsymbol{a_t} = \boldsymbol{R} \rightarrow \boldsymbol{a_t} = \begin{bmatrix} \sqrt{\boldsymbol{R}_{11}} & 0 & 0 \\ \boldsymbol{R}_{21} / \boldsymbol{a}_{\mathbf{t},11} & \sqrt{\boldsymbol{R}_{22} - \boldsymbol{a}_{\mathbf{t},21}^2} & 0 \\ \boldsymbol{R}_{31} / \boldsymbol{a}_{\mathbf{t},11} & \left(\boldsymbol{R}_{32} - \boldsymbol{a}_{21} \boldsymbol{a}_{\mathbf{t},31}\right) / \boldsymbol{a}_{\mathbf{t},22} & \sqrt{\boldsymbol{R}_{33} - \boldsymbol{a}_{\mathbf{t},31}^2 - \boldsymbol{a}_{\mathbf{t},32}^2} \end{bmatrix} \tag{2.47}$$

$$\mathbf{u_t}(\boldsymbol{x}, t) = \boldsymbol{a_t} \mathbf{v_t}(\boldsymbol{x}, t) \tag{2.48}$$

2.3. FEM

Before summarizing the general approach behind FEM, an introduction to the general problem is presented. First, in this paragraph, a simplistic representation is given of the structural problem that is desired to be modelled. In broad terms, the developed structural model is going to be used for the study of flow-induced vibrations in axial flow of nuclear fuel rods in PWR nuclear reactors. These components are characterized by their slenderness, having lengths of about 4 meters, and a diameter of approximately 1 centimeter [78], and are subjected to dynamic excitation by the water flow which absorbs the energy released by the nuclear reaction. In terms of support, the nuclear rods are fixed at both their ends [79]. Along their length, they present multiple pinned supports, in the form of spacer grids [79]. Under these assumptions, a simplified representation of the mechanical system to be studied is presented at the top of Figure 2.2.

Apart from the support conditions and the shape of the structure, to uniquely solve the structural problem one should know the loads applied onto the structure, its constitutive equation, as well as the initial conditions. In terms of the applied forces, as pointed out in Section 2.1, for the considered problem the external forcing is represented by the fluid's distributed loads. In the following, these are denoted as $\boldsymbol{f_S}$, having a unit equal to force per area. At the same time, volumetric body forces $\boldsymbol{f_B}$ should also be included. Since the timescales of the turbulence developed by the water are similar to those of the structure's vibrations, one must consider inertial effects for the structural analysis. In fact, the inertial forces represent the only relevant body forces for the considered application, and their mathematical model will be introduced in Subsection 2.3.1. For now, a schematic representation of the surface and body forces within the context of the general solid continuum problem is presented on the bottom left of Figure 2.2. Finally, the constitutive equation represents the relation between the tensions $\boldsymbol{\sigma_s}$ and the strains $\boldsymbol{\epsilon_s}$, where the two are defined locally. The strains are defined by differentiating the considered displacements with respect to the chosen set of coordinates [80]. Given the small relative amplitude of the vibrations, the strain-stress relation can be assumed to be linear [81].

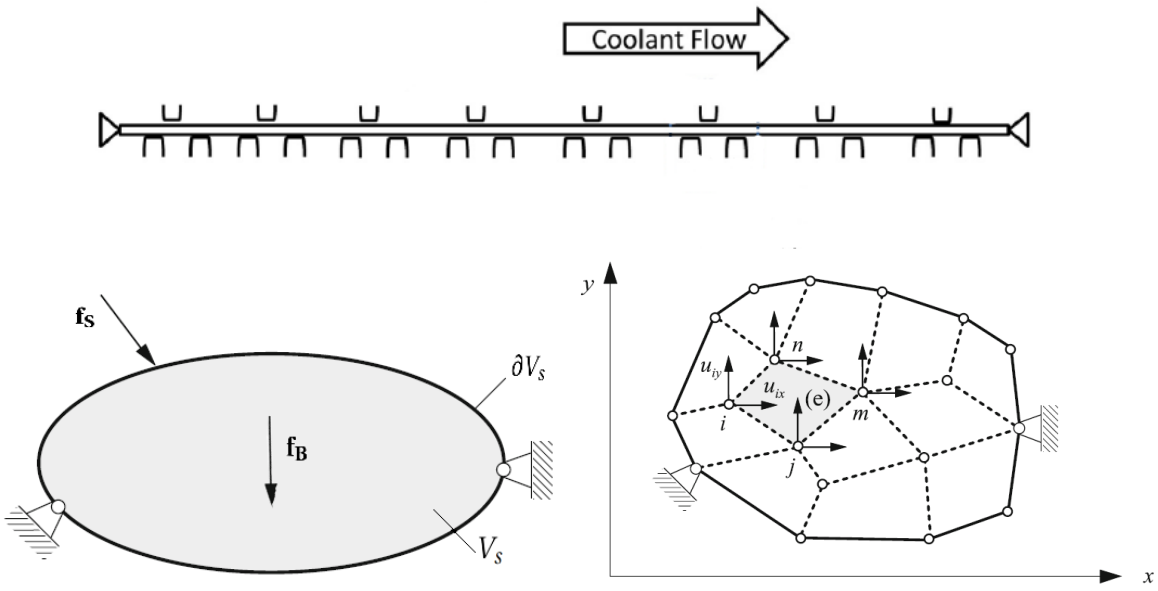


Figure 2.2: Top: Simplified structural model for an isolated nuclear fuel rod (taken from [33]); Bottom: schematic representation of the solid continuum problem (left), and of the corresponding FEM methodology (right) (taken from [80])

Having described the above properties of the structural problem being analyzed, the objective of the structural model can be reduced to [81]: given the solid's external forcing, geometry, support conditions, stress-strain law, and initial state, determine the corresponding structural displacement field throughout the solid continuum, $\boldsymbol{u}_{\mathbf{s,cont}}$. The FEM approach to this problem will be presented in detail in the following Subsection.

2.3.1. General approach

Starting from the problem formulated at the end of the introduction to Section 2.3, the FEM approach divides the solid continuum into a given number of elements, as seen on the bottom right of Figure 2.2. Subsequently, based on the expected load characteristics, the deformation of the nodes is described using a predefined set of displacements that is purported to accurately describe the deformed state (for example, on the bottom right of Figure 2.2, the vertical and horizontal translations are used). The solid continuum problem is then solved only for the combined displacements at all of the nodes, $\boldsymbol{u}$. To interpolate the nodal solution at any location $(x_\mathrm{i}, y_\mathrm{i}, z_\mathrm{i})$ inside the i$^{\mathrm{th}}$ element, denoted in the following as $\boldsymbol{u}_{\mathbf{s,cont},\,\mathrm{i}}$, shape functions are defined for each element [81]:

$$\boldsymbol{u}_{\mathbf{s,cont},\,\mathrm{i}}(x_\mathrm{i}, y_\mathrm{i}, z_\mathrm{i}) = \boldsymbol{N}_\mathrm{i}(x_\mathrm{i}, y_\mathrm{i}, z_\mathrm{i})\ \boldsymbol{u}_\mathbf{s}, \tag{2.49}$$

where $\boldsymbol{N}_\mathrm{i} \in \mathbb{R}^{3\times N_\mathrm{s}}$ and $\boldsymbol{u}_\mathbf{s} \in \mathbb{R}^{N_\mathrm{s}}$, with N_s representing the number of degrees of freedom used to describe the problem. Based on the choice of shape functions in (2.49), one can also redefine the strains. For a solid continuum, the strains are obtained by applying differentiation operators to the displacements [80]. The same differentiation operators can be applied to the shape functions in (2.49). In doing so, a discrete differentiation matrix $\boldsymbol{B}_\mathrm{i}(x_\mathrm{i}, y_\mathrm{i}, z_\mathrm{i}) \in \mathbb{R}^{N_\mathrm{s}\times N_\mathrm{s}}$ is obtained [81]. Hence, the shape functions lead to a discretized form of the strain vector as [81]:

$$\boldsymbol{\epsilon}_{\mathbf{s,cont},\mathrm{i}}(x_\mathrm{i}, y_\mathrm{i}, z_\mathrm{i}) = \boldsymbol{B}_\mathrm{i}(x_\mathrm{i}, y_\mathrm{i}, z_\mathrm{i})\boldsymbol{u}_\mathbf{s}, \tag{2.50}$$

where $\boldsymbol{\epsilon}_{\mathbf{s,cont},\mathrm{i}} \in \mathbb{R}^{N_\mathrm{s}}$. Next, the FEM makes use of the principle of virtual displacements. This represents a necessary condition for any structural equilibrium solution, and it stipulates that the virtual external work done by any arbitrary small displacement field that is zero at the location of the nodal points is equal to the virtual internal work done by the structure [81]:

$$\int_{V_\mathrm{s}} \delta\boldsymbol{\epsilon}_\mathbf{s}^T \boldsymbol{\sigma}_{\mathbf{s},\,\mathrm{i}}\, dV = \int_{V_\mathrm{s}} \delta\boldsymbol{u}_{\mathbf{s,cont}}^T \boldsymbol{f_B}\, dV + \int_{\partial V_\mathrm{s}} \delta\boldsymbol{u}_{\mathbf{s,cont}}^T \boldsymbol{f}_\mathbf{S}\, dS, \tag{2.51}$$

where the left side of (2.51) represents the internal work, while the right the external one. Further, the δ notation is used for virtual quantities. Since the FEM discretization covers the entire solid volume, the integral

identity (2.51) can be written by considering the integration for every finite element [81]:

$$\sum_{\mathrm{i}} \int_{V_{\mathrm{s,i}}} \delta \boldsymbol{\epsilon}_{\mathbf{s},\mathrm{i}}^{T} \boldsymbol{\sigma}_{\mathbf{s},\,\mathrm{i}}\, dV = \sum_{\mathrm{i}} \int_{V_{\mathrm{s,i}}} \delta \boldsymbol{u}_{\mathbf{s,cont}}^{T} \boldsymbol{f}_{\mathbf{B},\mathrm{i}}\, dV + \sum_{\mathrm{i}} \int_{\partial V_{\mathrm{s,i}}} \boldsymbol{u}_{\mathbf{s,cont}}^{T} \boldsymbol{f}_{\mathbf{S},\mathrm{i}}\, dS \tag{2.52}$$

Further, similar to (2.49) and (2.50), one can use the chosen set of shape functions to rewrite the virtual strain and displacements fields in (2.52), as can be seen in (2.53) and (2.54). Finally, since only linear, initially non-deformed states are used for the structural modelling in this work, the tensions within each element can be considered linear with respect to the strains, and equal to 0 when $\boldsymbol{\epsilon}_{\mathbf{s}} = \mathbf{0}$, leading to equation (2.55).

$$\boldsymbol{u}_{\mathbf{s,\,cont},\mathrm{i}}(x_{\mathrm{i}}, y_{\mathrm{i}}, z_{\mathrm{i}}) = \boldsymbol{N}_{\mathrm{i}}(x_{\mathrm{i}}, y_{\mathrm{i}}, z_{\mathrm{i}}) \delta \boldsymbol{u}_{\mathbf{s}}, \tag{2.53}$$

$$\delta \boldsymbol{\epsilon}_{\mathbf{s},\mathrm{i}} = \boldsymbol{B}_{\mathrm{i}}(x_{\mathrm{i}}, y_{\mathrm{i}}, z_{\mathrm{i}}) \delta \boldsymbol{u}_{\mathbf{s}}, \tag{2.54}$$

$$\boldsymbol{\sigma}_{\mathbf{s},\mathrm{i}} = \boldsymbol{C}_{\mathrm{i}} \boldsymbol{\epsilon}_{\mathbf{s},\mathrm{i}}. \tag{2.55}$$

Replacing (2.53)-(2.55) into (2.52), one obtains [81]:

$$\delta \boldsymbol{u}_{\mathbf{s}}^{T} \left[\sum_{\mathrm{i}} \int_{V_{\mathrm{s,i}}} \boldsymbol{B}_{\mathrm{i}}^{T} \boldsymbol{C}_{\mathrm{i}} \boldsymbol{B}_{\mathrm{i}}\, dV \right] \boldsymbol{u}_{\mathbf{s}} = \delta \boldsymbol{u}_{\mathbf{s}}^{T} \left[\sum_{\mathrm{i}} \int_{V_{\mathrm{s,i}}} \boldsymbol{N}_{\mathrm{i}}^{T} \boldsymbol{f}_{\mathbf{B},\mathrm{i}}\, dV + \sum_{\mathrm{i}} \int_{\partial V_{\mathrm{s,i}}} \boldsymbol{N}_{\mathrm{i}}^{T} \boldsymbol{f}_{\mathbf{S},\,\mathrm{i}}\, dS \right] \tag{2.56}$$

At this point, it is worth remembering that the body forces are given only by the inertial forces of the structure. Consequently, according to d'Alembert's principle, $\sum_{\mathrm{i}} \int_{V_{\mathrm{s,i}}} \boldsymbol{N}_{\mathrm{i}}^{T} \boldsymbol{f}_{\mathbf{B},\mathrm{i}}\, dV = -\sum_{\mathrm{i}} \int_{V_{\mathrm{s,i}}} \boldsymbol{N}_{\mathrm{i}}^{T} \left(\rho_{\mathrm{s,i}} \boldsymbol{N}_{\mathrm{i}} \ddot{\boldsymbol{u}}_{\mathbf{s}} \right) dV$, such that (2.56) becomes [81]:

$$\delta \boldsymbol{u}_{\mathbf{s}}^{T} \left[\sum_{\mathrm{i}} \int_{V_{\mathrm{s,i}}} \rho_{s,i} \boldsymbol{N}_{\mathrm{i}}^{T} \boldsymbol{N}_{\mathrm{i}}\, dV \right] \ddot{\boldsymbol{u}}_{\mathbf{s}} + \delta \boldsymbol{u}_{\mathbf{s}}^{T} \left[\sum_{\mathrm{i}} \int_{V_{\mathrm{s,i}}} \boldsymbol{B}_{\mathrm{i}}^{T} \boldsymbol{C}_{\mathrm{i}} \boldsymbol{B}_{\mathrm{i}}\, dV \right] \boldsymbol{u}_{\mathbf{s}} = \delta \boldsymbol{u}_{\mathbf{s}}^{T} \left[\sum_{\mathrm{i}} \int_{\partial V_{s,i}} \boldsymbol{N}_{\mathrm{i}}^{T} \boldsymbol{f}_{\mathbf{S},\,\mathrm{i}}\, dS \right]. \tag{2.57}$$

If one successively uses the unit vector for each entry of $\delta \boldsymbol{u}_{\mathbf{s}}$ in (2.57), then the familiar system (2.58) is obtained [81]:

$$\boldsymbol{M}_{\mathbf{s}} \ddot{\boldsymbol{u}}_{\mathbf{s}} + \boldsymbol{K}_{\mathbf{s}} \boldsymbol{u}_{\mathbf{s}} = \boldsymbol{f}_{\mathbf{s}}(t), \tag{2.58}$$

where

$$\boldsymbol{M}_{\mathbf{s}} = \sum_{\mathrm{i}} \int_{V_{\mathrm{s,i}}} \rho_{s,i} \boldsymbol{N}_{\mathrm{i}}^{T} \boldsymbol{N}_{\mathrm{i}}\, dV, \tag{2.59}$$

$$\text{and } \boldsymbol{K}_{\mathbf{s}} = \sum_{\mathrm{i}} \int_{V_{\mathrm{s,i}}} \boldsymbol{B}_{\mathrm{i}}^{T} \boldsymbol{C}_{\mathrm{i}} \boldsymbol{B}_{\mathrm{i}}\, dV \tag{2.60}$$

represent the global mass and stiffness matrices, respectively. In practice, though the formal definition of $\boldsymbol{K}_{\mathbf{s}}$ is used to compute the global stiffness matrix, for the mass matrix, a lumped approach is often preferred. The lumped approach consists of distributing the total inertia among a subset of the degrees of freedom, instead of rigorously computing the integrals in (2.59), and it leads to an easily invertible form of $\boldsymbol{M}_{\mathbf{s}}$. Though this last characteristic is useful for the computational efficiency of the time marching of the structural problem [81], it also leads to a worse prediction of the eigenfrequencies of the structure for a given number of elements [82]. For this study, a high accuracy in determining the eigenfrequencies are important both for the transfer of energy through TIV [83], as well as for the ROM methods presented in Section 2.4. Within the literature, a mass matrix calculated using (2.59) is known as consistent, since the same shape functions are used to define it as for the stiffness matrix. Hence, a consistent mass matrix approach will be used in this thesis.

It is also worth noting that in (2.58) no damping matrix is present. This is because for this research the structural damping is purposefully neglected. It is expected that for axial flow TIV in nuclear applications the fluid damping would be much larger than the structural one [38]. This assumption is also consistent with previous work within NRG on the topic of axial-flow TIV [38, 41].

Finally, throughout this Subsection, according to (2.49) and (2.50), it was assumed that the displacement and strain fields are obtained by considering the entire sets of displacements $\boldsymbol{u}_{\mathbf{s}}$ and strains $\boldsymbol{\epsilon}_{\mathbf{s}}$, respectively. However, in reality, the fields within the i^{th} element are written only with respect to the local nodes on the boundary of that particular element, resulting in a smaller size of the $\boldsymbol{N}$ and $\boldsymbol{B}$ vectors. Consequently, mass and stiffness matrices are obtained for each element, their dimensions equal to the total number of degrees of freedom that the bounding nodes possess. However, to obtain the global system in (2.58), it is required that

the local matrices are aligned with the global reference system, and that each local node is assigned a unique global numbering through a connectivity matrix [81]. In doing so, procedures can be formulated that allow each local degree of freedom's inertia and stiffness to be associated with the correct global degree of freedom. The technique chosen for the formation of the global system is briefly described in Subsection 3.4.2.

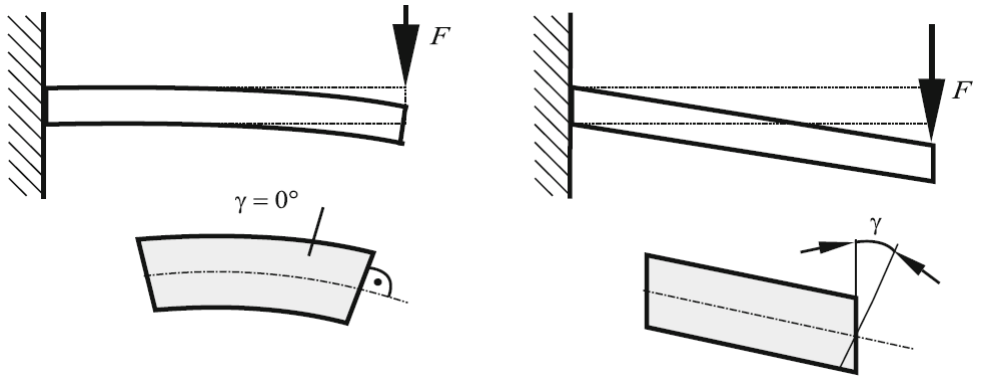


Figure 2.3: The different deformation assumptions for the Bernoulli (left) and the Timoshenko (right) beam elements, in terms of the relative movement of the cross-sections with respect to the neutral axis during bending; for this study, the Bernoulli model is considered; taken from [80]

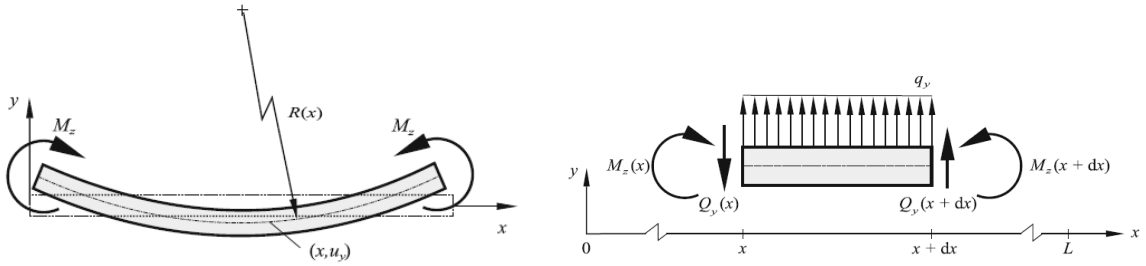


Figure 2.4: The infinitesimal model used to build the kinematic (left) and the equilibrium (right) conditions for the Euler beam model; taken from [80]

2.3.2. Beam elements

Beam elements represent a particular type of finite elements that can be used to construct the discretized system (2.58). They are particularly suitable for applications where the structure is slender and quasi-cylindrical, that is, where the variation of the cross-sectional shape and area in the axial direction is relatively small [84]. Furthermore, the beam element framework assumes that the solid is predominantly subjected to bending. All of these conditions are fulfilled by nuclear fuel rods, which have an aspect ratio $\frac{L}{D} \approx 400$, and which have a non-varying, circular cross-section at any axial position. Based on the assumed deformation behaviour of the structure during bending, within literature two types of bending elements can be identified: Bernoulli and Timoshenko, whose defining modelling assumptions can be visualised in Figure 2.3. More specifically, the two are distinguished based on whether the cross-section remains perpendicular to the neutral axis during bending: the Bernoulli element (left in Figure 2.3) does, while the Timoshenko one (right in Figure 2.3) doesn't [84]. The two are also known within the literature as the shear rigid and the shear flexible elements, respectively [80]. For the topic of this research, Bernoulli elements will be used. This is because nuclear fuel rods can be safely considered shear rigid, given the fact that they are thick-walled beams made out of rigid steel [79]. Though in principle one could have also used the Timoshenko elements to model the bending of the fuel elements, using a Bernoulli element approach is justified by the fact that no shear degrees of freedom need to be allocated. This leads to a reduction in the number of degrees of freedom considered for each node, and therefore to a reduction in the size of the structural problem (2.58) that needs to be solved for a given level of discretization. Hence, for the remainder of this work, only the Bernoulli beam model will be developed, along with the corresponding element theory. Though in determining the solution of a structural problem one is generally interested in both the displacements and the stresses, in the following, only the ways to determine

the latter are explained: for GTRF, one is particularly worried by the amplitudes of vibration [85], rather than the high stresses that could be found locally within the structure.

2.3.3. The Bernoulli beam model

Before the FEM model that is used to actually construct the global problem (2.58) is presented, the underlying deformation model for a single, continuous bending beam according to the Bernoulli model is presented. The assumptions of the Bernoulli beam model are:

1. The structure exhibits a linear-elastic behaviour, that is, the structural response is directly proportional to the applied stresses; this implies relatively small lateral displacements of the structures when compared to the cross-sectional area (or the length of the structure)
2. The cross-sections of the structure remain perpendicular to the neutral axis of the rod at all time instances at which the structure is deformed, as seen on the left of Figure 2.3
3. Throughout the deformation, the cross-sections remain plane and they maintain their cross-sectional contour (no warping) [80]

Under these assumptions, one can also conclude that the deformed state of the tridimensional structure can be described using the deformation of its unidimensional neutral axis, along with the normal cross-sections whose shape remains constant at all times. Throughout the deformation, the cross-sections move rigidly based on the translations and the rotations calculated at their corresponding axial position.

The end goal of the Bernoulli beam model is to determine the dependency between the external bending loads applied onto the solid, and the internal structural displacements of the rod. To this end, using the Bernoulli beam assumptions, one can determine the kinematics and equilibrium conditions of the model. In combination with the constitutive equation, the differential equation modelling fulfilling the purpose of the method is obtained. In the following, it is assumed that the $x-$ direction of the Cartesian system corresponds to the axial direction of the rod, and that the transverse translations due to the bending are in the $y-$ direction, and are denoted as $u_{s,y}$. Consequently, the local bending rotations are around the $z-$ axis, and are written $\varphi_{s,z}$. This one-plane bending model will be further extended to the $z-$ transverse translation in Subsection 2.3.5.

In terms of the kinematics, it is of interest to relate the axial strain of the beam, $\epsilon_{s,x}$, to the local transverse displacement $u_{s,y}$, that is, find a dependency $\epsilon_{s,x} = \epsilon_{s,x}(u_{s,y})$. This will be used directly within the constitutive equation (2.64). It is also important to link the bending rotations to the local bending translation, that is, to find a relation $\varphi_{s,z} = \varphi_{s,z}(u_{s,y})$. To do so, one can isolate a portion of the beam that is subjected only to pure bending. This is consistent with the assumption of a shear rigid element. The kinematic model can be seen on the left of Figure 2.4. By assuming small relative bending displacements and rotations ($u_{s,y} \ll L, \tan\varphi_{s,z} \approx \varphi_{s,z}$), it can be shown that [80]:

$$\epsilon_{s,x} = -y\frac{\mathrm{d}^2 u_{s,y}(x)}{\mathrm{d}x^2}, \qquad (2.61) \qquad\qquad \varphi_{s,z} = \frac{\mathrm{d}u_{s,y}(x)}{\mathrm{d}x}. \qquad (2.62)$$

Next, the equilibrium of the rod is computed, that is, the link between the externally applied transverse loads and the internal reactions generated inside the structure. For this, an infinitesimal element of the Bernoulli beam is analyzed while under a constant distributed load, $q_{s,y}$. The internal reactions take the form of a shear force, $Q_{s,y}$, and a bending moment, $M_{s,z}$, on each side of the infinitesimal element, that is, on both the positive and the negative face. For equilibrium, the internal reactions on the positive and negative face have opposite signs. The equilibrium model can be visualized on the right of Figure 2.4. By using a first-order model (approximating the internal reaction loads as $Q_{s,y}(x+\mathrm{d}x) \approx Q_{s,y}(x) + \frac{\mathrm{d}Q_{s,y}(x)}{\mathrm{d}x}\,\mathrm{d}x$, $M_{s,z}(x+\mathrm{d}x) \approx M_{s,z}(x) + \frac{\mathrm{d}M_{s,z}(x)}{\mathrm{d}x}\,\mathrm{d}x$, and by neglecting the second-order terms in $\mathrm{d}x$), the internal reactions are given by [80]:

$$\frac{\mathrm{d}^2 M_{s,z}(x)}{\mathrm{d}x^2} = -\frac{\mathrm{d}Q_{s,y}(x)}{\mathrm{d}x} = q_{s,y}. \qquad (2.63)$$

Having determined the connection between the external loads and the internal reaction forces in equation (2.63), the constitutive equations, linking the local axial stresses $\sigma_{s,x}$ with the axial strains $\epsilon_{s,x}$, will be used in combination with the kinematics to obtain the Bernoulli beam differential equation, that is, the link between the external loads and the structural displacements. Given that the Bernoulli beam model assumes small displacements, one can use Hooke's law as a constitutive law. In particular for this study, the structure is assumed to be isotropic, and one can write Hooke's law [80]:

$$\sigma_{s,x}(x) = E\,\epsilon_{s,x}(x), \tag{2.64}$$

where $\sigma_{s,x}$ represents the axial tension inside the rod, and E is a scalar equal to the structure's Young modulus. By replacing (2.64) in (2.61), one obtains [80]:

$$\sigma_{s,x}(x,y) = -Ey\frac{\mathrm{d}^2 u_{s,y}(x)}{\mathrm{d}x^2}. \tag{2.65}$$

Relation (2.65) describes the local tension at a given vertical distance from the neutral fiber of the beam during bending at the axial position x. If multiplied by the local distance from the rod's midline and integrated over the entire cross-section, one obtains the internal reaction moment of the entire section, $M_{s,z} = -\int_A y\sigma_{s,x}\,\mathrm{d}A$. Using this observation and (2.65), one obtains [80]:

$$M_{s,z} = -\int_A y\sigma_{s,x}\,\mathrm{d}A = \int_A yEy\frac{\mathrm{d}^2 u_{s,y}}{\mathrm{d}x^2}\,\mathrm{d}A, \text{ or} \tag{2.66}$$

$$M_{s,z} = EI_z\frac{\mathrm{d}^2 u_{s,y}}{\mathrm{d}x^2}, \tag{2.67}$$

where $I_z = \int_A y^2\,\mathrm{d}A$ is called the bending moment of inertia around the $z-$ axis. This is a characteristic of the cross-section of the structure, and for a circular rod, it is equal to $I_z = \frac{\pi D^4}{64}$. By using (2.67) in (2.63), one obtains the Bernoulli differential equation [80]:

$$\frac{\mathrm{d}^2}{\mathrm{d}x^2}\left(EI_z\frac{\mathrm{d}^2 u_{s,y}}{\mathrm{d}x^2}\right) = q_{s,y}. \tag{2.68}$$

In the case of this research, that is, for an isotropic beam of constant cross-section, E and I_z are constant, such that the Bernoulli beam equation becomes [80]:

$$\boxed{EI_z\frac{\mathrm{d}^4 u_{s,y}}{\mathrm{d}x^4} = q_{s,y}}. \tag{2.69}$$

To solve (2.69), one must also impose the boundary conditions that need to be applied at the ends of the rod. These will be explained in Subsection 2.3.4, once the concept of Bernoulli beam elements is introduced. Once (2.69) is solved for $u_{s,y}$, one can also obtain the bending rotation $\varphi_{s,z}$, by using the kinematic relation (2.62).

2.3.4. Bernoulli beam FEM elements

Subsection 2.3.3 presented the mathematical model that one would use to determine the structural translations and rotations $u_{s,y}(x)$ and $\varphi_{s,z}(x)$, for an entire isolated rod whose mechanical properties E and I_z are known. The idea can be further extrapolated: instead of using (2.69) once for the entire structure, discretize it using multiple Bernoulli beams, determine the adequate compatibility conditions at the interfaces of the elements, and solve the resulting system of unknowns, consisting of the displacements at the said interfaces. This corresponds to a FEM approach. Hence, starting from the Bernoulli beam theory, one needs to create a FEM element whose assumed variation of $u_{s,y}$ fulfills the following conditions:

- assumes a spatial variation of $u_{y,\mathrm{FEM}}$ inside the element that satisfies (2.69) if only nodal loads are imposed, that is, if $q_{s,y} = 0$; this is because within an FEM model the loads can only be applied at the locations where the structural solution is solved for
- satisfies the adequate compatibility conditions at the interface of all adjacent elements, and the boundary conditions at the ends of the rod

In satisfying these two conditions, one needs to solve the structural problem only at the interface between the rods, while inside the elements the problem is weakly satisfied. Having built a suitable FEM model, the shape functions will have also been obtained. With these, according to (2.59) and (2.60), the element stiffness and mass matrices can be obtained.

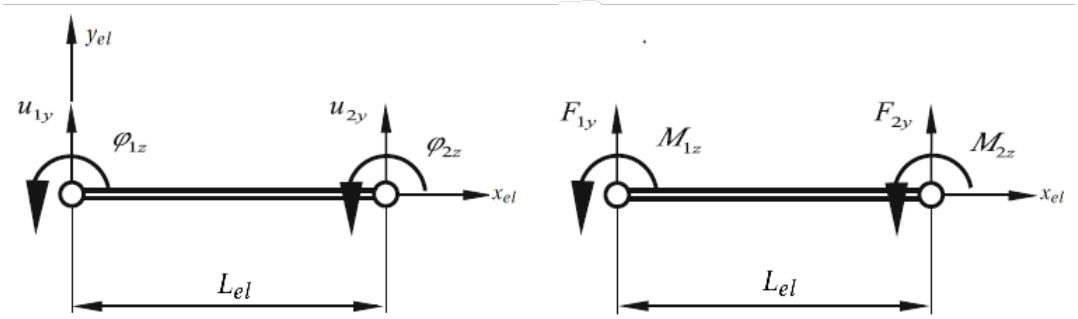


Figure 2.5: The Bernoulli beam element considered in this study for bending translation displacements in the $y-$ direction; image taken from [80]

Regarding the second requirement, if the considered element is not found at the ends of the rod, then one should satisfy the continuity of the bending translations and rotations $u_{s,y}$ and $\varphi_{s,z}$ at the interface between adjacent elements. What's more, for the considered applications, that is, nuclear fuel elements in axial flow, the boundary conditions consist of the clamped ends. Hence, locally, for the node at the ends of the structure, it should be imposed that $u_{s,y} = \varphi_{s,z} = 0$.

Under these requirements, one can use multiple types of beam elements. However, the one that is most commonly presented within the literature, is presented in Figure 2.5. It consists of two nodes, found at the ends of the element, and of two degrees of freedom for each node, a translation and a rotation, denoted as u_{1y}, φ_{1z}, and as u_{2y} and φ_{2z}, respectively. Within the element, the bending translation is assumed to vary as a third-order polynomial with respect to the local axial position, x_{el}:

$$u_{y,\text{FEM}}(x_{\text{el}}) = c_{1,\text{el}} + c_{2,\text{el}} \cdot x_{\text{el}} + c_{3,\text{el}} \cdot x_{\text{el}}^2 + c_{4,\text{el}} \cdot x_{\text{el}}^3. \tag{2.70}$$

In terms of the previously mentioned requirements for a Bernoulli beam element, it can be immediately observed that (2.70) can satisfy the homogeneous Bernoulli differential equation (2.69). For the second stipulated requirement, in imposing that $u_{y,\text{FEM}}(0) = u_{1y}$, $\frac{\mathrm{d}u_{y,\text{FEM}}}{\mathrm{d}x_{\text{el}}}(0) = \varphi_{1z}$, and $u_{y,\text{FEM}}(L_{\text{el}}) = u_{2y}$, $\frac{\mathrm{d}u_{y,\text{FEM}}}{\mathrm{d}x_{\text{el}}}(0) = \varphi_{2z}$, one can rewrite $u_{y,\text{FEM}}$ as [80]:

$$u_{y,\text{FEM}}(x_{\text{el}}) = \boldsymbol{N}^{\text{XY}}(x_{\text{el}}) \begin{bmatrix} u_{1y} \\ \varphi_{1z} \\ u_{2y} \\ \varphi_{2z} \end{bmatrix}, \tag{2.71}$$

where $\boldsymbol{N}^{\text{XY}}$ contains the shape functions of the considered Bernoulli beam element. More specifically, the spatially varying row vector $\boldsymbol{N}^{\text{XY}}$ is equal to:

$$\boldsymbol{N}^{\text{XY}}(x_{\text{el}}) = \begin{bmatrix} N_{1u}^{\text{XY}}(x_{\text{el}}) & N_{1\varphi}^{\text{XY}}(x_{\text{el}}) & N_{2u}^{\text{XY}}(x_{\text{el}}) & N_{2\varphi}^{\text{XY}}(x_{\text{el}}) \end{bmatrix}, \tag{2.72}$$

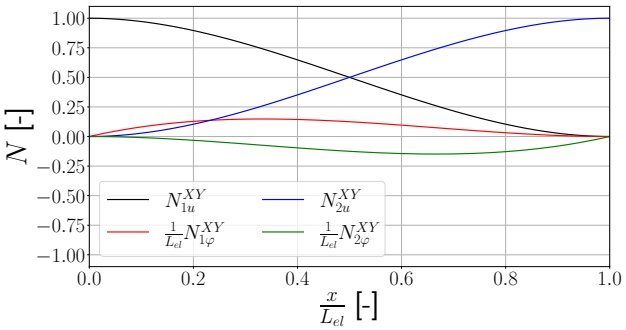
where [80]:

$$N_{1u}^{\mathrm{XY}}(x_{\mathrm{el}}) = 1 - 3\left(\frac{x_{\mathrm{el}}}{L_{\mathrm{el}}}\right)^2 + 2\left(\frac{x_{\mathrm{el}}}{L_{\mathrm{el}}}\right)^3, \tag{2.73}$$

$$N_{1\varphi}^{\mathrm{XY}}(x_{\mathrm{el}}) = x_{\mathrm{el}} - 2\frac{x_{\mathrm{el}}^2}{L_{\mathrm{el}}} + \frac{x_{\mathrm{el}}^3}{L_{\mathrm{el}}^2}, \tag{2.74}$$

$$N_{2u}^{\mathrm{XY}}(x_{\mathrm{el}}) = 3\left(\frac{x_{\mathrm{el}}}{L_{\mathrm{el}}}\right)^2 - 2\left(\frac{x_{\mathrm{el}}}{L_{\mathrm{el}}}\right)^3, \tag{2.75}$$

$$N_{2\varphi}^{\mathrm{XY}}(x_{\mathrm{el}}) = -\frac{x_{\mathrm{el}}^2}{L_{\mathrm{el}}} + \frac{x_{\mathrm{el}}^3}{L_{\mathrm{el}}^2}. \tag{2.76}$$



In analyzing equations (2.71) - (2.76), it can be observed that to each degree of freedom within the FEM problem a third-order polynomial with respect to the axial position is associated. On the other hand, the stiffness matrix can be computed using [80]:

$$\boldsymbol{K}_{\mathbf{el}}^{\mathbf{XY}} = EI_z \int_0^{L_{\mathrm{el}}} \begin{bmatrix} \frac{\mathrm{d}^2 N_{1u}^{\mathrm{XY}}(x_{\mathrm{el}})}{\mathrm{d}x_{\mathrm{el}}^2} \\ \frac{\mathrm{d}^2 N_{1\varphi}^{\mathrm{XY}}(x_{\mathrm{el}})}{\mathrm{d}x_{\mathrm{el}}^2} \\ \frac{\mathrm{d}^2 N_{2u}^{\mathrm{XY}}(x_{\mathrm{el}})}{\mathrm{d}x_{\mathrm{el}}^2} \\ \frac{\mathrm{d}^2 N_{2\varphi}^{\mathrm{XY}}(x_{\mathrm{el}})}{\mathrm{d}x_{\mathrm{el}}^2} \end{bmatrix} \begin{bmatrix} \frac{\mathrm{d}^2 N_{1u}^{\mathrm{XY}}(x_{\mathrm{el}})}{\mathrm{d}x_{\mathrm{el}}^2} & \frac{\mathrm{d}^2 N_{1\varphi}^{\mathrm{XY}}(x_{\mathrm{el}})}{\mathrm{d}x_{\mathrm{el}}^2} & \frac{\mathrm{d}^2 N_{2u}^{\mathrm{XY}}(x_{\mathrm{el}})}{\mathrm{d}x_{\mathrm{el}}^2} & \frac{\mathrm{d}^2 N_{2\varphi}^{\mathrm{XY}}(x_{\mathrm{el}})}{\mathrm{d}x_{\mathrm{el}}^2} \end{bmatrix} \mathrm{d}x_{\mathrm{el}}. \tag{2.77}$$

Using (2.77) and by doubly differentiating (2.73), one obtains [84]:

$$\boldsymbol{K}_{\mathbf{el}}^{\mathbf{XY}} = \frac{EI_z}{L_{\mathrm{el}}^3} \begin{bmatrix} 12 & 6\,L_{\mathrm{el}} & -12 & 6\,L_{\mathrm{el}} \\ 6\,L_{\mathrm{el}} & 4\,L_{\mathrm{el}}^2 & -6\,L_{\mathrm{el}} & 2\,L_{\mathrm{el}}^2 \\ -12 & -6\,L_{\mathrm{el}} & 12 & -6\,L_{\mathrm{el}} \\ 6\,L_{\mathrm{el}} & 2\,L_{\mathrm{el}}^2 & -6\,L_{\mathrm{el}} & 4\,L_{\mathrm{el}}^2 \end{bmatrix} \tag{2.78}$$

Similarly, for the mass matrix [80]:

$$\boldsymbol{M}_{\mathbf{el}}^{\mathbf{XY}} = \rho_{\mathrm{s}} \int_0^{L_{\mathrm{el}}} \begin{bmatrix} N_{1u}^{\mathrm{XY}}(x_{\mathrm{el}}) \\ N_{1\varphi}^{\mathrm{XY}}(x_{\mathrm{el}}) \\ N_{2u}^{\mathrm{XY}}(x_{\mathrm{el}}) \\ N_{2\varphi}^{\mathrm{XY}}(x_{\mathrm{el}}) \end{bmatrix} \begin{bmatrix} N_{1u}^{\mathrm{XY}}(x_{\mathrm{el}}) & N_{1\varphi}^{\mathrm{XY}}(x_{\mathrm{el}}) & N_{2u}^{\mathrm{XY}}(x_{\mathrm{el}}) & N_{2\varphi}^{\mathrm{XY}}(x_{\mathrm{el}}) \end{bmatrix} \mathrm{d}x_{\mathrm{el}}, \text{ or:} \tag{2.79}$$

$$\boldsymbol{M}_{\mathbf{el}}^{\mathbf{XY}} = \frac{\rho_{\mathrm{s}} \cdot \frac{\pi D^2}{4} \cdot L_{\mathrm{el}}}{420} \begin{bmatrix} 156 & 22\,L_{\mathrm{el}} & 54 & -13\,L_{\mathrm{el}} \\ 22\,L_{\mathrm{el}} & 4\,L_{\mathrm{el}}^2 & 13\,L_{\mathrm{el}} & -3\,L_{\mathrm{el}}^2 \\ 54 & 13\,L_{\mathrm{el}} & 156 & -22\,L_{\mathrm{el}} \\ -13\,L_{\mathrm{el}} & -3\,L_{\mathrm{el}}^2 & -22\,L_{\mathrm{el}} & 4\,L_{\mathrm{el}}^2 \end{bmatrix} \tag{2.80}$$

2.3.5. Two-plane bending

In Subsection 2.3.4, the considered Bernoulli model has been described for the situation where the translations due to bending occur in the $y-$ direction. In its current form, the model is complex enough to estimate the structural response for the nuclear fuel elements if the flow is assumed to be axisymmetric around the rod. However, for bundled configurations, this will not be the case. Analyzing the components on the right of Figure 1.1, the flow can be expected to be particularly tridimensional near the bounding walls, as well as near the spacer grid. This assumption is further supported by the fact that the supports of the spacer grids are not axisymmetric themselves [86]. Hence, it was considered a requirement that the beam model can estimate the deformations in multiple transverse planes of a given fuel element. In practice, this is done by also modelling the transverse translations in the $z-$ direction of the Cartesian reference system seen in Figure 2.5.

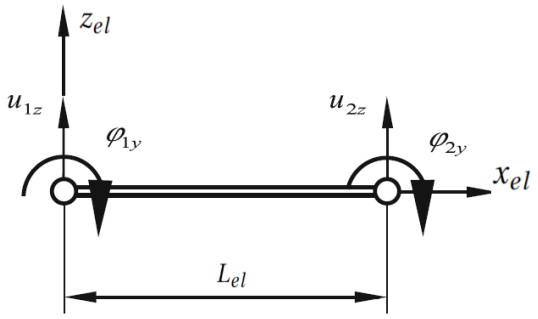


Figure 2.6: The Bernoulli beam element considered in this study for bending translation displacements in the $z-$ direction; image taken from [80]

In the remainder of this report, XY plane bending will refer to the component of the beam model presented in Subsection 2.3.4, while XZ plane bending to the one developed herein.

Analogous to Subsection 2.3.4, the beam element FEM model proposed for the XZ plane bending can be visualised in Figure 2.6. The main difference compared to Figure 2.5 is the clockwise orientation of the positive bending rotations for the two nodes. This leads to slightly different kinematics for the XZ bending plane, that is, $\varphi_{s,y} = -\frac{\mathrm{d}u_{\mathrm{s},z}(x)}{\mathrm{d}x}$. Though not visible in Figure 2.6, analogous to the XY bending plane case, the loads' positive orientation coincides with that of the nodal degrees of freedom, that is, F_{1y} and F_{2y} are positive if pointing upwards, while M_{1y} and M_{2y} if rotating clockwise. This change in orientation convention leads to slight sign differences in the kinematics and equilibrium relations, though the relations fundamentally remain the same [80]:

$$\epsilon_{\mathrm{s},x} = -y\frac{\mathrm{d}^2 u_{\mathrm{s},z}(x)}{\mathrm{d}x^2}, \tag{2.81}$$

$$\varphi_{s,y} = -\frac{\mathrm{d}u_{\mathrm{s},z}(x)}{\mathrm{d}x}, \tag{2.82}$$

$$\frac{\mathrm{d}^2 M_{\mathrm{s},y}(x)}{\mathrm{d}x^2} = \frac{\mathrm{d}Q_{\mathrm{s},z}(x)}{\mathrm{d}x} = -q_{\mathrm{s},z}. \tag{2.83}$$

For the XZ bending plane, the assumed variation of the bending translations remains a third-order polynomial, as in equation (2.70). Due to the different sign convention, the following relations between the XY and the XZ bending shape functions are found:

$$N^{\mathrm{XY}}_{1u}(x_{\mathrm{el}}) = N^{\mathrm{XZ}}_{1u}(x_{\mathrm{el}}), \tag{2.84}$$

$$N^{\mathrm{XY}}_{1\varphi}(x_{\mathrm{el}}) = -N^{\mathrm{XZ}}_{1\varphi}(x_{\mathrm{el}}), \tag{2.85}$$

$$N^{\mathrm{XY}}_{2u}(x_{\mathrm{el}}) = N^{\mathrm{XZ}}_{2u}(x_{\mathrm{el}}), \tag{2.86}$$

$$N^{\mathrm{XY}}_{1\varphi}(x_{\mathrm{el}}) = -N^{\mathrm{XZ}}_{1\varphi}(x_{\mathrm{el}}). \tag{2.87}$$

Compared to the XY bending plane, the modification of the shape functions (2.84) - (2.87) leads to slightly different stiffness and mass matrices, where individual entries may differ in sign [80]:

$$\boldsymbol{K}^{\mathbf{XZ}}_{\mathbf{el}} = \frac{EI_y}{L^3_{\mathrm{el}}}\begin{bmatrix} 12 & -6\,L_{\mathrm{el}} & -12 & -6\,L_{\mathrm{el}} \\ -6\,L_{\mathrm{el}} & 4\,L^2_{\mathrm{el}} & 6\,L_{\mathrm{el}} & 2L^2_{\mathrm{el}} \\ -12 & 6\,L_{\mathrm{el}} & 12 & 6\,L_{\mathrm{el}} \\ -6\,L_{\mathrm{el}} & 2L^2_{\mathrm{el}} & 6L_{\mathrm{el}} & 4L^2_{\mathrm{el}} \end{bmatrix}, \text{ and} \tag{2.88}$$

$$\boldsymbol{M}^{\mathbf{XZ}}_{\mathbf{el}} = \frac{\rho_{\mathrm{s}} \cdot \frac{\pi D^2}{4} \cdot L_{\mathrm{el}}}{420}\begin{bmatrix} 156 & -22\,L_{\mathrm{el}} & 54 & 13\,L_{\mathrm{el}} \\ -22\,L_{\mathrm{el}} & 4\,L^2_{\mathrm{el}} & -13\,L_{\mathrm{el}} & -3\,L^2_{\mathrm{el}} \\ 54 & -13\,L_{\mathrm{el}} & 156 & 22\,L_{\mathrm{el}} \\ 13\,L_{\mathrm{el}} & -3\,L^2_{\mathrm{el}} & 22\,L_{\mathrm{el}} & 4\,L^2_{\mathrm{el}} \end{bmatrix}, \tag{2.89}$$

where I_y represents the bending moment of inertia around the $y-$ axis. Having obtained the stiffness and mass properties for the two bending planes, they are combined within an 8×8 matrix for the entire beam element that will be multiplied with the displacement vector $\boldsymbol{u}^T_{\mathbf{el}} = \begin{bmatrix} u_{1y} & u_{1z} & \varphi_{1y} & \varphi_{1z} & u_{2y} & u_{\mathrm{s},2z} & \varphi_{\mathrm{s},2y} & \varphi_{2z} \end{bmatrix}$, as this is more convenient for the considered method of assembling the global system [87]. To build the combined matrices starting from (2.78), (2.80), (2.88) and (2.89), it is assumed that there is no cross-coupling between the two bending planes, that is, the translations and rotations associated with the bending in the

XY plane do not influence the displacements in the XZ one, and vice versa. This assumption is valid for shear rigid beams, and linear displacements [88]. Thus, to obtain the matrices for the two-plane bending structural model, one simply needs to use the entries in the matrices for each bending plane, and rearrange them correctly according to the degrees of freedom it corresponds to. In doing so, one obtains, for example, the following stiffness matrix for the element:

$$\begin{array}{c} \\ u_{1y} \\ u_{1z} \\ \varphi_{1y} \\ \varphi_{1z} \\ u_{2y} \\ u_{2z} \\ \varphi_{2y} \\ \varphi_{2z} \end{array} \begin{array}{c} \begin{array}{cccccccc} u_{1y} & u_{1z} & \varphi_{1y} & \varphi_{1z} & u_{2y} & u_{2z} & \varphi_{2y} & \varphi_{2z} \end{array} \\ \begin{bmatrix} 12I_z & 0 & 0 & 6I_z\,L_{\text{el}} & -12I_z & 0 & 0 & 6I_z\,L_{\text{el}} \\ 0 & 12I_y & -6\,I_yL_{\text{el}} & 0 & 0 & -12I_y & -6\,I_yL_{\text{el}} & 0 \\ 0 & -6\,I_yL_{\text{el}} & 4\,I_yL_{\text{el}}^2 & 0 & 0 & 6\,I_yL_{\text{el}} & 2\,I_yL_{\text{el}}^2 & 0 \\ 6\,I_zL_{\text{el}} & 0 & 0 & 4\,I_zL_{\text{el}}^2 & -6\,I_zL_{\text{el}} & 0 & 0 & 2\,I_zL_{\text{el}}^2 \\ -12I_z & 0 & 0 & -6\,I_zL_{\text{el}} & 12I_z & 0 & 0 & -6\,I_zL_{\text{el}} \\ 0 & -12I_y & 6\,I_yL_{\text{el}} & 0 & 0 & 12I_y & 6\,I_yL_{\text{el}} & 0 \\ 0 & -6\,I_yL_{\text{el}} & 2\,I_yL_{\text{el}}^2 & 0 & 0 & 6\,I_yL_{\text{el}} & 4\,I_yL_{\text{el}}^2 & 0 \\ 6\,I_zL_{\text{el}} & 0 & 0 & 2\,I_zL_{\text{el}}^2 & -6\,I_zL_{\text{el}} & 0 & 0 & 4\,I_zL_{\text{el}}^2 \end{bmatrix} \end{array} \frac{E}{L_{\text{el}}^3} = \boldsymbol{K_{\text{el}}} \tag{2.90}$$

In this Subsection, only the stiffness matrix is presented, but one can apply the same logic for forming the mass matrix. This is presented in Section A.1, Appendix A.

2.3.6. Equivalent nodal loads

Using the information presented in Subsection 2.3.5, one can now obtain the mass and stiffness matrices for a given beam element, where two bending planes are considered. Using an adequate assembly technique, the left-hand side of (2.58) can be obtained. However, the topic of how to obtain the nodal forces, that is, the right-hand side of (2.58), has not yet been discussed. As can be seen in Figure 2.4, the Bernoulli beam model takes as input the distributed loads $q_{\text{s},y}$ and $q_{\text{s},z}$ over the element. However, the FEM model only accepts concentrated loads at the nodes of the grid, that is, at the ends of each beam element. A link between the two needs to be formulated in order to use the FEM approach for solving the structural problem. This was implicitly done when going from the general formulation of the principle of virtual displacements (2.57) to the final form of the FEM system to be solved, (2.58). There, it was implied that the nodal forces $\boldsymbol{f}$ are equal to [81]:

$$\boldsymbol{f_{\text{s}}} = \sum_{\text{i}} \int_{\partial V_{\text{s,i}}} \boldsymbol{N}_{\text{i}}^T \boldsymbol{f}_{\text{s, i}}\, dS. \tag{2.91}$$

Hence, one obtains the nodal loads by weighing the local values of the shape functions with the distributed surface loads across the boundary, and by performing an integration of this quantity over all elements. Similar to the global mass matrix $\boldsymbol{M_{\text{s}}}$, both lumped and consistent approaches can be used to obtain $\boldsymbol{f_{\text{s}}}$. As mentioned in Subsection 2.3.1, in this work, a consistent approach is used for the $\boldsymbol{M_{\text{s}}}$ to best represent the dynamic characteristics of the structure. The same should also be done for $\boldsymbol{f_{\text{s}}}$, as the accuracy of using a consistent mass matrix may be lost if one uses a lumped formulation of the nodal forces [81]. In particular for this study, where beam FEM elements are used, (2.91) reduces to:

$$\boldsymbol{f_{\text{s}}^{\text{XY}}} = \int_0^{L_{\text{el}}} \boldsymbol{N}^{\text{XY}}(x_{\text{el}}) q_{\text{s},y}\ \text{d}x_{\text{el}}, \tag{2.92}$$

$$\boldsymbol{f_{\text{s}}^{\text{XZ}}} = \int_0^{L_{\text{el}}} \boldsymbol{N}^{\text{XZ}}(x_{\text{el}}) q_{\text{s},z}\ \text{d}x_{\text{el}}. \tag{2.93}$$

2.3.7. Numerical time integration

The FEM system of equations (2.58) represents a dynamic system. The solution of the structural problem is obtained using specific techniques. These are traditionally separated into two categories [81]: direct integration, where the solution at different time instances is obtained successively by working directly onto the structural problem as defined in (2.58), or Mode Superposition, where first the problem is transformed into modal coordinates and a different formulation is used to obtain the (unique) structural solution. In the following, only the former is explained. In this work, the Mode Superposition technique is only used for comparison purposes and is therefore only briefly discussed in Section A.2, Appendix A. For more information on the technique, Bathe's book [81] is recommended.

The direct integration technique consists of two steps [81]. First, the equation (2.58) is only fulfilled at discrete time instances t_i, where for simplicity of notation it is assumed that the timestep Δt is constant throughout

the transient. Secondly, by making assumptions on how the structural acceleration, velocity, and displacements vary between the calculated time instances, and by using previously obtained solutions, the time integration algorithm can be used to propagate the structural solution in time. Based on the hypotheses made for the variation of these structural response properties, various methods can be obtained.

In particular for this research, the standard Newmark algorithm is chosen [81]. This represents an implicit method, that is, the dynamic solution is obtained starting from the formulation of the FEM problem (2.58) at the timestep for which the structural solution is desired to be solved, t_{i+1}:

$$\boldsymbol{M_s}\,{}^{t_{i+1}}\ddot{\boldsymbol{u}}_\mathbf{s} + \boldsymbol{C_s}\,{}^{t_{i+1}}\dot{\boldsymbol{u}}_\mathbf{s} + \boldsymbol{K_s}\,{}^{t_{i+1}}\boldsymbol{u}_\mathbf{s} = {}^{t_{i+1}}\boldsymbol{f}_\mathbf{s}. \tag{2.94}$$

Next to this, the Newmark method assumes the following dependencies for the structural velocities and accelerations:

$$\begin{cases} {}^{t_{i+1}}\dot{\boldsymbol{u}}_\mathbf{s} = {}^{t_i}\dot{\boldsymbol{u}}_\mathbf{s} + \left[(1-\gamma)\cdot {}^{t_i}\ddot{\boldsymbol{u}}_\mathbf{s} + \gamma\cdot {}^{t_{i+1}}\ddot{\boldsymbol{u}}_\mathbf{s}\right]\Delta t \\ {}^{t_{i+1}}\boldsymbol{u}_\mathbf{s} = {}^{t}\boldsymbol{u}_\mathbf{s} + {}^{t}\dot{\boldsymbol{u}}_\mathbf{s}\Delta t + \left[\left(\frac{1}{2}-\alpha\right)\cdot {}^{t}\ddot{\boldsymbol{u}}_\mathbf{s} + \alpha\cdot {}^{t_{i+1}}\ddot{\boldsymbol{u}}_\mathbf{s}\right]\Delta t^2 \end{cases}, \tag{2.95}$$

where γ and α represent constant scalars. Thus, the Newmark algorithm assumes that the structural velocities and displacements can be obtained using a linear combination of their respective derivatives for the current timestep, $i+1$, and the previous one, i. By using the assumptions in equation (2.95), the terms in the dynamic system (2.94) can be rearranged so that the accelerations ${}^{t_{i+1}}\ddot{\boldsymbol{u}}_\mathbf{s}$ are determined. Subsequently, by knowing the solution of the displacement, velocity and acceleration at the previous timestep, the current velocity ${}^{t_{i+1}}\dot{\boldsymbol{u}}_\mathbf{s}$ and displacement ${}^{t_{i+1}}\boldsymbol{u}_\mathbf{s}$ can be obtained.

Having presented the definition and solution process of the Newmark method, its main advantage can be explained: versatility. Based on the combination of values of γ and of α, numerical time integration schemes with different advantageous properties can be obtained, such as unconditional stability $\left(\text{if } 2\alpha \geq \gamma \geq \frac{1}{2}\right)$, or numerical damping $\left(\text{if } \gamma > \frac{1}{2}\right)$ [89]. For the Newmark parameters used in this, it was chosen that $\alpha = \frac{1}{4}$, $\gamma = \frac{1}{2}$. This corresponds to a trapezoidal scheme, for which the acceleration is assumed to vary linearly between two different timesteps. The motivation for this choice is twofold. For one, this particular Newmark scheme does not introduce any numerical damping [89], which is a desirable property given the potential interest in the added damping effect of the fluid for TIV. What's more, the trapezoidal scheme was also used within NRG-FSIFOAM by van den Bos [38] and by Freitas [41]. Hence, the choice of the time integration scheme also allows for a more reliable comparison between NRG-beamFoam and the workflow it is supposed to replace.

2.4. ROM

The motivation of this research is to determine computationally efficient structural models that can be used to study the FSI occurring in axial water flow within PWR nuclear reactors. To this end, the theory behind a simplified 1D beam element FEM solver was already presented in Section 2.3. However, Reduced Order Models (ROMs) are analyzed as a potential method to further reduce the costs associated with the structural computation, as it previously showed potential in similar studies dealing with the topic of simulation axial flow TIV in nuclear reactors [36, 62, 90].

This Section begins by describing the general concept behind the ROM, and by offering a categorization of the different techniques that were analyzed for this research. The two are presented in Subsections 2.4.1 and 2.4.2. Subsequently, a dedication discussion is carried for each separate family of ROM methods that was identified in literature. This is done in Subsections 2.4.3 - 2.4.5. Finally, the conclusions of the analysis of the different ROM methods, including the methods that will be further pursued in this thesis, are presented in Subsection 2.4.6.

2.4.1. General idea of ROM

As shown in Subsection 2.3.1, using an FEM approach the structural problem reduces to:

$$\boldsymbol{M_s}\ddot{\boldsymbol{u}}_\mathbf{s} + \boldsymbol{K_s}\boldsymbol{u}_\mathbf{s} = \boldsymbol{f}_\mathbf{s}(t), \tag{2.58}$$

where $\boldsymbol{M_s}$ and $\boldsymbol{K_s}$ represent the mass and stiffness matrices resulting from the discretization. The size of the system when all degrees of freedom are considered, denoted in this Section as N_{FEM}, is generally of order

$\mathcal{O}(10^4)$ for a single beam when using the 3D Lagrange-element FEM approach of deal.II in NRG-FSIFOAM [38], and is expected to be of order $\mathcal{O}(10^2 - 10^3)$ for NRG-beamFoam. The computational costs increase exponentially with the size of the grid [81], which means that it will become increasingly challenging to simulate complex structures. The general objective of ROM methods is to represent the initial set of displacements $\boldsymbol{u_s}$, of size $\mathbb{R}^{N_{\text{FEM}}}$, in terms of a new set of coordinates, $\boldsymbol{z}$, of size $\mathbb{R}^{N_{\text{ROM}}}$, which is significantly lower than the initial FEM basis [91]. This inevitably leads to representation errors of the initial FEM displacements, which will be denoted as $\boldsymbol{\chi_{\text{ROM}}}$. The most fundamental objectives of a ROM method are to minimize the errors in the representation of the original field, while also doing so in a computationally efficient manner [92]. Though the error that is desired to be minimized is dependent on $\boldsymbol{\chi_{\text{ROM}}}$, the two are not necessarily coincident. This is particularly clear in frequency weighted methods, such as the frequency-weighted Balanced Truncation, as presented in Subsection 2.4.4.1.

Mathematically, the process of representing the initial displacements using the reduced basis is called projection, and the ROM process described in the above paragraph can be summarized as:

$$\boldsymbol{u_s} = \boldsymbol{V_{\text{ROM}}}\boldsymbol{z} + \boldsymbol{\chi_{\text{ROM}}}, \tag{2.96}$$

where $\boldsymbol{V_{\text{ROM}}} \in \mathbb{R}^{N_{\text{FEM}} \times N_{\text{ROM}}}$ is called the projection matrix. Starting from these simple observations, a multitude of ROM methods have been developed independently, in three main fields: numerical mathematics, controls, and structural analysis [93]. As will be seen in the following sections, though some methods were born in different engineering areas, they can still be considered for mechanical applications.

2.4.2. Classification of ROM methods

Although one can classify the families of methods in multiple ways, based on their origin, properties, or computational cost (see, for example, [93]), in this Section a classification similar to that given by Antoulas, who studied general applications of ROM, is used [92]. In his work, the author subdivided the available methods into two families: Singular Value Decomposition (SVD) and Krylov. Apart from those, in this work modal methods are introduced as an additional third category, that has been particularly useful in mechanical engineering. For the SVD methods, only two of the four initial families were considered: Balanced Truncation (BT) and Proper Orthogonal Decomposition (POD). This is because for Empirical Gramians and Hankel Approximators no applications were found for second-order mechanical systems, and hence their feasibility for this study is difficult to assess. Furthermore, for the Krylov methods, only a brief study was undertaken, as they did not seem to bring significant advantages over their counterparts, as will be explained in Subsection 2.4.5. Particular interest was offered to studying modal methods families: mode displacement (MD), Mode Acceleration (MA), Modal Truncation Augmentation (MTA), and Load Dependent Ritz Vectors (LDRV). Though harmonic modal methods have also been published [94–97], they are not detailed due to their incapacity to effectively simulate broadband excitation, as that generated by the turbulent fluid flow for this research. Under these observations, the main families of ROM methods that were considered in this study can be seen in Figure 2.7.

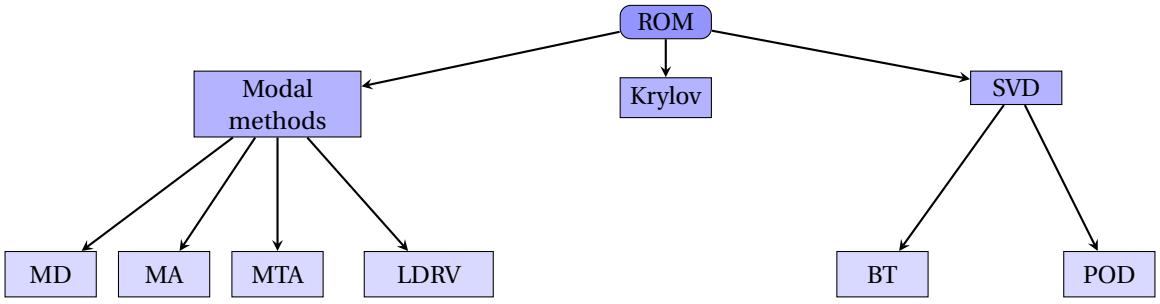


Figure 2.7: Classification of the ROM families considered for this research

2.4.3. Modal methods

The modal methods have historically been the main reduction model used in structural dynamics, and are focused on accurately representing the characteristic intrinsic behaviour of structures, rather than the link between the loads and the displacements, as methods in Sections 2.4.4 and 2.4.5 are.

The idea behind the initial FEM discretization, (2.58), is that one can divide a domain into a set of control volumes, inside of which a simple variation law of the unknowns can be accurately assumed, such as linear or quadratic. Based on this, one can deduce that the FEM method is inherently spatial, and is based on being able to assume simple variations (linear, quadratic) of the unknowns within a small subdomain of the structure. As a consequence, the FEM approach is also able to accurately describe the localized fields of stresses and displacements, which is important in static analyses. However, it does not take into account the intrinsic properties of the *entire* structure, that characterize its dynamic behaviour. For example, it has been observed experimentally (for example, by Païdoussis [83] for a TIV case) that the energy transfer from the fluid to the structure occurs at frequencies in a narrow band around the characteristic eigenfrequencies of the system, that can only be obtained through an eigenanalysis if one considers the structure as a whole. Even more important, the spatial response of the rods in axial flow for nuclear applications is dependent on the eigenmodes, where especially the first mode is visible [38]. Apart from the intuitive justification of using the eigenmodes and the eigenfrequencies as a basis to describe the structural problem, they also present an additional useful property as a candidate for a ROM projection basis: orthogonality. The eigenmodes and eigenfrequencies are obtained from solving the free, undamped vibration problem associated with a structure [98]:

$$\boldsymbol{M_s}\ddot{\boldsymbol{u}}_\mathbf{s} + \boldsymbol{K_s}\boldsymbol{u_s} = \mathbf{0}, \tag{2.97}$$

which based on a harmonic formulation, $\boldsymbol{u_s} = \boldsymbol{\phi_s} e^{i\omega_s t}$, can be written as:

$$\left(\boldsymbol{K_s} - \omega_s^2 \boldsymbol{M_s}\right)\boldsymbol{\phi_s} = \mathbf{0}, \tag{2.98}$$

where $\boldsymbol{\phi_s}$ and ω_s represent the structural eigenmodes and (circular) eigenfrequencies. By solving (2.98) successively, one obtains both the eigenfrequencies and the associated spatial eigenmodes of the mechanical system. Hence, the eigenvalue problem can be solved multiple times to obtain multiple eigenpairs. For a problem with N_{FEM} unknown displacements, one can find up to N_{FEM} eigenpairs that satisfy (2.98). In practice, N_{ROM} eigenmodes are used, where the number of selected eigenmodes is much lower than the size of the initial displacement field, $N_{\text{ROM}} \ll N_{\text{FEM}}$. The solution to the eigenvalue problem is not uniquely determined, as for each eigenfrequency the eigenmodes are unique up to a scaling factor. To make the eigensolution unique, as will be seen in (2.101), it is convenient to scale each eigenmode such that they are orthogonal with respect to the original mass and stiffness matrices:

$$\boldsymbol{\phi}_{\mathbf{s,j}}^T \boldsymbol{M_s} \boldsymbol{\phi}_{\mathbf{s,k}} = \delta_{jk}, \quad \boldsymbol{\phi}_{\mathbf{s,j}}^T \boldsymbol{K_s} \boldsymbol{\phi}_{\mathbf{s,k}} = \omega_{\text{s,j}}^2 \delta_{jk}, \ \forall\, j,k \leq N_{\text{FEM}} \tag{2.99}$$

where δ_{jk} represents the Kronecker delta operator. With the newly formed eigenbasis, $\boldsymbol{\phi}_{\mathbf{s,i}}$, with $i = 1,\ldots,N_{\text{ROM}}$, one can approximate the displacement field as:

$$\boldsymbol{u_s} = \sum_{\text{i}=1}^{N_{\text{ROM}}} a_{\text{s,i}}(t)\boldsymbol{\phi}_{\mathbf{s,i}} + \boldsymbol{\chi_{\text{ROM}}} = \boldsymbol{\Phi_s}\boldsymbol{a_s}(t) + \boldsymbol{\chi}_{\text{ROM}}, \tag{2.100}$$

where $\boldsymbol{a_s}(t)$ are called the generalized modal coordinates of the eigendecomposition. Comparing the general ROM projection law, (2.96), with (2.100), one can deduce that for modal methods the projection matrix $\boldsymbol{V_{\text{ROM}}}$ is the eigenbasis matrix $\boldsymbol{\Phi_s}$. In principle, if one solved the eigenvalue problem (2.98) N_{ROM} times, the exact solution could be retrieved, such that in (2.100) $\boldsymbol{\chi_{\text{ROM}}}$ becomes $\mathbf{0}$. In such a case, the technique would be called Mode Superposition. This will be used in Chapter 4 for verification purposes. However, this would lead to no reduction in the number of coordinates used to describe the problem. It is also important to note that for the assumed decomposition in (2.100), the eigenmodes are space-dependent only, while each generalized coordinate is a scalar that only corresponds to time. This means that in using a modal analysis one assumes that for all time instances the displacements can be obtained as a summation of the same spatial eigenmodes, by simply scaling each of them in time. This is in contrast to damped modes, whose spatial distribution of the eigenmodes may vary in time. In this regard, the mode superposition corresponds to the general separation of variables technique to solve partial differential equations. For simplicity of notation, in the following, it is assumed that the error vector $\boldsymbol{\chi_{\text{ROM}}}$ is negligible, however keeping in mind that the modal

transformation is an approximation. As the eigenmodes in (2.100) are known, the problem of determining the displacements reduces to that of finding the generalized coordinates $a_{s,i}$. These can be obtained by replacing the eigendecomposition (2.100) in the original FEM system, (2.97), and projecting back onto the reduced dual space of the original formulation by multiplying with the transpose of the original projection matrix:

$$\boldsymbol{\Phi}_\mathbf{s}^T \boldsymbol{M}_\mathbf{s} \boldsymbol{\Phi}_\mathbf{s} \ddot{\boldsymbol{a}}_\mathbf{s} + \boldsymbol{\Phi}_\mathbf{s}^T \boldsymbol{K}_\mathbf{s} \boldsymbol{\Phi}_\mathbf{s} \boldsymbol{a}_\mathbf{s} = \boldsymbol{\Phi}_\mathbf{s}^T \boldsymbol{f}_\mathbf{s}(t). \tag{2.101}$$

Based on (2.99), one can write (2.101) as:

$$\ddot{\boldsymbol{a}}_\mathbf{s} + \boldsymbol{\Omega}_\mathbf{s}^2 \boldsymbol{a}_\mathbf{s} = \boldsymbol{\Phi}_\mathbf{s}^T \boldsymbol{f}_\mathbf{s}(t), \text{ where} \tag{2.102}$$

$$\boldsymbol{\Omega}_\mathbf{s}^2 = \begin{bmatrix} \omega_{s,1}^2 & & \\ & \ddots & \\ & & \omega_{s,n}^2 \end{bmatrix}$$

Equation (2.102) represents an uncoupled system of Ordinary Differential Equations (ODEs). Indeed, this represents a consequence of the orthogonality of the eigenmodes. Starting from the original problem (2.58), of size N_{FEM}, by using the modal method approximation (2.100) a system of ODEs is obtained, where the size is given by N_{ROM}. Hence, the system is reduced, and one can apply traditional ODE solving techniques to obtain the values of the generalized coordinates. Once these are known, the displacements can be retrieved by simply replacing $\boldsymbol{a}_\mathbf{s}$ in the original transformation (2.100). It is also worth noting that in using a modal decomposition ROM, the system of equations (2.102) preserves the order of the original FEM system (2.58). Thanks to this, the same numerical time integration scheme can be used for both the original FEM system (2.58) and the reduced system (2.102). Thus, in this work, the Newmark integration scheme presented in Subsection 2.3.7 is used for both the FEM and the ROM solver.

2.4.3.1. Assumptions and limitations

Based on the very definition of the original problem, (2.58), written for the undeflected shape of the structure, it is supposed that the displacements are small, and that they vary linearly with respect the internal stresses. For non-linear deformations, a new, thin-walled structure formulation is written instead of (2.58) (see, for example, [98]). Nevertheless, the discussion of modal methods (and of all ROM methods) in this thesis is only made for linear cases.

By using traditional modal methods, it is assumed that the eigenmodes and the eigenfrequencies are provided as input. In practice, the eigenmodes are obtained either experimentally or numerically. The latter is much more common, especially during the design phase of a product. The numerical computation of the eigenmodes consists of solving (2.98). Since the system is of size N_{FEM}, for complex geometries the computations can become expensive, even if efficient iterative methods such as Lanczos or subspace iteration are used to gradually increase the considered eigenbasis [99]. In principle, this can hinder the computation of eigenmodes for entire bundles, as the computational cost could be too high. This drawback is addressed by Load-Dependent Ritz Vectors (LDRV), as will be presented in Subsection 2.4.3.5.

When the modal methods were introduced in Subsection 2.4.3, their working was explained based on the intrinsic behaviour of structures subjected to arbitrary loading. However, this is completely independent of the applied loads for a given problem. In the generalized coordinates decoupled system of ODEs (2.102), which represents the reduced problem that needs to be solved, the excitation on the right-hand side for a given eigenmode is obtained by computing the dot product between the initial load distribution and that particular mode shape. The result of this dot product represents the reduced excitation that actually activates a given eigenmode in the reduced system. The relative impact of the external forces within the ROM is hence given not only by their absolute value, but also by their alignment with particular modes. For example, a pure torsion eigenmode will not be excited by a strictly axial loading, which means that its associated generalized coordinate will be equal to 0 at all times. This example implies that the inclusion of the pure torsion eigenmode in the reduction basis of a purely axial problem would be useless. This suggests that the basic modal approach can be inefficient and slowly converging [100]. Even if one used a criterion to only consider the highly activated eigenmodes within the final reduced eigenbasis, the computation of the neglected eigenmodes can be expensive. Again, LDRV methods improve this limitation. Furthermore, it could be that the force distribution on the initial domain $\boldsymbol{f}_\mathbf{s}(t)$ contains spectral components above the highest

eigenfrequency $\omega_{s,p}$ considered in the modal approximation (2.100). In such a situation, the high-frequency components are not included in the reduced problem. Of course, a force misrepresentation leads to an error in the displacements of the problem within the reduced system. In fact, the solution of the generalized coordinates within (2.102) is exact, and the inaccuracies of the modal approximation can only originate in the misrepresentation of the reduced excitations, that is, of the RHS forcing term in (2.102) [101], or in the generation of a non-orthogonal reduced basis [102]. In practice, it is recommended that if the highest frequency component of the excitation is known, one should take an eigenbasis that has a bandwidth that is 1.5-2 times larger [99, 103]. Based on these limiting observations, the Mode Acceleration (MA) and Modal Truncation Augmentation (MTA) methods have been developed, as will be presented in Subsections 2.4.3.3 and 2.4.3.4, respectively, to account for higher-frequency contributions.

Furthermore, when the modal approximation was presented in (2.100), it was pointed out that if the number of the modes being considered is smaller than the initial size of the problem, $N_{\text{ROM}} \ll N_{\text{FEM}}$, an error $\boldsymbol{\chi}_{\mathbf{ROM}}$ is added to the displacement representation. Since one does not always know the excitation associated with the higher eigenmodes $\boldsymbol{\phi}_{\mathbf{s},j}^T \boldsymbol{f}_{\mathbf{s}}(t)$, for $N_{\text{ROM}} < j < N_{\text{FEM}}$, it is not possible to determine *a priori* the error in representing the exact solution due to the truncated set of higher-mode excitations. In literature, it is said that the modal methods do not have an error bound. This is in contrast with other families of methods, such as Balanced Truncation. This is relevant in practice because the use of modal methods requires user input to assess the quality of the reduced model, thereby increasing the difficulty of using a highly efficient, automated workflow.

Finally, the full potential of modal methods cannot be used for TIV applications. Although not explicitly stated in Subsection 2.4.3, modal methods are best used for static or periodic loads [104]. In this situation, the system (2.102) only needs to be solved once for all of timesteps. Due to the chaotic and broadband properties of the turbulent excitation [41], this is not applicable for this FSI application. Instead, the benefit of using the ROM is that at each timestep a system of reduced size N_{ROM} needs to be solved, (2.102), instead of one of size N_{FEM}, (2.58).

2.4.3.2. Mode Displacement (MD)

The mode displacement (MD) method implies the use of the methodology that led to (2.102), without any additional changes. In doing so, it is hence assumed that the N_{ROM} selected eigenmodes cover all the relevant excitation frequencies. The main advantage of MD is that it is the easiest method to implement among all other modal methods that will be presented in the remainder of Subsection 2.4.3. Thanks to the fact that the method models the internal behaviour of the structure, rather than focusing on an accurate depiction of a particular input-output behaviour of the structure, MD was found to be more robust than control-based methods such as Balanced Truncation or Krylov approaches by Besselink et al. [93]. In particular, it was found that the performance of the ROM is less dependent on the choice of the input (nodal loads) and output (nodal displacements) for the ROM. However, the relatively simplistic method leads to inheriting all of the disadvantages presented in Subsection 2.4.3.1: the prerequisite of knowing the solution to the eigenanalysis, the possibility of including unimportant eigenmodes, and the risks of misrepresenting the forces. In particular, the third one is found to be the major disadvantage of MD over the other types of modal methods. In literature, studies mostly compare the performance of MD with that of MA, presented in Subsection 2.4.3.3, and it is found that for the same number of considered eigenmodes the latter method is more accurate [97, 99, 105, 106]. It is important to point out that the neglect of the high-frequency excitation components doesn't lead only to a misrepresentation in the total energy and in the amplitude of forcing due to the time variation of the high-frequency loads, but also to the corresponding modification of the spatial loading [99, 107]. As a consequence, both acceleration (due to the lack of representation in time) and displacements (due to the neglect of their spatial distribution at a given time instance) are erroneous. Because of the high spatial variation of the high-frequency excitation components, their effect on the displacement can be seen as quasi-static. As a consequence, their local effects are modulated, and the mean amplitude of each excitation becomes most important.

2.4.3.3. Mode Acceleration (MA)

As was explained in Subsection 2.4.3.2, the neglect of high-frequency components at a given timestep leads to the effect of not taking into account the forcing of the highly fluctuating spatial components at a given time instance. Though explanations for the concept behind MA are offered in other works (see for example [108]), the explanation by Maddox [98] is provided in the following.

The solution to the uncoupled system (2.102) should satisfy the original problem (2.58) as accurately as possible. Based on this requirement, the MA displacements $\boldsymbol{u}_{\mathbf{s,MA}}$ should therefore ideally verify exactly that:

$$\boldsymbol{K}_{\mathbf{s}}\boldsymbol{u}_{\mathbf{s,MA}}(t) = \boldsymbol{f}_{\mathbf{s}}(t) - \boldsymbol{M}_{\mathbf{s}}\ddot{\boldsymbol{u}}_{\mathbf{MA}}(t). \tag{2.103}$$

Next, the MA assumes that the modal approximation is able to completely represent the dynamics of the structure within the reduced system, that is, the effect of the corresponding accelerations. Under this assumption, the acceleration term can be written as $\ddot{\boldsymbol{u}}_{\mathbf{s,MA}}(t) = \sum_{\mathrm{i}=1}^{N_{\mathrm{ROM}}} \ddot{a}_{\mathrm{s,i}}(t)\boldsymbol{\phi}_{\mathbf{s,i}}$ (notice the =, instead of ≈ or the addition of an error vector $\boldsymbol{\chi}_{\mathrm{ROM}}$), such that (2.103) becomes:

$$\boldsymbol{u}_{\mathbf{s,MA}}(t) = \boldsymbol{K}_{\mathbf{s}}^{-1}\boldsymbol{f}_{\mathbf{s}}(t) - \sum_{\mathrm{i}=1}^{N_{\mathrm{ROM}}} \frac{\boldsymbol{\phi}_{\mathbf{s,i}}\ddot{a}_{\mathrm{s,i}}(t)}{\omega_{\mathrm{s,i}}^2}. \tag{2.104}$$

Rearranging (2.102), and substituting the general acceleration term, $\ddot{a}_{\mathrm{s,i}} = \boldsymbol{\phi}_{\mathbf{s,i}}^T\boldsymbol{f}_{\mathbf{s}}(t) - \omega_{\mathrm{s,i}}^2 a_{\mathrm{s,i}}$, one obtains:

$$\boldsymbol{u}_{\mathbf{s,MA}}(t) = \sum_{\mathrm{i}=1}^{N_{\mathrm{ROM}}} \boldsymbol{\phi}_{\mathbf{s,i}} a_{\mathrm{s,i}}(t) + \underbrace{\left(\boldsymbol{K}_{\mathbf{s}}^{-1} - \sum_{\mathrm{i}=1}^{N_{\mathrm{ROM}}} \frac{\boldsymbol{\phi}_{\mathbf{s,i}}\boldsymbol{\phi}_{\mathbf{s,i}}^T}{\omega_{\mathrm{s,i}}^2}\right)\boldsymbol{f}_{\mathbf{s}}(t)}_{\text{quasi-static correction}}. \tag{2.105}$$

Starting from the mode displacement approach, the additional assumption that the acceleration term is fully represented by the initial modal approximation was added. The assumption is analogous to assuming that the higher frequency modes that are truncated in the reduced system only contribute statically to the solution, through the mean of the amplitude of their excitation, while the instantaneous variations are not significant. This is why in literature the MA is categorized as a quasi-static correction. Comparing (2.96) to (2.105), it can indeed be seen that the MA consists of adding a correction vector to the initial MD formulation. As the MA only corrects the static displacements, it can also be interpreted as a way of accurately representing the effect of the stiffness of the original problem in the reduced system [109]. The changes in the implementation, when compared to MD, are minimal. The quasi-static correction in (2.105) is fully computed using the already known eigenmodes, without increasing the size of the eigenbasis. The additional cost is then to compute the right term of (2.105), a matrix-to-vector product of size N_{FEM}, that can be readily computed if the eigenmodes are known.

Over time, the original MA method was used as a basis for other approaches. The most important is that of higher-order MA corrections. With the approach initially proposed by Rixen [99, 110] an arbitrary accuracy can be theoretically obtained, while still using the same eigenbasis. On the other hand, an improvement to the method of Rixen was presented by Paola and Failla [111], where fewer eigenmodes were used to apply the quasi-static corrections, but this was shown to perform worse than the original approach for low-frequency excitation, as is relevant for this study. Nevertheless, none of these higher-order corrections can be applied, as they are formulated by assuming a harmonic load over the structure, as opposed to a turbulent broadband excitation.

Next, the characteristics of the MA method are discussed. Due to the correction term (2.105), the method can reproduce the static displacements of the structure when solving the reduced system. At the same time, its dependency on $\frac{1}{\omega_{\mathrm{s,i}}^2}$ means that convergence is achieved faster than with MD when increasing the number of considered eigenmodes within the reduction basis [105]. The method also does not add significant computational cost when compared to MD, but is generally found to deliver significant improvements, especially in the low-frequency range of the excitations, which is of particular interest for TIV. On the other hand, the improvement of the MA in the low-frequency range excitation is found to also lead to a reduction in performance at higher ones [109, 111]. At the same time, the omission of the effect of the truncated modes over the velocity and the acceleration of the structural degrees of freedom can lead to slower convergence when compared to the Modal Truncation Augmentation approach described in Subsection 2.4.3.4.

2.4.3.4. Modal Truncation Augmentation (MTA)

The Modal Truncation Augmentation (MTA) method represents a natural evolution to the Mode Acceleration methods, where the effects of the high-frequency truncated modes within the excitation spectral decomposition are assumed to be not only static, but also dynamic. This has stronger consequences over the original workflow of MD than the MA family did, as will be explained below.

Similar to the Mode Acceleration method, the MTA starts from the observation that the forces are misrepresented within the reduced space due to the truncation of high-frequency modes. For the MTA approach, it is again assumed that the corrections applied need to be essentially quasi-static, since the excitation frequency range is assumed to be contained within the eigenfrequency band of the reduced space [103]. However, instead of correcting the preexisting eigenmodes obtained from MD, the idea is now to add new vectors to the reduction basis to improve the effect of the truncated modes over the spatial distributions of the displacements [103].

The MTA first considers the difference between the initial spatial loads that were introduced into the problem, $\boldsymbol{f}_{\mathbf{s}}(t)$ in (2.58), and the spatial loads that are contained within the reduced system, $\boldsymbol{f}_{\mathbf{s,ROM}}(t)$. Assuming that this residual is applied statically, the residual displacement vector $\boldsymbol{u}_{\mathbf{s,r}}$ can be computed using:

$$\boldsymbol{K}_{\mathbf{s}}\boldsymbol{u}_{\mathbf{s,r}}(t) = \boldsymbol{f}_{\mathbf{s}}(t) - \boldsymbol{f}_{\mathbf{s,ROM}}(t) = \boldsymbol{f}_{\mathbf{s}}(t) - \boldsymbol{M}_{\mathbf{s}}\boldsymbol{\Phi}_{\mathbf{s}}\boldsymbol{\Phi}_{\mathbf{s}}^{T}\boldsymbol{f}_{\mathbf{s}}(t). \tag{2.106}$$

The vector in (2.106), which can be interpreted as a Ritz vector [98], is subsequently used to reduce the original system, and to obtain reduced mass and stiffness matrices, $\widetilde{\boldsymbol{M}}_{\mathbf{s}} = \boldsymbol{u}_{\mathbf{s,r}}^{T}\boldsymbol{M}_{\mathbf{s}}\boldsymbol{u}_{\mathbf{s,r}}$, $\widetilde{\boldsymbol{K}}_{\mathbf{s}} = \boldsymbol{u}_{\mathbf{s,r}}^{T}\boldsymbol{K}_{\mathbf{s}}\boldsymbol{u}_{\mathbf{s,r}}$. The Ritz vector is then transformed into the first eigenmode of the reduced dynamic problem:

$$\left(\widetilde{\boldsymbol{K}}_{\mathbf{s}} - \tilde{\omega}_{\mathrm{s}}^{2}\widetilde{\boldsymbol{M}}_{\mathbf{s}}\right)\widetilde{\boldsymbol{\phi}}_{\mathbf{s}} = \mathbf{0}. \tag{2.107}$$

The steps (2.106) and (2.107) can be undertaken before any of the static displacements are obtained, that is, before either the original problem (2.58) or the reduced system (2.102) is solved. Hence, one can simply extend the eigenbasis $\boldsymbol{\Phi}_{\mathbf{s}}$ used for the MD method with the newly acquired Ritz orthogonal vector, $\widetilde{\boldsymbol{\phi}}_{\mathbf{s}}$ from (2.107), to obtain $\boldsymbol{\Phi}_{\mathbf{s},\mathrm{MTA}}$:

$$\boldsymbol{\Phi}_{\mathbf{s},\mathrm{MTA}} = \begin{bmatrix}\boldsymbol{\Phi}_{\mathbf{s}} & \widetilde{\boldsymbol{\phi}}_{\mathbf{s}}\end{bmatrix}. \tag{2.108}$$

Due to (2.108), it can be inferred that when using the MTA method the number of ODE's solved in step (2.102) increases with 1 when compared to the MD or the MA approach. This implies that an additional ODE is solved for the modal coordinate associated with the MTA correction. When compared to the MA method, this implies that the appended vector contributes dynamically to the solution [103]. This is expected to increase the performance of the ROM when subjected to high-frequency excitations that are outside of the eigenfrequency range of the method. Though higher-order MTA methods have been proposed [99, 110], they are not detailed here due to their application only to harmonic excitations.

On the other hand, the feasibility of the method for the considered application is unclear. Historically, it has only been used by first decomposing the nodal forcing $\boldsymbol{f}_{\mathbf{s}}(t)$ as a superposition of *multiple* d time-varying load shapes, where $d > 1$:

$$\boldsymbol{f}_{\mathbf{s}}(t) \approx \sum_{\mathrm{i}=1}^{d} g_{\mathrm{i}}(t)\mathbf{f}_{\mathbf{s},\mathrm{i}}(\boldsymbol{x}) = \mathbf{F}_{\mathbf{0}}(\boldsymbol{x})\mathbf{g}(t). \tag{2.109}$$

However, the methodology (2.106)-(2.108) was presented by assuming that $d = 1$. Due to the broadband and chaotic behavior of turbulence, the decomposition (2.109) is not easily obtained for this study. At the same time, it is not certain what the performance of the MTA when using a single load vector would be. What's more, since the external forcing $\boldsymbol{f}_{\mathbf{s}}(t)$ changes at every timestep, the computations in equations (2.106)-(2.107) would have to be undertaken at every timestep. Consequently, the eigenbasis would also change, leading to a different form of the reduced system (2.102) at each time instance. By contrast, for the MD and MA methods, the reduced system remains the same, except for the excitation computed on the right-hand side of the equation. In practice, this means that using the MD and the MA reduces the number of operations that has to be executed at each timestep.

2.4.3.5. Load Dependent Ritz Vectors (LDRV)

The Load Dependent Ritz Vectors (LDRV) methods represent the main alternative to the eigenmode-based methods presented up to this point in Subsection 2.4.3. The improvement of LDRV over the standard MD is that for the former the loads applied onto the structure are used to form a reduction basis. By contrast, the previous modal methods started from the eigenmodes and used them to estimate the loads [99]. It can be used both by starting from the original excitation $\boldsymbol{f}_{\mathbf{s}}(t)$ from (2.58), or by using a load decomposition as in (2.109). In the following, only the former option is presented, as this is the only one relevant for the topic of this thesis. The original method of Wilson [112] is described, with subsequent variations being mentioned.

At the beginning of Subsection 2.4.3, it was argued that since the eigenmodes and eigenfrequencies characterize the response of the structure to external loads, they are good building blocks for approximating the dynamic response. However, in the same Subsection it was later pointed out that the activation of each mode is given by the reduced excitation, obtained by the inner product between the corresponding eigenmode and the force distribution, and hence eigenmodes that are orthogonal to the loads have no contribution to the solution. This can make the potentially expensive step of computing the eigenanalysis (2.98) unjustified. As pointed out by Wilson [112], the only stringent requirements for the considered reduced basis are that the considered modes are orthogonal, due to the projection on the dual space that must be done to obtain the uncoupled system (2.101), and that the eigenbasis is $\boldsymbol{K}_{\mathbf{s}}-$ and $\boldsymbol{M}_{\mathbf{s}}-$ orthonormal. Although the eigensolution does fulfill these criteria, it is not the only suitable candidate. To ensure that the modes within the reduced basis would actually contribute to the solution, one can generate a first eigenmode guess, $\boldsymbol{x}^*_{\mathbf{s},1}$, starting from the inputted spatial distributions of the loads:

$$\boldsymbol{K}_{\mathbf{s}}\boldsymbol{x}^*_{\mathbf{s},1} = \boldsymbol{f}_{\mathbf{s}}(t), \tag{2.110}$$

where the * notation is used for a guess value. Once a guess is offered, the vector is $\boldsymbol{M}_{\mathbf{s}}-$ normalized, such that $\boldsymbol{x}^T_{\mathbf{s},1}\boldsymbol{M}_{\mathbf{s}}\boldsymbol{x}_{\mathbf{s},1} = 1$. At this point, the reduced basis only takes into account the static effects of the loads. To account for the dynamic effects, one could equate the displacements of a second mode to an inertial force depending on $\boldsymbol{x}_{\mathbf{s},1}$. This force would be equal to $-\omega^2_{\mathrm{s},1}\boldsymbol{M}_{\mathbf{s}}\boldsymbol{x}_{\mathbf{s},1}$. However, since for this method one does not know the eigenfrequencies *a priori*, the guess is made as:

$$\boldsymbol{K}_{\mathbf{s}}\boldsymbol{x}^*_{\mathbf{s},2} = \boldsymbol{M}_{\mathbf{s}}\boldsymbol{x}_{\mathbf{s},1}. \tag{2.111}$$

After the step in (2.111), $\boldsymbol{x}^*_{\mathbf{s},2}$ is $\boldsymbol{M}_{\mathbf{s}}$ normalized as $\boldsymbol{x}_{\mathbf{s},2} = \boldsymbol{x}^*_{\mathbf{s},2} - \boldsymbol{x}_{\mathrm{s},1}\boldsymbol{M}_{\mathbf{s}}\boldsymbol{x}^*_{\mathrm{s},1}$, and added to the basis. However, one now has to correct also for the dynamic effects associated with $\boldsymbol{x}_{\mathbf{s},2}$. This leads to the iterative algorithm of the LDRV, where the current step dynamically corrects the previous one, and is subsequently $\boldsymbol{M}_{\mathbf{s}}-$ normalized. This loop is executed for a predefined number of times, such that in total N_{ROM} Ritz vectors are obtained.

At this point of the problem, a predefined number of Ritz vectors N_{ROM} is obtained, and the basis they span is $\boldsymbol{M}_{\mathbf{s}}-$ orthonormal. It is assumed that the current Ritz vectors are stored column-wise in the matrix $\boldsymbol{X}_{\mathbf{s,Ritz}} \in \mathbb{R}^{N_{\mathrm{FEM}} \times N_{\mathrm{ROM}}}$. To also make them $\boldsymbol{K}_{\mathbf{s}}-$ orthonormal, based on (2.99), one needs to obtain the eigenfrequencies $\omega_{\mathrm{s,i}}$. This is done in the reduced space of the problem, of size $\mathbb{R}^{N_{\mathrm{ROM}}}$, rather than in the original one of size $\mathbb{R}^{N_{FEM}}$, by projecting the original matrices using $\boldsymbol{X}_{\mathbf{s,Ritz}}$, and obtaining $\boldsymbol{M}_{\mathbf{s,Ritz}} = \boldsymbol{X}^T_{\mathbf{s,Ritz}}\boldsymbol{M}_{\mathbf{s}}\boldsymbol{X}_{\mathbf{s,Ritz}}$, $\boldsymbol{K}_{\mathbf{s,Ritz}} = \boldsymbol{X}^T_{\mathbf{s,Ritz}}\boldsymbol{K}_{\mathbf{s}}\boldsymbol{X}_{\mathbf{s,Ritz}}$, and subsequently solving the reduced eigenproblem:

$$\left(\boldsymbol{K}_{\mathbf{s,Ritz}} - \omega_{\mathrm{s,Ritz,i}}\boldsymbol{M}_{\mathbf{s,Ritz}}\right)\boldsymbol{z}_{\mathbf{Ritz},\,\mathrm{i}} = \mathbf{0},\ i = 1,\ldots,N_{\mathrm{ROM}} \tag{2.112}$$

Once all eigenvectors in (2.112) are computed, the final Ritz vectors can be obtained as:

$$\boldsymbol{X}_{\mathbf{s,LDRV}} = \boldsymbol{X}_{\mathbf{s,Ritz}}\boldsymbol{Z}_{\mathbf{Ritz}}, \tag{2.113}$$

where $\boldsymbol{Z}_{\mathbf{Ritz}} = \left[\boldsymbol{z}_{\mathbf{Ritz},1}\ \boldsymbol{z}_{\mathbf{Ritz},2}\ldots\boldsymbol{z}_{\mathbf{Ritz},\mathrm{NROM}}\right]$. Due to the fact that a projected eigenvalue problem (2.112) is used instead of the original one (2.98), the eigenpairs represent approximations, and their accuracy is dependent on the size of the initial Ritz basis.

The first improvement of the Wilson algorithm came from Nour-Omid [113], who observed the similarities with the Lanczos algorithm for a Krylov subspace, thereby reducing the costs associated with the iterative orthogonalization process. Léger [114] also proposed a different orthogonalization process that improves stability by using a different iterative process. Gu et al. [115] proposed a Quasi-Static Ritz Vectors variant,

which focused on increasing the accuracy of the ROM force representation around multiple user-imposed excitation frequencies. For the considered application, a possible candidate could be the eigenfrequencies of the structure that are known to be activated the most by the turbulent flow.

The main advantage of the method is that it does not require the eigenanalysis as a prerequisite anymore. Furthermore, the construction of the reduction basis starting from the loads implies that all basis vectors contribute to the approximation within the ROM dual space. This ought to eliminate the inclusion of unused eigenmodes that MD, MA, and MTA methods may be subjected to. An implication of this is that for a given required accuracy the basis that one needs to use is smaller when compared to standard eigenmode-based approaches. Another advantage of the method is that one can define different error norms to *a priori* estimate the accuracy of the method. In the initial article of Wilson [112], he proposed a force representation error, equal to the difference between an initial load spatial distribution and that obtained by combining the Ritz vectors. Another often used criterion is that of the modal mass participation [116]. Others have been proposed, such as the included forcing function percentage, based on the ratio between the total sum of Power Spectral Density of the included frequencies and the total one, which requires the use of a Fourier Transform [117]. In principle, as the Ritz vectors are added iteratively to the basis, at each successive step these errors can be used to ascertain whether the basis is large enough, and this could be used to automate the workflow. The available variations of the original LDRV method also brought their specific advantages, such as improved computational cost and convergence.

In terms of disadvantages, the LDRV method presents similar limitations as the MTA method. The LDRV method has only been used in combination with a load decomposition (2.109). As was previously mentioned, due to the broadband energy content of turbulence, this approach is not feasible for the topic treated in this thesis. Furthermore, since the external forcing $\boldsymbol{f}_{\mathbf{s}}(t)$ is continuously changing throughout the transient computations, it is required that the basis of Ritz vectors is recomputed at each subiteration, and that the reduced system (2.102) is changed continuously. By contrast, the rest of the modal methods only require the computation of the eigenbasis once, at the beginning of the transient simulation.

2.4.4. SVD methods

The Singular Value Decomposition (SVD) methods try to accurately contain the relation between the input and the output of a given physical system. For a first-order system, this can be written as a single transfer matrix [92], which in the following will be called $\boldsymbol{H}$, of size $\boldsymbol{H} \in \mathbb{R}^{N_{\text{FEM}} \times N_{\text{out}}}$, where N_{FEM} is the original size of the system and N_{out} is the number of outputs. Since for most applications $N_{\text{FEM}} \neq N_{\text{out}}$, the system matrix $\boldsymbol{H}$ is rectangular, and the concept of its eigendecomposition does not hold. A similar idea is the Singular Value Decomposition, wherein the matrix $\boldsymbol{H}$ is factorized as a product between a set of orthonormal vectors, each of which has an associated scaling factor, called a singular value, which are stored in a diagonal matrix $\boldsymbol{\Sigma}_{\mathbf{SVD}}$. However, due to its rectangular layout, $\boldsymbol{H}$ is written as a product between a left- and a right- orthonormal base, denoted as $\boldsymbol{F}_{\mathbf{SVD}}$ and as $\boldsymbol{G}_{\mathbf{SVD}}$, respectively. Hence, the SVD factorization has the form:

$$\boldsymbol{H} = \boldsymbol{F}_{\mathbf{SVD}} \boldsymbol{\Sigma}_{\mathbf{SVD}} \boldsymbol{G}_{\mathbf{SVD}}^{T}, \tag{2.114}$$

where $\boldsymbol{F}_{\mathbf{SVD}} \in \mathbb{R}^{N_{\text{FEM}} \times N_{\text{FEM}}}$, $\boldsymbol{G}_{\mathbf{SVD}} \in \mathbb{R}^{N_{\text{out}} \times N_{\text{out}}}$, $\boldsymbol{\Sigma}_{\mathbf{SVD}} \in \mathbb{R}^{N_{\text{FEM}} \times N_{\text{out}}}$. Equation (2.114) represents the SVD of the modified full system (2.58). The SVD ROM methods, use only a few of the left- and right- singular vectors, N_{ROM}, that are contained along the columns of $\boldsymbol{F}_{\mathbf{SVD}}$ and $\boldsymbol{G}_{\mathbf{SVD}}$, denoted as $\boldsymbol{F}_{\mathbf{red}}$ and $\boldsymbol{G}_{\mathbf{red}}$, to represent the system in an approximated form as:

$$\boldsymbol{H} \approx \boldsymbol{F}_{\mathbf{red}} \boldsymbol{\Sigma}_{\mathbf{red}} \boldsymbol{G}_{\mathbf{red}}^{T}, \tag{2.115}$$

where $\boldsymbol{F}_{\mathbf{red}} \in \mathbb{R}^{N_{\text{FEM}} \times N_{\text{ROM}}}$, $\boldsymbol{\Sigma}_{\mathbf{red}} \in \mathbb{R}^{N_{\text{ROM}} \times N_{\text{ROM}}}$, $\boldsymbol{G}_{\mathbf{red}} \in \mathbb{R}^{N_{\text{out}} \times N_{\text{ROM}}}$. Starting from (2.115), the SVD ROM methods aim to make the best choice of the N_{ROM} reduced left- and right- singular vectors, as well as to accurately and efficiently compute their associated singular values, $\boldsymbol{\Sigma}_{\mathbf{red}}$.

2.4.4.1. Balanced Truncation

The trend of automating the functioning of most devices and processes within the last few decades has inevitably led to the development of numerous control applications. In practice, such systems can have a large number of degrees of freedom, which makes the real-time response prohibitive due to computational complexity. To achieve feasible computational costs, model reduction is often employed by starting from the linear state-space formulation of the system. State-space model reduction techniques could also be applied

to the current problem, as the original second-order dynamic FEM problem (2.58) can be cast into this form [118]. In particular, Balanced Truncation (BT) is a ROM method that is based on transforming the initial state variables into a convenient coordinate system such that it allows for only the effect of the N_{ROM} most important states to be used within the simulation. The method presents distinct advantages over any other ROM family considered herein, in particular, that of offering an error bound, which is why it is analyzed in the following. In this Subsection, only a brief summary of a part of the vast literature within the field of BT is provided. The focus is on techniques that are more closely related to mechanical engineering applications.

The BT method was formulated for a state space representation of the full system. For our problem, the inputs are the loads applied onto the structure, the state variables are the displacements, and the output represents a subset of the displacements that are of interest. As an example of a particular choice of displacements that might be of interest, if GTRF is studied, the deformations of the beam elements near the supports may be selected. In control theory, two important characteristics of the different states of a first-order system are their observability and controllability. The first refers to how well the change in the characterized state is reflected in the outputs of the system, while the latter to the extent to which an initial state can be brought to a desired one in a finite amount of time [93]. The controllability and observability of a given system can be expressed by using the Gramian matrices. Doing so is convenient because the eigenanalysis of the Gramians leads to both an ordering of the directions that are most controllable or observable within the system through the eigenvectors, as well as their respective quantitative impact over observability and controllability based on the corresponding eigenvalues. The Gramians are obtained by solving the coupled Lyapunov equations [93].

Starting from the definition and properties of the Gramians, the Balanced Truncation is built on the idea that the input-output behaviour is the most important property of the original system that needs to be preserved when it is reduced. In this regard, it is desired that states which are both controllable (as they require the least amount of force to be displaced, therefore best characterizing the input-state) and observable (least energy needed to produce the displacements of interest for the system) are those retained. Based on this observation, Moore [119] found that there exists a unique linear transformation of the original system, after which the states of the system are ordered by taking into account their observability and controllability with equal importance. Once this transformation is undertaken, one can simply retain the first N_{ROM} transformed states, associated with the first N_{ROM} Hankel values, as those have the greatest impact over the controllability and observability, and hence best conserve the input-output behavior of the system.

The most expensive part of the Balanced Truncation algorithm is the determination of the Gramians. This is done by solving the Lyapunov equations. For systems where the number of state variables is small or medium (order smaller than $\mathcal{O}(10^3)$), standard methods can be used. However, these are not suitable for larger systems [93], as the number of associated operations is of order $\mathcal{O}(N_{\mathrm{FEM}}^3)$, and even more limiting, the storage requirements are of order $\mathcal{O}(N_{\mathrm{FEM}}^2)$ [120]. Though it is not clear at this moment what the required discretization of the rods using beam elements is, since the number of rods within a bundle is in the order of hundreds, and to fully capture the physics multiple bundles may have to be measured [29], it is likely that in the future the workflow will have a structural discretization of order $\mathcal{O}(10^4)$. If traditional methods were used to obtain the Gramians, the Balanced Truncation would become the most expensive ROM method presented in this work [93]. Though more efficient computation methods to solve the Lyapunov equations have been put forward, by, for example, Gugercin and Li [120], they are not feasible for methods that have a large number of inputs and outputs, as is the case for this study. This makes the BT ROM method unfeasible for this project, despite its specific advantages, such as its stability and error bound.

2.4.4.2. Proper Orthogonal Decomposition (POD)

The POD method, introduced by Lumley [121] at the end of the 1960s, is based on using a reliable sample dataset acquired at different spatial locations and at different timesteps, and subsequently trying to determine a set of spatially-distributed shape functions called Proper Orthogonal Modes (POMs) that best approximate the observed distributions at all time instances. The starting point of the POD method is a set of sample results at different spatial locations and time instances that are connected to the unknowns of the problem that is desired to be reduced. Subsequently, the input is written as a linear superposition of the POMs. From the requirement that the POMs should satisfy the lowest Frobenius norm of the difference between them and the reference data, the POMs can be obtained [122]. The POD ROM has the advantage of capturing the highest level of energy for a given dimension of the reduction basis [123]. Nevertheless, for our applications, the

obtaining of the required input sampled data is not possible. In the case when an entire bundle is considered, there are no experimental facilities, nor enough available computational power to do a high-fidelity benchmark simulation within NRG. Furthermore, the step of generating high-fidelity data for the POD would likely be cumbersome if multiple different cases need to be run. As a consequence, the POD method is not feasible.

2.4.5. Krylov

Similar to the Balanced Truncation method, Krylov approaches are based on preserving the input-behaviour during the ROM simplification. However, instead of ordering the states based on their impact and maintaining only the most important ones, the Krylov method sets out to approximate the general transfer function of the system altogether, by using a Taylor polynomial expansion around a main frequency, and trying to preserve as many coefficients of the expansion as possible. In literature, these polynomial coefficients are also called moments. As a consequence of this, in some works the Krylov-based ROM methods are also known as *moment matching* methods. Similar to the BT ROM method, the starting point of the Krylov method is the state-space formulation of a system. It was proven that the projection matrix $\boldsymbol{V}_{\mathbf{ROM}}$ from equation (2.96) that can fulfill the moment matching property can be obtained by determining the Krylov subspace of matrix products depending on the state space matrices and their inverses [93]. For conciseness, the mathematical model of the Krylov is not presented here, but the interested reader is referred to the recent review article of Rafiq and Bazaz [124].

Multiple methods to determine the said Krylov subspace have been put forward, such as Padé via Lanczos [125], or Arnoldi [126]. The main advantage of the Krylov method, especially when compared to BT, is that the computational resources required for the reduction are lower. This is because all variations of the Krylov algorithms are based on matrix-vector multiplications of sparse matrices [92], leading to less storage requirements and to fewer operations undertaken within the algorithm. It was also found that the Krylov method can be more robust than the BT. Witteveen [127, 128] and Besselink et al. [93] concluded that regardless of the inputs and outputs chosen to represent the system within the ROM, when compared to BT, moment matching methods can more readily identify all excitation modes being activated.

Nevertheless, despite numerous potential benefits, there are also disadvantages to the Krylov approaches, which make them unsuitable for this study. Most notably, the method is formulated in the frequency domain, making it impractical for broadband TIV excitations. What's more, due to the fact that the loads are distributed along the entirety of the rods, multiple inputs have to be taken into account. This makes the particular studied application a Multiple Input Multiple Output (MIMO) system. For MIMO configurations, no method to preserve the stability of the system was proven [124]. The lack of guarantee regarding stability means that the Krylov methods are comparable to the modal methods in terms of the useful properties that they possess. However, the modal methods have been found to perform better in terms of estimating the eigenfrequencies of the system in the two studies of Witteveen that were previously mentioned in this subsection [127, 128], as well as in the overall estimation of the transfer function of Besselink when the inputs and outputs of the reduction are misaligned with those of interest [93].

One last disadvantage of the methods is the complexity of implementation. Because we are dealing with a MIMO system, either the block Arnoldi or the block Lanczos algorithm has to be implemented. Apart from the determination of the columns themselves, one needs to add additional steps to mitigate the risk of linear dependency between the different columns. For example, in the Matrix-Padé via Lanczos method one eliminates previous vectors that are observed to be linearly dependent [129]. At the same time, the number of vectors required for the reduced basis has never been researched for nuclear fuel column applications, and it could not be accurately estimated during the literature study phase of this thesis. Nevertheless, the block Arnoldi and the block Lanczos methods are known to become unstable if large bases are used [124]. In practice, due to their complexity, the implementation of such methods in the allocated time for the project would be ill-advised.

2.4.6. Conclusions

Throughout this Section, different potential ROM methods from different families has been presented. It was assumed that the full system that is desired to be reduced is represented for the structural side by 1D FEM beam elements, whose inputs are given by all the loads across the length of the beam, while the outputs are a selected subset of the local displacements. Under these assumptions, the main advantages and disadvantages of the considered ROM methods are presented in Table 2.1. It is found that **only modal methods may be feasible for reducing the dimensionality of the problem within this study**. Despite numerous advantages, the other methods present at least one major disadvantage for the considered application. Those are marked with red in Table 2.1. Conversely, the ROM methods written in green, and whose names have been boxed, are the ones that will be implemented for this thesis. Though BT offers stability and an error bound, for the current application the computation of the Gramians, a step necessary for any of its variants, is prohibitively expensive, since no "cheap" iterative method that has been published applies to a large MIMO system as the one analyzed here. Although POD may offer the possibility of an efficient reduction for a small size of the basis, its use here is not possible since it requires high-fidelity data to be provided for the case that is desired to be reduced. At the moment, nor the available computational power or the experimental facilities within NRG allow this dataset to be obtained. Finally, even though the Krylov methods are relatively inexpensive from a computational point of view, while also allowing for accurate reductions at given excitation frequencies, they are still fundamentally frequency-based methods. This implies that one would have to transform the fluid loads into the Fourier domain. The feasibility of doing so for the broadband turbulent forcing is dubious. Hence, the use of Krylov appears to over-complicate the desired reduction process.

On the other hand, the implementation of the eigenmode-based modal methods is relatively simple, because a dedicated eigensolver can be easily integrated within NRG-beamFoam, since the global mass and stiffness matrices $\boldsymbol{M}_\mathbf{s}$ and $\boldsymbol{K}_\mathbf{s}$ are already computed for the FEM solver presented in Section 2.3, and the eigenproblem (2.98) can be readily computed. Though the original Mode Displacement method presents a few limitations, **it is the modal approach that is easiest to implement, and it therefore represents a good starting point for the ROM solver developed for this thesis**. Some of the limitations of the MD method are partially solved by more sophisticated models, such as Mode Acceleration (MA), Mode Truncation Augmentation (MTA), and Load-Dependent Ritz Vector (LDRV). However, the last two bring decisive disadvantages. Given the random forcing pattern of turbulence, the reduced system (2.102) would have to change at every timestep, making the computation of the dynamic solution more complicated. By contrast, for the MD and the MA methods the left side of the reduced system remains constant, making the implementation simpler. What's more, both the MTA and the LDRV methods were proposed based on a decomposition into multiple spatially-varying components. However, given the random and broadband behaviour of the turbulent forcing, it is not clear how this decomposition would be conducted for this thesis. Hence, one would have to resort to a single load vector, that is changed with every new timestep. Hence, **the MA method is also found to be a suitable ROM candidate, thanks to its balance between accuracy, flexibility, and implementation effort**. Though the GMA represents an improved, higher-order correction of the MA method, it is not feasible for this study, since it can only be used for harmonic load analyses.

ROM method	Advantages	Disadvantages
MD	• Ease of implementation	• Inefficient and slow converging • Does not reproduce the static solution • Superfluous eigenmodes may be computed • Eigenanlysis required *a priori* • No error bound
MA	• Convergence improvement over MD • Low additional cost over MD • Good performance for low-frequency forcing	• Only applies linear correction to MD • Inertial effects of high-frequency excitations neglected • Superfluous eigenmodes may be computed • Eigenanlysis required *a priori* • No error bound
GMA	• Improved accuracy for low-frequency loads compared to MA • Nonlinear iterative corrections	• Requires load decomposition, (2.109) • Number of corrections limited by numerical errors • Superfluous eigenmodes may be computed • Eigenanlysis required *a priori* • No error bound
MTA	• Inertial effects of high-frequency excitations included • Improved behaviour for small reduction eigenbasis	• Requires recomputation of the reduced system (2.102) at each subiteration • Traditionally used with a load decomposition, (2.109) • Computationally expensive • Superfluous eigenmodes may be computed • Eigenanlysis required *a priori* • No error bound
LDRV	• NO eigenanlysis required *a priori* • Better convergence than conventional modal methods • Error estimators can be used *a priori*	• Requires recomputation of the reduced system (2.102) at each subiteration • Traditionally used with a load decomposition, (2.109) • Can become unstable for complex loads
Krylov	• Cheaper and more robust than BT • High accuracy obtained around user-inputted frequencies	• Requires Fourier decomposition of the loads • Difficult to implement
BT	• Well-defined error bound • Preserves stability	• Expensive and unstable for MIMO systems • Accuracy dependent on the chosen inputs and outputs • Difficult to implement
POD	• Low order of basis required for good accuracy	• Requires high-fidelity simulations or experiments as input

Table 2.1: Advantages and disadvantages of the considered ROM methods

3

Methodology

In Chapter 1, it was pointed out that given the potential two-way coupled phenomena that may occur for axial water flow over fuel elements in nuclear reactors, a two-way coupled workflow is required between the fluid and the structure. The main topic of this research is to propose an efficient structural solver that should replace the current 3D FEM approach used by NRG for the modelling of axial-flow TIV in nuclear reactors. In Chapter 2, potential FEM and ROM models were presented as candidates for the solution of the structural problem. However, determining the displacements of the structural nodes represents only part of the steps taken when considering the entire workflow developed for the FSI problem. A conceptual flowchart of the steps taken in the FSI problem is described in Section 3.1. Section 3.2 presents the structure of the PIMPLE algorithm that is used for the fluid computations in the AniPFM model. This information is relevant since the ultimate goal of NRG-beamFoam workflow is to work with the AniPFM model. The proposed data structure for integrating the FSI solver with the PIMPLE is presented in Section 3.3. Section 3.4 offers an extensive description of each of the components in the FSI workflow. As will be detailed in this Chapter, apart from the structural solver itself, other parts of the FSI workflow that had been previously dealt with by the coupling software preCICE [130] were newly implemented for this work. Each newly added component to the workflow had to be independently verified before continuing the development of the methodology. Subsequently, the entire FSI workflow was tested using laminar, URANS, and AniPFM models. The individual and combined test cases are presented in Section 3.5.

3.1. General guidelines

To numerically simulate the fluid and the structural behaviour, one often uses two different frameworks for solving the mathematical models associated with each. First, the fluid model is described. Within NRG, different finite-volume approaches are used [38]. The most complex fluid model that is used for axial-flow TIV is the URANS-based AniPFM. The ultimate goal of the work in this thesis is thus to be coupled with the AniPFM. In Subsection 6.2.1, it is shown that when coupled with NRG-beamFoam, the computation of the pressure and velocity fluctuations by AniPFM can be done once per timestep, at the first FSI subiteration that is executed for the current timestep, $N_{\text{loop}} = 1$. Apart from the AniPFM, NRG-beamFoam should also be capable of accurate coupling with simpler fluid models, such as URANS, for verification and validation purposes [38].

Next, the structural solver is discussed. The model proposed herein is essentially FEM-based: as was already mentioned in Chapter 2, even if one uses one of the ROM approaches to reduce the dimension of the problem, the results are still projected back to the previously generated FEM grid. In this context, specific challenges arise, as was pointed out in Section 2.1, namely those of transferring the information from one grid to the other, as well as that of minimizing the partitioning error between the solvers. For the former demand, mapping routines need to established between the fluid and the structural meshes, while for the latter, a subiteration algorithm must be implemented. Thus, apart from the structural solver itself, this thesis will also treat the mapping between the domains, as well as the subiteration algorithm between the two. Based on these observations, the conceptual workflow proposed in this thesis can be seen in Figure 3.1.

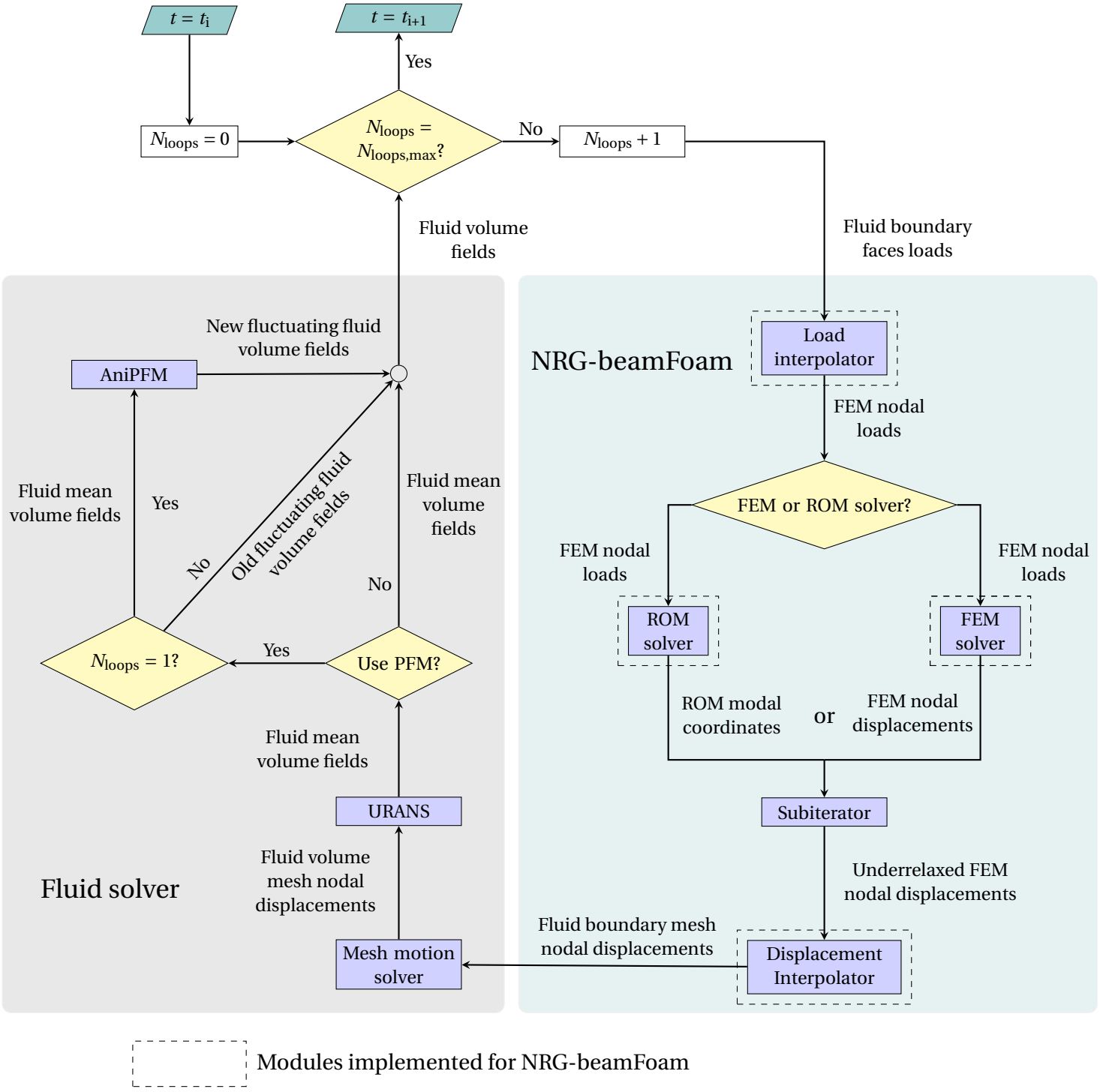


Figure 3.1: Methodology of the thesis

The outcome of this research is going to be used by NRG within its axial FSI workflow, which has been validated for TIV of single rods [38, 41]. In this context, the work herein would ideally be integrated seamlessly with the preexisting components of the workflow that will interface with the newly implemented NRG-beamFoam features. Namely, the components that would like to be reused are the fluid and the mesh motion solver, as can be seen in Figure 3.1. The state-of-the-art for the two within NRG are the AniPFM model for the fluid, and a Laplace-equation based motion solver for the mesh motion displacement [41]. Both of these are implemented in OpenFOAM [42], an open-source object-oriented C++ library that is used especially for CFD simulations. More specifically, the AniPFM and the mesh motion solver have been used in OpenFOAM 8.

Though any version of OpenFOAM can be coupled with an external library to obtain the inputs it needs for the mesh motion solver and the fluid solver (indeed, this is exactly what was done before the start of this thesis within NRG, by using deal.II and preCICE inside the NRG-FSIFOAM [38]), this approach leads to a computational overhead when compared to using modules of OpenFOAM, due to the additional memory and data transfer routines that an OpenFOAM adapter requires [131]. Furthermore, it is important to note that the AniPFM and the mesh motion solver have only been validated using the PIMPLE algorithm for the solution of the fluid fields. Hence, to avoid additional verification work, it is also preferable to encapsulate the steps in Figure 3.1 within the PIMPLE algorithm. In conclusion, it is then desirable, if possible, to also implement the new components of the FSI workflow directly within the PIMPLE algorithm in OpenFOAM 8. The feasibility of doing so will be presented in the following Section.

3.2. PIMPLE algorithm and mesh motion solver

The PIMPLE algorithm represents a combination between the Semi-Implicit-Method-Of-Pressure-Linked-Equations (SIMPLE) and the Pressure-Implicit-of-Split-Operations (PISO) algorithm. Both of these are pressure-based coupled solvers, which are generally used for incompressible flows. The pressure-based coupled solvers are based on the sequential solution of the velocity and then of the pressure equations [132]. Once these are solved, one computes the additional turbulent and temperature fields. For this work, only the turbulence will be computed, as the exchange of thermal energy is not yet part of the modelling objectives of this research.

To better explain the working of the PIMPLE algorithm, the technique of solving for the velocity field and the pressure is detailed. For both SIMPLE and PISO, the first step is to compute a first guess for the momentum predictor for the velocity field, by discretizing the momentum equation:

$$\boldsymbol{U_f}\cdot\nabla\boldsymbol{U_f}-\nabla\cdot(\nu_f\nabla\boldsymbol{U_f})=-\nabla p\rightarrow\mathscr{M}\boldsymbol{U_f}=-\nabla p, \tag{3.1}$$

where ν_f represents the dynamic viscosity of the fluid, and $\mathscr{M}$ is a known matrix dependent on the discretization scheme and the mesh configuration. By computing (3.1), the velocity field satisfies the conservation of momentum inside the fluid domain. However, the continuity constraint is not yet satisfied. The fulfillment of the mass conservation is therefore done by calculating the pressure from the continuity equation, equation (3.3), and then correcting the velocity field accordingly, equation (3.4). To do so efficiently, however, the left-hand side of the momentum equations $\mathscr{M}\boldsymbol{U_f}$ is first split into a diagonal, easily invertible part, $\mathscr{A}\boldsymbol{U_f}$, and $\mathscr{H}$, which can be interpreted as the matrix containing the effects of the velocity field's non-linearity. The decomposition of the matrix $\mathscr{M}$ is given in equation (3.2). The use of the decomposition (3.2) becomes apparent in (3.3), where one now needs to invert only $\mathscr{A}$ instead of the entire $\mathscr{M}$, thereby making this step cheaper. It is also worth noting that within literature the step in (3.1) is known as the momentum predictor, while that in (3.4) as the momentum corrector.

$$\mathscr{M}\boldsymbol{U_f}=-\nabla p\rightarrow\mathscr{A}\boldsymbol{U_f}-\mathscr{H}=-\nabla p\rightarrow\boldsymbol{U_f}=\mathscr{A}^{-1}\mathscr{H}-\mathscr{A}^{-1}\nabla p. \tag{3.2}$$

$$\nabla\cdot\boldsymbol{U_f}=0\xrightarrow{(3.2)}\nabla\cdot\left[\mathscr{A}^{-1}\mathscr{H}-\mathscr{A}^{-1}\nabla p\right]=0\rightarrow\nabla\cdot\left[\mathscr{A}^{-1}\nabla p\right]=\nabla\cdot\left[\mathscr{A}^{-1}\mathscr{H}\right] \tag{3.3}$$

$$\boldsymbol{U_f}=\mathscr{A}^{-1}\mathscr{H}-\mathscr{A}^{-1}\nabla p. \tag{3.4}$$

Though it might appear as if by executing the steps (3.1)-(3.4) once the velocity and the pressure fields satisfy both the continuity and the momentum equations, this is not so: the $\mathscr{H}$ matrix is dependent on the velocity solution $\boldsymbol{U_f}$, as can be seen in (3.2), and therefore one needs to update $\mathscr{H}$ by using the latest version of the velocity field $\boldsymbol{U_f}$, obtained using (3.4). Of course, the procedure of reupdating $\mathscr{H}$ based on the solution of the momentum corrector should be done iteratively until convergence. The SIMPLE algorithm does so by re-computing equations (3.1)-(3.4) at all iterations, into what is known as the outer corrector loop. On the other hand, the PISO algorithm only recomputes equations (3.2)-(3.4), computing the momentum predictor (3.1) only once per timestep, at the first iteration of the pressure-velocity coupled solution algorithm. The PISO iteration procedure is known as the inner loop. The SIMPLE algorithm has been historically used for steady state cases, while the PISO represents a newer proposal, developed in the context of transient simulations.

Starting from the general philosophies of the SIMPLE and of the PISO algorithm, the PIMPLE algorithm combines the two. First of all, for each timestep, the user can impose a given number of outer correctors applied at each timestep. However, for each outer corrector loop the user can also impose the number of inner (PISO) loops (3.2)-(3.4) to be executed. The merging of the SIMPLE and of the PISO algorithm in this manner leads

to specific advantages, such as for example that of allowing for the use of a large timestep, higher than that imposed by having a Courant number of one [74]. Furthermore, it is important to note that within the PIMPLE algorithm, one can impose the number of non-orthogonal correctors that are desired be executed. This consists of recomputing the equation (3.3) for a given number of times, before moving on to the momentum corrector (3.4) at a given inner loop. This is done so as to increase the accuracy of computing the diffusion term within the momentum equation [132].

Outside of the solution for the velocity-pressure coupling, the PIMPLE algorithm also includes the solution of other fields. For example, as mentioned in the previous Section, the turbulent fields are obtained after the pressure and the velocity have been computed. This can be done either within each outer corrector loop, or at the end of the lastly executed outer loop of each timestep. Furthermore, the PIMPLE algorithm is also compatible with built-in mesh deformation techniques within OpenFOAM. Similar to the turbulent fields, the mesh can be deformed either once per timestep, or at every outer corrector of the computation. However, it is important to note that the mesh is deformed *before* the pressure-velocity coupling is solved.

In this context, it is important to understand whether and how the workflow developed herein can be interfaced with the PIMPLE algorithm. Analyzing Figure 3.1, it can be seen that the communication between the preexisting fluid and mesh motion solver and the newly implemented routines of NRG-beamFoam is done through two sets of data. First, at the top right, one needs to input the loads on the cell faces of the fluid on the FSI interface. Second, on the bottom left, once the displacements on the FSI interface corresponding to the fluid loads are computed, they need to be transferred back to the fluid domain. More specifically, the structural displacements should serve as Boundary Conditions (BCs) to the mesh motion solver, which is called before the fluid fields are computed, as also mentioned above. The transfer of the fluid loads inside the workflow is achievable. Indeed, to compute the loads on the FSI surface, one needs to retrieve the local values of the pressure and of the viscous and turbulent forces. Within the OpenFOAM architecture, both are easily attainable, as they are attributes of the `fvMesh` class, containing the mesh topological information, as well as the corresponding values of the fluid fields. How one could impose the displacements as boundary conditions for the boundary is more involved, and is presented in the following paragraph.

```
223  void Foam::displacementLaplacianFvMotionSolver::solve()
224  {
225      // The points have moved so before interpolation update
226      // the motionSolver accordingly
227      movePoints(fvMesh_.points());
228
229      diffusivity().correct();
230      pointDisplacement_.boundaryFieldRef().updateCoeffs();
231
232      Foam::solve
233      (
234          fvm::laplacian
235          (
236              diffusivity().operator()(),
237              cellDisplacement_,
238              "laplacian(diffusivity,cellDisplacement)"
239          )
240      );
241  }
```

```
128  void Foam::waveDisplacementPointPatchVectorField::updateCoeffs()
129  {
130      if (this->updated())
131      {
132          return;
133      }
134
135      const polyMesh& mesh = this->internalField().mesh()();
136      const Time& t = mesh.time();
137
138      const scalarField points(waveNumber_ & patch().localPoints());
139
140      const scalar timeRamp = timeRamp_->value(t.value());
141
142      const scalarField startRamp(startRamp_->value(points));
143
144      const scalarField endRamp
145      (
146          endRamp_->value(points[points.size() - 1] - points)
147      );
148
149      Field<vector>::operator=
150      (
151          timeRamp*startRamp*endRamp*amplitude_*cos(omega_*t.value() - points)
152      );
153
154      fixedValuePointPatchField<vector>::updateCoeffs();
155  }
```

Figure 3.2: Code snippets for calling the execution of the PDE associated with the mesh motion solver (left), as well as for a typical implementation of the `updateCoeffs()` routine that updates the shape its associated BCs (right)

To understand how to impose the displacements of the FSI boundary for the mesh motion solver, one needs to look at the specific class implementation of the mesh motion solver used for the computation. Based on the considerations that will be presented in Subsection 3.4.5, for this study this corresponds to `Foam::displacementLaplacianFvMotionSolver`. The solution of the PDE associated with the mesh motion displacements is called using the `displacementLaplacianFvMotionSolver::solve()` method, as can be visualised on the left of Figure 3.2. There, before the solution of the Laplace equation is solved (lines

232-239), the reconfiguration of the mesh is called by using the `updateCoeffs()` method of the `pointDisplacement_` attribute (line 230). The `pointDisplacement` class has specific boundary conditions, imposed by the user in the `0/pointDisplacement` dictionary of OpenFOAM, such as `waveDisplacement` (which imposes a time-dependent sinusoidal shape of the boundary), or `surfaceDisplacement` (maps the local displacements of the fluid mesh onto a user-imposed surface). Each of these BC classes contains its own implementation of the `updateCoeffs()` method. As an example, the implementation for the `waveDisplacement` class is given on the right of Figure 3.2. It is important to note that for each implementation of `updateCoeffs()` within the different mesh motion BC class, a `Field<vector>` attribute is passed, equal to the displacement of each fluid node with respect to the initial mesh configuration.

3.3. Proposed OpenFOAM architecture

Based on the discussion in Subsection 3.2, a possible connection between the displacement interpolator and the mesh motion solver, as visible in Figure 3.1, is to create a new BC class for the mesh motion solver, that can compute the structural response for each of the fluid nodes on the FSI interface, and to pass those values to the mesh motion solver using its specific `updateCoeffs()` method. This ensures the required connection of the proposed workflow with the mesh motion solver, while the one between the fluid solver and the load interpolator is facilitated by the properties of the `fvMesh` class, as mentioned in the previous Subsection.

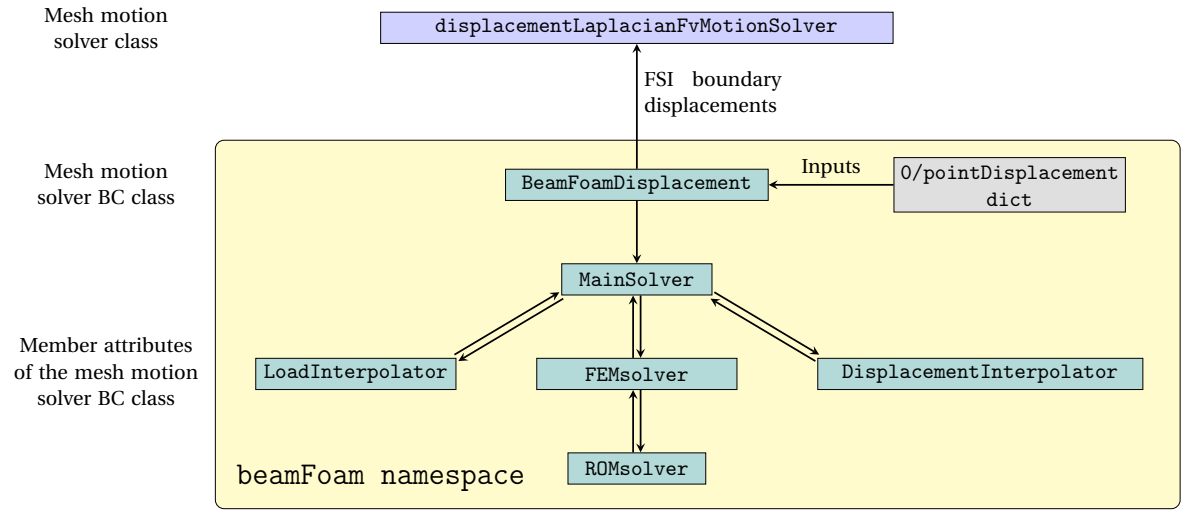


Figure 3.3: The proposed OpenFOAM architecture

A class that satisfies these conditions was created for this thesis, and it was called `BeamFoamDisplacement`. To provide the nodal displacements, one has to successively execute the different blocks presented in Figure 3.1, that is, the load interpolator, the structural solver, the underrelaxation of the displacements, and the displacement interpolator. For each of these, a standalone class was created, with the exception of the subiterator, whose implementation is straightforward, see equation (2.9), and will be integrated inside the FEM solver module. To facilitate the transfer of information between the classes (that is, the connecting arrows in Figure 3.1), and to eliminate the need to store redundant information for the problem in multiple classes, a `MainSolver` object was created, such that each of the three main components of the FSI workflow is constructed from `MainSolver`, and any can both receive and send information from and to it. Given that the FEM and the ROM solvers only differ in the size of the dynamic system that needs to be solved, with the same discretization and time integration scheme ultimately being used for both, the ROM solver class is built as a subclass of the FEM solver. The inputs to the problem are offered through the `0/pointDisplacement` dictionary which is used to construct the class. An example of the inputs for the problem is given in Section B.1, Appendix B. A visual representation of the proposed OpenFOAM architecture can be seen in Figure 3.3, for a patch called `inner`. For all of the subclasses implemented in this work, and especially for the `FEMsolver`, use was made of the Eigen C++ open-source library [133], thanks to its extensive linear algebra library that could be readily used.

Another important aspect of the problem was that of the parallelization requirement of the workflow. The AniPFM fluid model is designed to be run on multiple cores, where the fluid domain is decomposed into multiple volumes that are independently computed by each solver, and the boundary information is subsequently communicated between the different processors where needed. In this context, it was important to ensure that NRG-beamFoam is capable of working on multiple cores. Furthermore, due to the decomposition method of the fluid domain, the inputs to the NRG-beamFoam workflow, that is, the fluid boundary faces loads, are stored inside separate memory partitions that are allocated for each core. Thus, a first communication routine between the different processors is necessary to gather all of the necessary inputs for the problem. After this, the parallel integration of NRG-beamFoam could be done in two ways. One solution was to compute all of the steps of the NRG-beamFoam methodology in Figure 3.1 on a single core. Subsequently, once the displacements are obtained for all of the fluid nodes on the FSI boundary, they are then scattered back to the different processors whose associated subdomains contain nodes on the FSI boundary. However, this approach has the disadvantage of requiring an additional communication call between the processors, on top of the computation of the NRG-beamFoam loop itself. Instead, it was preferred to communicate the global inputs of the NRG-beamFoam workflow to all processors within the parallel run, and to subsequently compute the NRG-beamFoam loop on each core. Thus, each processor already has in its memory the nodal solution of the structural problem, and the location of their associated fluid nodes on the FSI boundary. As will be explained in Subsection 3.4.4, this is enough information to compute the outputs of the NRG-beamFoam workflow.

3.4. Individual FSI workflow components modelling

Having presented what the conceptual steps in the FSI workflow are in Subsection 3.1, as well as their corresponding OpenFOAM data structures in Subsection 3.3, it is time to present the characteristics for each component in the FSI workflow in Figure 3.1: the load interpolator (Subsection 3.4.1), the structural solvers (Subsection 3.4.2), the subiterator (Subsection 3.4.3), and the displacement interpolator (Subsection 3.4.4). Next to these, a description of the Fluid solver (Subsection 3.4.6), and of the mesh motion solver (Subsection 3.4.5) is provided.

3.4.1. Load interpolator

As can be seen in Figure 3.1, the role of the load interpolator module is to map the local loads obtained at the fluid faces on the FSI interface to the nodal FEM loads. The general mathematical model to do so is given by equation (2.91). In particular for the considered beam FEM model, one applies the formula to obtain the concentrated lateral forces and moments for the left and the right nodes of any element, using the convention on the right of Figure 2.5 for the XY bending plane. For the XZ bending plane, the convention sign is applied using the orientation as presented for the displacements in Figure 2.6. In rewriting (2.91) for each of these concentrated loads, the following relations are obtained for the XY bending plane of an arbitrary i$^{\text{th}}$ element:

$$F_{1y,\text{i}} = \int_0^{L_{\text{el}}} q_{\text{s},y,\text{i}}(x_{\text{el}}) N_{1u}^{\text{XY}}(x_{\text{el}})\, \text{d}x_{\text{el}}, \tag{3.5}$$

$$M_{1z,\text{i}} = \int_0^{L_{\text{el}}} q_{\text{s},y,\text{i}}(x_{\text{el}}) N_{1\varphi}^{\text{XY}}(x_{\text{el}})\, \text{d}x_{\text{el}}, \tag{3.6}$$

$$F_{2y,\text{i}} = \int_0^{L_{\text{el}}} q_{\text{s},y,\text{i}}(x_{\text{el}}) N_{2u}^{\text{XY}}(x_{\text{el}})\, \text{d}x_{\text{el}}, \tag{3.7}$$

$$M_{1z,\text{i}} = \int_0^{L_{\text{el}}} q_{\text{s},y,\text{i}}(x_{\text{el}}) N_{2\varphi}^{\text{XY}}(x_{\text{el}})\, \text{d}x_{\text{el}}, \tag{3.8}$$

and similarly for the XZ bending plane:

$$F_{1z,\text{i}} = \int_0^{L_{\text{el}}} q_{\text{s},z,\text{i}}(x_{\text{el}}) N_{1u}^{\text{XZ}}(x_{\text{el}})\, \text{d}x_{\text{el}}, \tag{3.9}$$

$$M_{1y,\text{i}} = \int_0^{L_{\text{el}}} q_{\text{s},z,\text{i}}(x_{\text{el}}) N_{1\varphi}^{\text{XZ}}(x_{\text{el}})\, \text{d}x_{\text{el}}, \tag{3.10}$$

$$F_{2z,\text{i}} = \int_0^{L_{\text{el}}} q_{\text{s},z,\text{i}}(x_{\text{el}}) N_{2u}^{\text{XZ}}(x_{\text{el}})\, \text{d}x_{\text{el}}, \tag{3.11}$$

$$M_{1y,\text{i}} = \int_0^{L_{\text{el}}} q_{\text{s},z,\text{i}}(x_{\text{el}}) N_{2\varphi}^{\text{XZ}}(x_{\text{el}})\, \text{d}x_{\text{el}}. \tag{3.12}$$

Within equations (3.5)-(3.12), one knows the values for the shape functions, as presented in (2.73)-(2.76), and in (2.84)-(2.87). However, the other members of the equations, $q_{\text{s},y,\text{i}}$ and $q_{\text{s},z,\text{i}}$, are unknown. They represent the distributed 1D lateral load in the $y-$ and the $z-$ direction, respectively. Furthermore, a numerical technique to compute the integrals over the elements needs to be chosen to obtain the nodal loads. Finally, another challenge is posed by the fact that the inputs to the load interpolator, that is, the fluid information on the finite-volume cell faces at the FSI boundary, are provided within the coordinate system of OpenFOAM.

On the other hand, equations (3.5)-(3.12) are written in the local reference of the beam elements, where it is possible that the two coordinate systems do not coincide. Hence, in using the load interpolator, three aspects need to be considered: the determination of the 1D distributed loads $q_{s,y,i}$ and $q_{s,z,i}$ in the OpenFOAM reference system, their transformation into the beam element reference system, and the technique for computing the integrals in equations (3.5)-(3.12). Each of these is treated individually in Subsections 3.4.1.1-3.4.1.3.

3.4.1.1. Computation of the 1D distributed loads

To compute the values of $q_{s,y,i}$ and $q_{s,z,i}$, other studies that use beam-element FEM solvers for FSI studies of TIV employ a simplified approach, where the 1D distributed loads are assumed to be uniform over each element. This is used, for example, by Liu et al. [64, 65], paired with an LES model. Vivaldi and Ricciardi [63] coupled the beam element only to a URANS fluid model for validation purposes, but their end goal is to use it for a final workflow containing an LES model. These studies suggest that one could in principle use constant distributed loads over a beam element for TIV simulations. However, at the beginning of the thesis, this was thought to be an assumption that is too limiting. This is because the use of a constant load distribution over the entire beam element is equivalent to computing an average of all of the flowfield variations over it. Since the ratio between the number of fluid and structural nodes on the FSI interface was expected to be quite large, the averaging would be done over a significant number of fluid elements, which could reduce the benefits associated with the high spatial resolution of the fluid solver. Consequently, for a uniform load distribution assumption, using a coarse structural discretization could be detrimental to the FSI accuracy and convergence. Even more importantly, it was believed that in using a different assumption for the variation of $q_{s,y}$ and $q_{s,z}$ over a given element, one could use a coarser structural grid, hence improving the computational costs of the workflow. Nevertheless, for redundancy and comparison purposes, a load mapping model where the forces are assumed to be constant over the element was also implemented, as a particular case of the method that is proposed in the following.

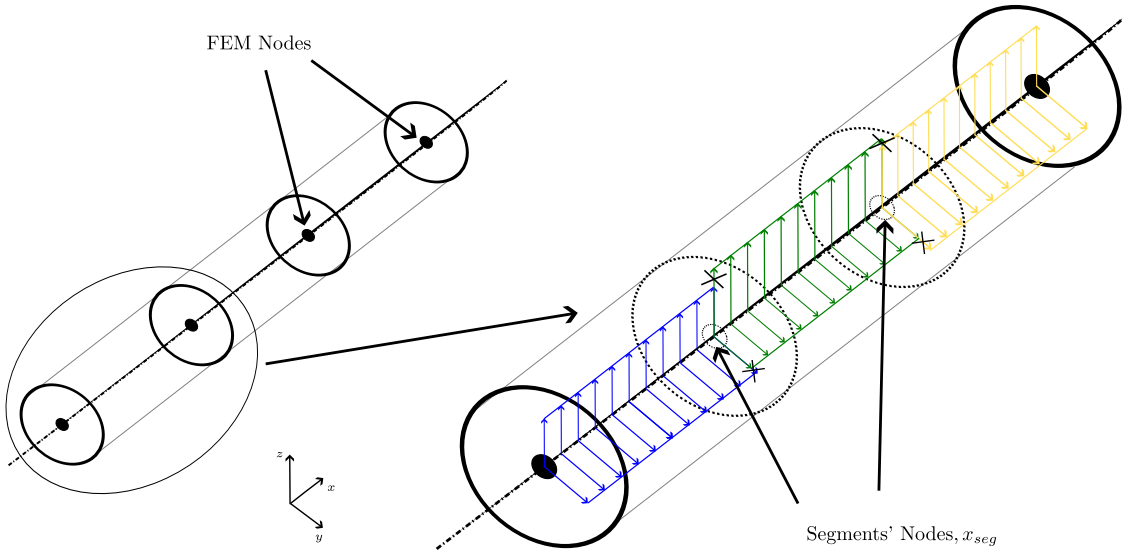


Figure 3.4: A visual representation of the subdivision of each beam element into segments, where for each segment a constant 1D load distribution is assumed for each of the two bending planes

Hence, to optimize the number of unknowns within the structural solver, for this thesis, **it is assumed that the 1D distributed loads $q_{s,y}$ and $q_{s,z}$, are piecewise continuous over a given beam element**. This leads to an additional subdivision of all beam elements, that is only formalized within the load interpolator. In the remainder of this work, those will be called segments, and they are assumed to be of equal length. The total number of segments for each of the elements is imposed by the user as an input to the problem. A representation of the segments' model is presented in Figure 3.4, where one can observe the addition of the segments' nodes to the main axis of the beam elements. For clarity, the load distributions are already represented in the beam reference system, though the method of how to do so is presented in Subsection 3.4.1.2.

Having presented the assumed variation of the distributed loads $q_{s,y}$ and $q_{s,z}$ over an element, it is explained how these are obtained starting from the fluid information onto the FSI interface. In doing so, one can resort to one of two options. The first is to project the pressures of the fluid onto the imaginary external surface of the segments, and use this 3D field to obtain the distributed loads. This is consistent with satisfying the stress equilibrium conditions on the entire FSI boundary, for the maximum level of discretization that the solution formulation allows. On the other hand, one could also transfer a smaller number of concentrated *forces* to the 1D structure, hence only satisfying the stress condition at the points where the loads are applied. While the former is the rigorous approach, the latter is significantly cheaper, and it represents a reasonable approach for as long as the size of the cell faces is significantly smaller than that of the outer surface of a segment. Hence, the transfer of concentrated forces from the fluid to the structure is preferred in this study.

To assign the fluid forces to their corresponding segment, when the numerical problem is initialized, a connectivity matrix is computed between the fluid cell face centers and all of the segments used in the discretization of the structural problem. The entries within this connectivity matrix are simply based on computing the position of a given cell face center relative to the axial location of the segments' nodes, for the initial (undeformed) shape of the fluid mesh, using the beam coordinate system. Thus, if the fluid cell face center has an axial position that is smaller or equal to that of the node of a given segment that is farthest from the beam origin, the entire force associated with that cell is transferred to the said segment. Furthermore, if the entries within the connectivity matrix correspond to the situation where no fluid cell face centers are assigned to any of the segments, that is, the segments are locally smaller than the cell faces, an error is thrown by the solver. This is done to mitigate the effect of only considering the FSI stress equilibrium condition in a pointwise manner, as mentioned in the paragraph above. After the initialization, each time the load interpolator module is called, the connectivity matrix is reused, such that the current cell face forces are incremented to the corresponding segment.

Once the total forces applied onto each segment are computed, one needs to transform these into 1D distributed loads. This is done by assuming that the segment is rigid throughout the deformation, and that the forces are evenly distributed over its theoretical outer surface. For the considered application, the former is a reasonable assumption: the amplitudes of the vibrations due to TIV are never expected to be larger than 1% of the rod's diameter. The latter hypothesis is consistent with the unidimensionality of the beam model, when combined with the above hypothesis that in each bending plane the segments are uniformly loaded. Under these assumptions, for an arbitrary j$^{\text{th}}$ segment, the average distributed loads in each coordinate direction of the OpenFOAM reference system, $\boldsymbol{q}_{\mathbf{s,\,OF,\,j}}$, can be obtained by computing the average pressure force over the segment, and subsequently multiplying this by the circumference of the undeformed rod:

$$\boldsymbol{q}_{\mathbf{s,\,OF,\,j}} = \frac{\overbrace{\sum_{\text{faces, j}} A_{\text{faces, j}} \left(p\mathbf{n} + \boldsymbol{\tau}_{\mathbf{f,ik}} \cdot \mathbf{n}\right)_{\text{faces,j}}}^{\text{total cell face forces assigned to segment j}}}{\underbrace{\sum_{\text{faces, j}} A_{\text{faces, j}}}_{\text{total cell face area assigned to segment j}}} \cdot \pi D, \tag{3.13}$$

where faces, j denotes the fluid cell faces on the FSI boundary associated with the j$^{\text{th}}$ segment, p represents the fluid pressure value stored at each cell face, $\boldsymbol{\tau}_{\mathbf{f,\,ik}}$ the effective stress tensor (combining both the turbulent and the viscous effects), $\mathbf{n}$ the cell face normal, and D the undeformed rod diameter. It is worth noting that in (3.13) p refers to the mean pressure for laminar and URANS fluid models, and to the instantaneous pressure when used with AniPFM, where on top of the URANS average pressure, one adds the pressure fluctuations. The fluid models were detailed in Section 2.2.

Finally, the main sources of modelling errors as seen by the author for the chosen approach for computing the 1D distributed loads are presented, along with recommendations to mitigate them. Apart from the previously mentioned issue of transferring loads instead of stresses, equation (3.13) assumes that the external surface of the FSI boundary is perfectly cylindrical. In practice, this is in fact discretized using finite volumes of the fluid mesh, and can therefore be seen as a combination of straight cell faces trying to emulate the circular surface. Thus, the error that is introduced by this assumption increases the coarser the fluid discretization is. Nevertheless, it was expected that for the fluid discretization that is required for AniPFM simulations, the

error introduced would be negligible, especially compared to the other current uncertainties of the mode [41]. What's more, in using the segments, one equivalates the total forces applied onto a potentially irregular surface formed by the fluid cell faces onto a cylindrical domain associated with a segment, leading to a redistribution of the fluid forces applied onto the structure. This issue is especially relevant for unstructured grids, and it is mitigated by having the cell faces be relatively small compared to the segment. If this condition is achieved, then the surface formed by the fluid cell faces associated with a given segment will closely resemble its hypothetical cylindrical shape. On the other hand, using a small number of segments is not desirable either, since one risks aliasing important fluid features. Hence, in principle, there should exist an optimum number of segments that one would use within the fluid simulation.

3.4.1.2. Coordinate transformation between the OpenFOAM and the beam coordinate system

Using equation (3.13), the average distributed loads for a given segment are obtained in the OpenFOAM reference system. However, the structural solvers used herein are constructed by assuming that the axial direction corresponds to the $x-$ axis of the reference system. Such a choice of coordinates is denoted in this study as a beam reference system, and an example is visible in Figure 3.4. To describe the transformation between the two frames of reference, the terminology x_{OF}, y_{OF}, and z_{OF} will be used to describe the OpenFOAM axes, while x, y and z refer to the beam reference system. The one user input required by the transformation is the normalized direction vector of the main axis of the beam in the OpenFOAM reference system, called $\boldsymbol{L_{\text{OF}}}$. The normalized direction of the rod's main axis in the beam reference system, $\boldsymbol{L_{\text{beam}}}$, is hence equal to $(1,0,0)$. The purpose of this Subsection is to determine the rotation matrix, $\boldsymbol{R_{\text{rot}}}$, such that:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \boldsymbol{R_{\text{rot}}} \begin{bmatrix} x_{\text{OF}} \\ y_{\text{OF}} \\ z_{\text{OF}} \end{bmatrix}. \tag{3.14}$$

The transformation between the OpenFOAM reference system is obtained as a counterclockwise rotation around the axis perpendicular to both $\boldsymbol{L_{\text{OF}}}$ and $\boldsymbol{L_{\text{beam}}}$, by an angle θ_{rot}. The direction vector of the normal, $\boldsymbol{e_{\text{rot}}}$, is obtained by normalizing the cross product between $\boldsymbol{L_{\text{OF}}}$ and $\boldsymbol{L_{\text{beam}}}$, equation (3.15). The angle θ_{rot} is obtained based on the cross-product and the scalar product of $\boldsymbol{L_{\text{OF}}}$ and $\boldsymbol{L_{\text{beam}}}$. Since the former is proportional to the sine of θ_{rot}, and the latter to the cosine of θ_{rot}, the arctangent function can be applied to the ratio between the two to obtain θ_{rot}, as is done in equation (3.16).

$$\boldsymbol{e_{\text{rot}}} = \frac{\boldsymbol{L_{\text{OF}}} \times \boldsymbol{L_{\text{beam}}}}{|\boldsymbol{L_{\text{OF}}} \times \boldsymbol{L_{\text{beam}}}|}, \tag{3.15}$$

$$\theta_{\text{rot}} = \tan^{-1}\left(\frac{\boldsymbol{L_{\text{OF}}} \times \boldsymbol{L_{\text{beam}}}}{\boldsymbol{L_{\text{OF}}} \cdot \boldsymbol{L_{\text{beam}}}}\right). \tag{3.16}$$

Having determined $\boldsymbol{e_{\text{rot}}}$ and θ_{rot}, one can obtain the rotation matrix $\mathbf{R_{\text{rot}}}$ as [134]:

$$\mathbf{R_{\text{rot}}} = \begin{bmatrix} \cos\theta_{\text{rot}} + e_1^2(1-\cos\theta_{\text{rot}}) & e_1 e_2(1-\cos\theta_{\text{rot}}) - e_3\sin\theta_{\text{rot}} & e_1 e_3(1-\cos\theta_{\text{rot}}) + e_2\sin\theta_{\text{rot}} \\ e_2 e_1(1-\cos\theta_{\text{rot}}) + e_3\sin\theta_{\text{rot}} & \cos\theta_{\text{rot}} + e_2^2(1-\cos\theta_{\text{rot}}) & e_2 e_3(1-\cos\theta_{\text{rot}}) - e_1\sin\theta_{\text{rot}} \\ e_3 e_1(1-\cos\theta_{\text{rot}}) - e_2\sin\theta_{\text{rot}} & e_3 e_2(1-\cos\theta_{\text{rot}}) + e_1\sin\theta_{\text{rot}} & \cos\theta_{\text{rot}} + e_3^2(1-\cos\theta_{\text{rot}}), \end{bmatrix} \tag{3.17}$$

where $e_1, e_2,$ and e_3 represent the components of the normalized rotation vector, $\boldsymbol{e_{\text{rot}}^T} = (e_1\ e_2\ e_3)$. Using (3.14) and (3.17), the displacements, coordinate vectors, and distributed loads can be transformed from the OpenFOAM to the beam reference system. For example, the distributed loads in the beam reference system, $\boldsymbol{q_{\text{s, j}}}$, are obtained as:

$$\boldsymbol{q_{\text{s, j}}} = \boldsymbol{R_{\text{rot}}}\boldsymbol{q_{\text{s, OF, j}}}. \tag{3.18}$$

Based on (3.18), the constant values of the distributed loads $q_{\text{s},y}$ and $q_{\text{s},z}$ over a segment, as integrated in (3.5)-(3.12), are obtained as the second and third components of $\boldsymbol{q_{\text{s, j}}}$.

3.4.1.3. Choice of integration technique

Using the techniques presented in Subsections 3.4.1.1 and 3.4.1.2, one can compute the pointwise values of the integrands in (3.5)-(3.12) at all axial locations. The only thing that needs to be done before the concentrated nodal forces are obtained is to determine the integration technique with respect to the axial position in the said equations.

In this thesis, a Simpson 3/8 integration rule is used [135]. This belongs to the family of Newton-Cotes integration formulas, where the integrands are approximated as simple functions, for which the integration value is known analytically. For example, the trapezoidal method approximates the function being integrated as a linear polynomial, while the Simpson 1/3 rule as second-order one. However, the Simpson 3/8 rule was chosen because for this quadrature rule the integrand is approximated by a third-order Lagrange polynomial [135]. This choice of the approximation is consistent with the shape functions used for the displacements, which are of third order for the translational, and second-order for the rotational degrees of freedom, as can be seen from equations (2.73)-(2.76), and (2.84)-(2.87). Given the order of the shape functions, it was therefore considered as the most adequate approach for numerical integration. Consequently, in using the Simpson 3/8 rule, if uniform load distributions $q_{s,y}$ or $q_{s,z}$ are imposed over the entire element, the exact nodal forces and moments are obtained, according to equations (3.5)-(3.12). The general formula for the Simpson 3/8 rule stipulates that the integral I_{seg} of the integrand f_{seg} from 0 to L_{el} is [135]:

$$I_{\text{seg}} \approx \frac{3}{8} L_{\text{el}} \left[f_{\text{seg}}(x_{seg,0}) + 3 \sum_{\text{i}=1,3\nmid\text{i}}^{N_{\text{seg}}-1} f_{\text{seg}}\left(x_{seg,\text{i}}\right) + 2 \sum_{\text{i}=1}^{N_{\text{seg}}/3-1} f_{\text{seg}}\left(x_{seg,3\text{i}}\right) + f_{\text{seg}}(x_{seg,N_{seg}}) \right]. \tag{3.19}$$

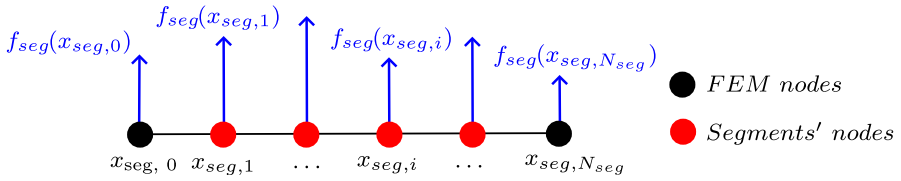


Figure 3.5: Visual representation of the terms in equation (3.19)

For a better understanding of (3.19), the different terms are visually represented in Figure 3.5. $x_{\text{seg, i}}$ refers to the axial location of the segments' *nodes*. For each element, the first and the last nodes of the segments' correspond to the nodes of the FEM grid. To compute the integrals in (3.5)-(3.12), one simply needs to use (3.19), replacing I_{seg} with the concentrated nodal loads on the left-hand side, and f_{seg} with the corresponding integrands on the right-hand side. While the values of the shape functions are known from their analytical definitions, the values of the 1D distributed loads at the nodes of the segments $q_{s,y}(x_{\text{seg}})$ and $q_{s,z}(x_{\text{seg}})$ are obtained by computing the mean between the constant values of the distributed loads stored in adjacent segments. This is visualised using black crosses at the inner segments' interface in Figure 3.4. It is also worth noting that as a consequence of the use of the Simpson 3/8 rule, the number of segments for each element needs to be a multiple of 3 [135]. If this is not the case, the solver is stopped prematurely, and the user notified.

3.4.2. FEM and ROM solver

Using the approach presented in Subsection 3.4.1, the nodal loads can be computed in the *local* frame of reference of each beam element. Indeed, the theory presented in Subsections 2.3.4 and 2.3.6 only refers to a single element, isolated from the rest of the structural discretization. Nevertheless, for a unique solution of the displacements, the solid equilibrium can only be solved using a global formulation [81]. First, to compute the element stiffness and mass matrices, equations (2.90) and (A.1), the FEM solver takes as input the geometrical and physical properties of the rod from the `BeamFoamDisplacement` subdictionary, as visualised in Figure 3.3. Similarly, the method also reads the values of γ and of α for the Newmark algorithm, and of the initial value of the underrelaxation factor $\omega_{\text{UR},0}$ for the subiteration scheme. Subsequently, starting from the local matrices and concentrated loads, the technique presented by Khennane [87] is used to form the global system. It is worth mentioning that although usually a rotation transformation between the local and global frame of reference is required at this stage for general FEM solvers, this is not the case for the work herein, since the two stretch along the same axis, given the straight and cylindrical shape of the structural domain. The process of forming the global system requires building two matrices. First, a connectivity matrix is built, where for each of the two nodes of an element a global node number is assigned. Next, the active degrees of freedom for each global node are numbered, and the data is stored into a nodal freedom matrix. Using the two, one can iterate over the element matrices, and add the corresponding inertia and stiffness for each degree of freedom to the global mass and inertia matrices.

The ROM solver is constructed as a subclass of the FEM solver. The user can impose both the number of modes used to reduce the structure, as well as whether a Mode Displacement or a Mode Acceleration ROM is preferred. It is important to note that an even number of modes needs to be used if both bending planes are active inside the structural solver to accurately reconstruct the structural solution. When constructed, the ROM solver computes and stores the eigenmodes and eigenvalues resulting from (2.98), and subsequently obtains (2.102). Once this is solved for the modal coordinates $boldsymbola_{\mathbf{s}}$, those are transferred back to the nodal FEM using (2.100) for the MD, and (2.105) for the MA approach.

3.4.3. Subiterator

It is important to remember that the desired FSI workflow is two-way coupled, that is, the loop in Figure 3.1 may need to be computed multiple times at each timestep before an acceptable similarity between the fluid loads and the corresponding structural displacements is obtained at the FSI interface. As was explained in Subsection 2.1, this is done within the field of FSI using an underrelaxation approach based on a fixed-point iteration scheme.

Given the inclusion of the FSI loop inside the PIMPLE algorithm, as described in Subsection 3.2, the structural solution is obtained before that of the fluid at a given FSI subiteration. Consequently, the proposed FSI methodology is a serial, Fluid-Structure (F-S) workflow. Furthermore, the Aitken method will be used, equation (2.9). This choice is based on a good balance between speed of convergence and ease of implementation [136]. At each timestep, the same initial value of the underrelaxation factor $\omega_{\mathrm{UR},0}$ is used. The vector used in (2.9) is that of the smallest set of degrees of freedom that uniquely describe the structural displacements. More specifically, if the user chooses the FEM solver, this is equal to the nodal displacements vector, $\boldsymbol{u}_{\mathbf{s}}$ in equation (2.58), or to the generalized modal coordinate $\boldsymbol{a}_{\mathbf{s}}$ from the decomposition (2.100) for a ROM solver. The choice of using the displacements, rather than the loads, for the underrelaxation, is arbitrary, and it is based on the fact that when using the underrelaxation of the displacements for the NRG-FSIFOAM workflow, good convergence is obtained. Finally, the number of FSI subiterations that one can use for a given timestep is constant throughout the transient, and it can be imposed by setting the desired number of Outer Correctors (OC) for the PIMPLE algorithm.

Having presented the current characteristics of the subiteration module within the proposed FSI workflow, a comparison is drawn with the previous NRG-FSIFOAM workflow, and some of the current limitations are pointed out. For the NRG-FSIFOAM workflow used previously by van den Bos [38] or Freitas [41], an IQN-ILS algorithm available within preCICE was used, for a parallel coupling scheme. A slight loss of computational efficiency currently occurs when compared to the NRG-FSIFOAM approach, as the IQN-ILS iteration method is known to converge faster than the Aitken method [58]. Furthermore, the old FSI workflow was able to control the number of iterations executed at each timestep based on the value of the FSI residual obtained at two consecutive subiterations. This cannot be replicated within the PIMPLE algorithm, unless one modifies the `pimpleControl` class in OpenFOAM, responsible for checking the convergence of the outer loop of the PIMPLE algorithm. The current use of a fixed number of FSI subiterations at each timestep not only leads to an increase in the computational expenses, but also to the necessity of conducting a manual study by the user of the number of OC required for each FSI subiteration. Hence, both the fixed-point scheme and the control of the number of FSI subiterations at a given timestep are points where the current approach can be improved.

3.4.4. Displacement interpolator

The purpose of the displacement interpolator module, as visible from Figure 3.1, is to map the structural displacements at the nodes of the structure to the fluid nodes on the FSI boundary. Those displacements consist of both translations and rotations, as explained in Subsection 2.3.4. It is important to point out that in using a 1D beam element FEM discretization at the FSI interface one can choose to not deal with transferring information from one non-conformal grid to another, but rather with doing so between an idealized continuous solid surface, as given by the Bernoulli beam element model, and the discrete finite volume grid of the fluid solver. Consequently, in imposing the fluid nodes' displacements on the FSI boundary, one can assume that these belong to the idealized beam surface, without having to use the information from the neighbouring structural nodes, as is done, for example, using a compactly supported Radial-Basis Function (RBF) approach [137].

To construct the said idealized outer surface of the structure starting from the nodal solution inside the FEM solver, it is reminded that in building the FEM problem, one assumed in equation (2.49) that the displacements inside a given element i can be fully described by using the shape functions and the nodal displacements. Rewriting the same equation for the Bernoulli beam element model, using only the nodal displacements associated with the i$^{\text{th}}$ element, and for the XY bending plane, one obtains equation (2.71). For a given i$^{\text{th}}$ element, and for both of the bending planes, the lateral translations $u_{\text{s},y,\text{i}}$ and $u_{\text{s},z,\text{i}}$ at a given axial position within the element, $x_{\text{el, i}}$, are given by:

$$u_{\text{s},y,\text{i}}(x_{\text{el, i}}) = \boldsymbol{N}_{\text{i}}^{\text{XY}}(x_{\text{el,i}}) \begin{bmatrix} u_{1y,\text{i}} \\ \varphi_{1z,\text{i}} \\ u_{2y,\text{i}} \\ \varphi_{2z,\text{i}} \end{bmatrix}, \tag{3.20}$$

$$u_{\text{s},z,\text{i}}(x_{\text{el, i}}) = \boldsymbol{N}_{\text{i}}^{\text{XZ}}(x_{\text{el, i}}) \begin{bmatrix} u_{1z,\text{i}} \\ \varphi_{1y,\text{i}} \\ u_{2z,\text{i}} \\ \varphi_{2y,\text{i}} \end{bmatrix}. \tag{3.21}$$

Equations (3.20) and (3.21) describe the deformation of a cross-section located at the axial position x_{el} within the local coordinate system of the i$^{\text{th}}$ element, regardless of the circumferential position. At the same time, based on the kinematics of the Bernoulli beam model presented in equations (2.62) and (2.82), the corresponding rotations at the same axial position can be computed as:

$$\varphi_{z,\text{i}}(x_{\text{el, i}}) = \frac{\text{d}\boldsymbol{N}_{\text{i}}^{\text{XY}}(x_{\text{el,i}})}{\text{d}x_{\text{el,i}}} \begin{bmatrix} u_{1y,\text{i}} \\ \varphi_{1z,\text{i}} \\ u_{2y,\text{i}} \\ \varphi_{2z,\text{i}} \end{bmatrix}, \tag{3.22}$$

$$\varphi_{y,\text{i}}(x_{\text{el, i}}) = -\frac{\text{d}\boldsymbol{N}_{\text{i}}^{\text{XZ}}(x_{\text{el,i}})}{\text{d}x_{\text{el,i}}} \begin{bmatrix} u_{1z,\text{i}} \\ \varphi_{1y,\text{i}} \\ u_{2z,\text{i}} \\ \varphi_{2y,\text{i}} \end{bmatrix}. \tag{3.23}$$

Equations (3.20)-(3.23) fully describe the deformation of the cross section at axial location $x_{\text{el, i}}$ for the beam model considered in this work, where the continuity of the shape functions and their first derivatives ensure a smooth displacement field. To link the beam deformations to the displacements of the fluid nodes, an absolute formulation of the mesh deformation is preferred, given the large simulation times required for TIV numerical simulations, and the predisposition of relative formulations to accumulate errors throughout the transient [73]. Furthermore, it is assumed that the deformations of the rod are small enough that the axial position $x_{\text{el, i}}$ remains constant throughout the transient. Starting from this, to compute the corresponding displacements of a given fluid node, one simply needs to assign to it a beam element that it belongs to, as well as the value of $x_{\text{el, i}}$. Using these and the current nodal solution, that is, the column vectors in (3.20)-(3.23), one can compute the displacements for each node.

Finally, it is worth remembering that as pointed out in Subsection 3.2, the `DisplacementInterpolator` class needs to give as output the total translations of each fluid node with respect to the initial position, though the beam element nodal solution consists of both bending translations and rotations. To this end, the lateral deformations corresponding to the rotations are computed using the rotation matrices around the $y-$ and the $z-$ axis, $\boldsymbol{R}_{\mathbf{rot,y}}$ and $\boldsymbol{R}_{\mathbf{rot,z}}$, and the current and initial position vector of the fluid node with respect to the beam axis, $\boldsymbol{r}_{\mathbf{beam}}$, and $\boldsymbol{r}_{\mathbf{beam,0}}$:

$$\begin{cases} u_{\text{s},\varphi_{\text{s},z}} = \boldsymbol{R}_{\mathbf{rot,z}}\boldsymbol{r}_{\mathbf{beam}} - \boldsymbol{r}_{\mathbf{beam,0}} = \begin{bmatrix} \cos\varphi_{\text{s},z} & -\sin\varphi_{\text{s},z} & 0 \\ \sin\varphi_{\text{s},z} & \cos\varphi_{\text{s},z} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ r_{\text{beam},y} \\ r_{\text{beam},z} \end{bmatrix} - \begin{bmatrix} 0 \\ r_{y,0} \\ r_{z,0} \end{bmatrix} \\ u_{\text{s},\varphi_{\text{s},y}} = \boldsymbol{R}_{\mathbf{rot,y}}\boldsymbol{r}_{\mathbf{beam}} - \boldsymbol{r}_{\mathbf{beam,0}} = \begin{bmatrix} \cos\varphi_{\text{s},y} & 0 & \sin\varphi_{\text{s},y} \\ 0 & 1 & 0 \\ -\sin\varphi_{\text{s},y} & 0 & \cos\varphi_{\text{s},y} \end{bmatrix} \begin{bmatrix} 0 \\ r_{\text{beam},y} \\ r_{\text{beam},z} \end{bmatrix} - \begin{bmatrix} 0 \\ r_{y,0} \\ r_{z,0} \end{bmatrix} \end{cases}, \tag{3.24}$$

where the convention for the determination of $\boldsymbol{r}_{\mathbf{beam}}$ is presented in Figure 3.6 for a fluid node P. By summing up the contributions in (3.20), (3.21), and in (3.24), the total lateral displacements corresponding to the nodal FEM solution is obtained for a given fluid node at an axial position $x_{\text{el,i}}$. However, those have been obtained in the beam reference system. To determine the inputs to be passed to the mesh motion solver, these need to be mapped back to the OpenFOAM reference system. This is done by inverting the transformation matrix $\mathbf{R}_{\mathbf{rot}}$ in (3.14):

$$\begin{bmatrix} x_{\mathrm{OF}} \\ y_{\mathrm{OF}} \\ z_{\mathrm{OF}} \end{bmatrix} = \boldsymbol{R}_{\mathbf{rot}}^{-1} \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \quad (3.25)$$

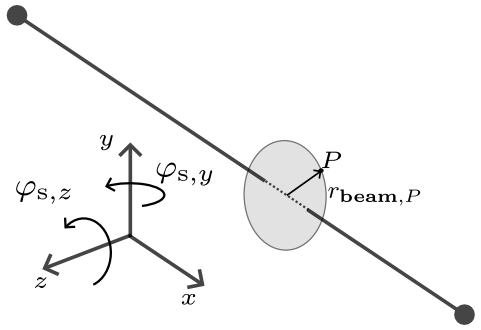


Figure 3.6: Visual representation of the terms in equation (3.24)

3.4.5. Mesh motion solver

The mesh motion solver is outside the main implementation objective of this thesis, but key characteristics are nevertheless presented, along with additional references for the interested reader.
For this application, a Laplacian smoothing approach is used to obtain the deformations of the fluid nodes of the mesh [138]. As implied in Section 3.2, the provided boundary conditions are in the form of displacements of the boundaries of the fluid domain. In practice, all of the surfaces are fixed throughout the transient simulation, except for the FSI interface, onto which the NRG-beamFoam methodology presented in this thesis is applied. Within the Laplace equation, a diffusivity parameter that is inversely proportional to the distance between a fluid node and the FSI interface is added. This has the effect of limiting the propagation of the deformations associated with the dynamic movement of the FSI boundary into the fluid domain. Once the Laplace equation is solved for the displacements, the new deformation is applied onto the `fvMesh` class in OpenFOAM. It is also important to note that although the user only provides as input the structural displacements at the FSI interface, once these are known for all of the boundaries, the velocity boundary condition at the FSI interface are updated accordingly. In turn, this implies that the dynamic FSI boundary condition is fulfilled.

Though the method is known to have certain disadvantages, such as the relatively high computational cost, and the lack of control over the distance between the fluid nodes along the normal of the FSI boundary [138], this is the same mesh motion solver that was also used by van den Bos [38], or by Freitas [41], for the NRG-FSIFOAM that is desired to be replaced. Hence, for the reason of reproductibility and to be able to better explain the differences observed between the two workflows, the exact same motion solver settings were used.

3.4.6. Fluid solver

The equations associated with the fluid solver was presented in Section 2.2. For this thesis, they are solved using the OpenFOAM [42] library. In Section 2.2.1, it was shown that an integral formulation of the Navier-Stokes can be used to discretize the fluid domain into multiple Finite-Volume cells. Numerically, this integral equation, (2.19), is solved using Gauss quadratures, and to compute the fluxes between the neighbouring cells it is required that different quantities are obtained at the cell faces. This is done by utilizing interpolation, starting from the known solution computed at the finite volume cell centers. It is thus required that the user imposes how the interpolation is executed for the different terms. Apart from these settings, which mathematically define the fluid computations at a given time instance, a time integration scheme to advance the flow must also be specified.

Within the OpenFOAM framework, all of the settings for the inputs described in the previous paragraph are provided through the `fvSchemes` dictionary, where an example used for laminar simulations is provided in Appendix D, Subsection D.1.1. Within `fvSchemes`, the user provides different subdictionaries indicating the interpolation settings desired for the Gauss quadratures of the different terms in the Navier-Stokes equations (gradient, divergence, and Laplacian, using the `gradSchemes`, `divSchemes`, and `laplacianSchemes`

subdictionaries, respectively). Furthermore, the interpolation settings for obtaining the flux at the cell faces for the convective terms are specified to the fluid solver using `interpolationSchemes`, the dot product between the gradient of a quantity and the cell face surface normal for non-orthogonal meshes is interpolated based on `snGradSchemes`, and the time marching scheme is imposed using `ddtSchemes`. The last entry of the `fvSchemes` dictionary, `wallDist`, is associated with the technique to compute the distance between the fluid centers and boundary faces to the closest wall. This is relevant, for example, for calculating wall-normal gradients. Further, when using the SST k-ω [75] turbulence model, new scalar fields are generated for the turbulent kinetic energy k and the specific dissipation rate ω, and separate transport equations are solved for each, as presented in equations (2.25)-(2.26). Thus, additional entries in `fvSchemes` must be provided for k and ω.

The use of the AniPFM fluid model implies that new fields need to be computed for the fluid domain, that is, the pressure and the velocity fluctuations, p' and $\mathbf{u_t}$. What's more, the convection of the non-dimensional velocity fluctuations $\mathbf{v_t}$ within the AniPFM model, (2.43), leads to the need to solve additional transport equations within the fluid domain. By using the AniPFM, the Reynolds stress tensor $\boldsymbol{R}$ is also required at the cell faces to transform the values of $\mathbf{v_t}$ to those of $\mathbf{u_t}$. Hence, the AniPFM framework requires imposing new discretization schemes for the interpolation of quantities depending on not only $\mathbf{u_t}$ and $\mathbf{v_t}$ at the cell faces, but also on $\boldsymbol{R}$. Because of this, compared to the URANS FSI cases, additional entries need to be added to the `fvSchemes` dictionary.

Using the imposed discretization schemes for solving the Navier-Stokes, linearized systems are obtained for the unknown fluid fields. Additionally, since a mesh motion solver is used, an additional equation must be solved for the mesh deformations in each direction of the reference system. Once this is done, pressure corrections associated with the mesh motion movement are computed. Given the high dimensionality of each of these systems, the solution for each is computed iteratively, rather than directly, until the residuals obtained between consecutive iterations are considered small enough to assume convergence. Consequently, settings corresponding to the approximate solution of the fluid equations need to be provided.

In OpenFOAM, the solver settings for each of the fields that need to be computed to obtain the solution are provided through the `fvSolution` dictionary. In the same file, specific settings associated with the PIMPLE algorithm are also passed. For the PIMPLE algorithm, the user can impose residual criteria for the last outer corrector at a given timestep that are different from the ones that were used for all previous outer correctors at the said timestep. This is done, for example, for the pressure field `p`, through a corresponding `pFinal` subdictionary entry within `fvSolution`. The values imposed in the `p` subdictionary will therefore serve as residuals for the initial outer corrector loops, while those in `pFinal` only for the last outer corrector at each timestep. As an example, solver settings used for the laminar FSI cases in this work are presented in Appendix D, Subsection D.1.2. Similarly to the discretization schemes, the use of more complex fluid models, such as URANS or AniPFM, leads to additional PDE's that need to be solved, and thus, to the requirement of imposing additional solver settings in `fvSolution`.

When using the AniPFM, special attention must be paid to the prescription of the boundary conditions for the inlet and the outlet. Dirichlet boundary conditions are used for both the pressure and the velocity fluctuations, as this was argued to be more reliable than using Neumann boundary conditions by van den Bos [38], given a potential risk of obtaining a spurious linear or a parabolic pressure profile inside the domain. What's more, if one used the same uniform inflow velocity profile as for the URANS case, then the flow would decay to turbulence only downstream of the inlet [38]. In reality, for the original experiment, the turbulence was already well developed at the inlet of the numerical domain considered for this simulation. To avoid this modelling error, the non-uniform velocity fluctuations for the inlet and the outlet are obtained by mapping the values of the fluctuation fields from axial locations inside the brass beam domain. This is done using the `mapped` BC in OpenFOAM, where a mean value of 0 for the fluctuations in all directions is imposed for the inlet and the outlet, according to the Reynolds decomposition principle. The surface from which the velocity fluctuation boundary values are obtained is imposed using the `constant/polyMesh/boundary` dictionary. Both the BC subdictionaries imposed for the inlet and the outlet for the `Ufluct` field, as well as the associated mapping settings in `constant/polyMesh/boundary`, can be analyzed in Section E.3, Appendix E. There, it can be seen that for the inlet the velocity fluctuations are mapped based on the values found 0.1 m downstream of the inlet, and for the outlet based on those 0.1 m upstream. Starting from the velocity fluctuation

fields imposed at the inlet and the outlet of the domains, the zero-average pressure fluctuations are obtained as [38, 41]:

$$p' = \frac{\rho u'^2}{\sqrt{2}} - \frac{\overline{\rho u'^2}}{\sqrt{2}}. \tag{3.26}$$

3.5. Test cases

For the verification and validation of the methodology proposed in this thesis, it is preferable to use an experimental study of TIV that is relevant for axial flow FSI in nuclear reactor applications, such that the main physics for which the model is built are present. It is also desirable that it represents a numerically inexpensive case, to speed up the verification and validation process, and that the benchmark is well-established and has been simulated numerically by other researchers in the past, to be able to contextualize the results obtained herein. In particular, it is best if the NRG-FSIFOAM workflow was already used on the said benchmark case, as this would save time in setting up the numerical simulation. Being the only case satisfying all of these criteria, the experiment of Chen and Wambsganss [1] was chosen. Though other established benchmark cases exist, such as the experimental study of Lillberg et al. [139] for an instrumentation guide tube, the computational costs associated with the Chen and Wambsganss study are significantly lower, since the used grid is approximately five times smaller [38, 50]. It is also the main case that has been used in the recent development of the AniPFM model, both by van den Bos [38], and more recently by Freitas [41]. This allows for a direct comparison between the newly implemented NRG-beamFoam methodology, and the latest, most optimal iteration of NRG-FSIFOAM. Apart from the experiment of Chen and Wambsganss [1], which is predominantly used for the verfication of this thesis, a typical fuel element geometry was also used at the beginning of this thesis. Thus, the two geometries are described in Subsections 3.5.1 and 3.5.2.

3.5.1. Brass beam in turbulent flow

The research findings of Chen and Wambsganss [1] were published in 1972. They presented a particular study of a series of annular flows. The experiments measured the dynamic response of a clamped brass beam to axial water flow excitation at different velocities. At the time, the main objective was to measure the effects of added mass and of added damping of water flow over fuel pins in nuclear reactors. The experiment was done using a brass rod of length of 1.19 m, and a diameter D of 1.27 cm. Hence, the rod is characterized by $\frac{L}{D} = 93.7$. The outer diameter D_O of the experimental channel is equal to 2.54 cm, such that $\frac{D_O}{D} = 2$. The brass was fixed at both of its ends, as is also the case for the application considered herein. The extensive properties of the fluid and of the brass beam are presented in Table 3.1. In the remainder of this work, the experiment Chen and Wambsganss will also be referred to as the brass beam case.

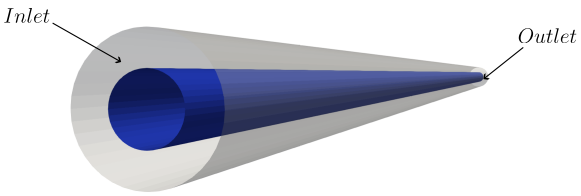


Figure 3.7: Visual representation of the experimental setup of Chen and Wambsganss [1]

Fluid Properties (Water)	
Density (ρ_f)	1000 kg/m^3
Kinematic Viscosity (ν_f)	1×10^{-6} m^2/s
Structural Properties (Brass)	
Density (ρ_s)	8400 kg/m^3
Elasticity Modulus (E)	107 GPa
Poisson Ratio	0.331

Table 3.1: Structural and fluid properties for the Chen and Wambsganss case [1]; taken from [50]

In terms of flow conditions, the test was conducted at 13 velocities between 0 and 36.58 m/s, where for each of these the rod is initially assumed to be still. From the experiment, data such as the natural frequencies, the damping ratios, and the RMS amplitudes of the vibrations are provided. Within literature, results are usually provided at 10, 20, and 30 m/s, which corresponds to a Reynolds range of $\mathrm{Re} = \frac{U_{\mathrm{axial}} D_h}{\nu_f}$ between $1.2 - 3.7 \cdot 10^6$, where $D_h = D - D_c$ represents the hydraulic diameter of the rod. Apart from van den Bos [38], this benchmark case was used by de Ridder et al. [140], Nazari et al. [141], and Kottapalli [39], using different modelling approaches. De Ridder et al. [140] use a URANS approach and an IQN-ILS coupling scheme to compute the modal response of the brass beam case. Nazari et al. [141] used an LES fluid model, and coupled it to a structural solver. All of this was done in Ansys, where the "System coupling" functionality was used to transfer information between the domains (no further information on the coupling algorithm was provided in the

study). Finally, Kottapalli [39] also used an earlier, isotropic version of the AniPFM. For his research, similar to the later studies of van den Bos [38] and Freitas [41], the AniPFM was linked to the deal.II structural solver, where the subiteration between the two was executed using the IQN-ILS algorithm. A comparison between the results of Kottapalli [39] and of van den Bos [38] in terms of eigenfrequencies, damping, and RMS amplitude can be seen in Figure 3.8. Due to time constraints, within this work, FSI simulations were only conducted at 10 m/s.

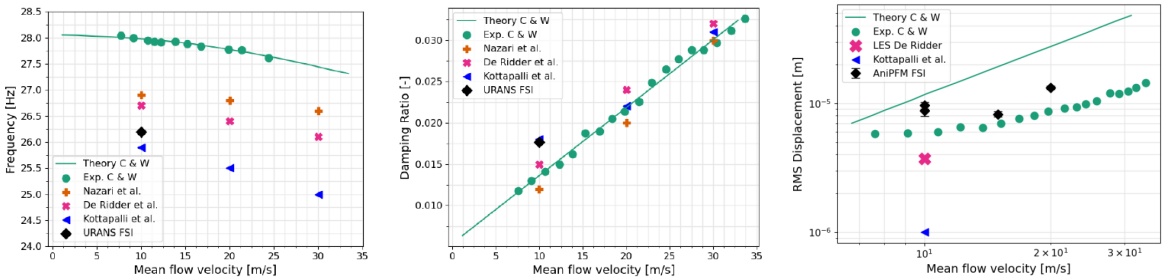


Figure 3.8: Comparison between numerical for different approaches applied to the Chen and Wambsganss case [1], in terms of eigenfrequencies (left), damping ratios (middle), and amplitude RMS due to TIV (right); taken from [38]

3.5.2. Fuel element geometry

In contrast to the rest of this work, rather than the geometry of the brass beam, the structural domain used to validate the FEM solver corresponds to a geometry that is typical for a nuclear fuel element. The structure is clamped at its ends, has a circular cross-section of a diameter equal to 1 cm, a length of four meters, and it is made out of steel assumed to have a density equal to 7850 kg/m^3, with a Young's modulus equal to 200 GPa. A 1D schematic of the structure used is presented in Figure 3.9. For the FEM verification, the structure is always discretized using 10 beam elements, and hence in the same Figure the corresponding FEM nodes are represented.

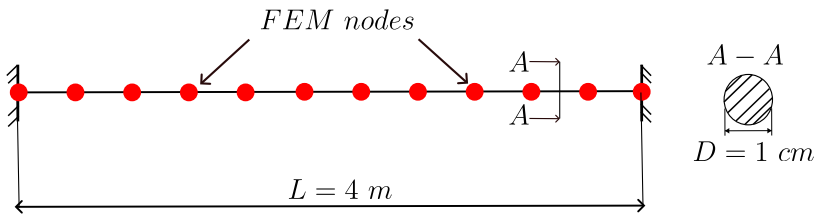


Figure 3.9: 1D schematic representation of the structure used for the FEM solver validation

4

Verification of the main individual components of NRG-beamFoam

As was established in Chapter 3, this work proposes significant changes to the previous NRG-FSIFOAM workflow. With the new structural solvers being implemented directly into OpenFOAM, it is also necessary that the transfer of information is correctly handled between the fluid and the structural solver. This led to the implementation of the load interpolator and displacement interpolator modules. Further, to handle strong coupling between the two domains, an efficient Aitken subiteration algorithm was proposed. Hence, within the work of this thesis, numerous modules of the FSI workflow had to be programmed in OpenFOAM, as can also be seen in the methodology diagram in Figure 3.1. A specific challenge is posed by the fact that all of the different components need to work together seamlessly to obtain reliable results for the FSI behaviour. However, testing the combined working of all of the modules is risky, as the bugs and modelling errors can become significantly more difficult to localize. To pinpoint the potential sources of error more easily, tests were devised for the main components of the newly implemented methodology. Within this Chapter, for each of the different modules, the verification procedure is presented along with the corresponding results. The tests have been undertaken starting from the new FEM solver, the most important modification brought to the previous workflow. This is presented in Section 4.1. Once the working of this module was proven, the verification of the mapping routines is presented in Sections 4.2 and 4.3, for the load interpolator and the displacement interpolator, respectively. The verification of the subiterator and the ROM solver will be treated later in this report, in Chapters 5 and 7, respectively.

4.1. FEM solver

The FEM solver represents the most important module of this work, as it intrinsically defines the structural response within the FSI problem. It was also built "from scratch" for this thesis, and it is therefore crucial that it is thoroughly verified before moving on to the other steps of the research.

The working of the FEM solver follows the general steps of solving a structural problem using the FEM [87]: defining the inputs to the problem (the geometrical and mechanical properties of the structure, the discretization settings, the time integration scheme settings, the loads applied at the structural nodes), defining the local stiffness and mass matrices, assembling the global matrices and vectors in (2.58), and implementing the time integration algorithm based on the assumptions (2.95). Starting from this well-established set of procedures, the FEM solver was verified sequentially, in three steps. The passing of all three tests is therefore intended to certify the working of the FEM solver. Before each test is described, it is worth remembering that in contrast to the other modules presented in this Chapter, the verification of the FEM solver is conducted using the fuel element geometry presented in Subsection 3.5.2.

4.1.1. Tests description

The purpose of the first test is to verify the assembling of the global stiffness matrix and the global forcing vector by comparing the output of the NRG-beamFoam FEM solver to that of a commercial Finite Element Analysis software for a simple static case, where concentrated nodal loads are applied in each bending plane

of the beam, close to its fixed ends. A sketch of the applied loads can be seen in Figure 4.1, where it was considered that $F_y = 10$ N, and $F_z = 8$ N. Though the choice of the values of the loads is arbitrary as long as the response of the structure remains linear, these values were picked to match the amplitude of vibration for the URANS-FSI results of van den Bos [38] for the brass beam case.

As a reference for the static results that ought to be obtained for the first test, the FEM solver Abaqus was used. To ensure similarity between NRG-beamFoam and Abaqus, the same geometric and physical properties of the fuel element structure proposed in Subsection 3.5.2 were imposed for both (length, circular cross-section shape, density, Young's modulus). Furthermore, the same type of Bernoulli elements used to discretize the problems were used, that is, the ones presented in Subsection 2.3.4. Within Abaqus, this consisted of using a B33 linear beam element, which assumes a cubic shape function, and a linear treatment of the structural problem [142].

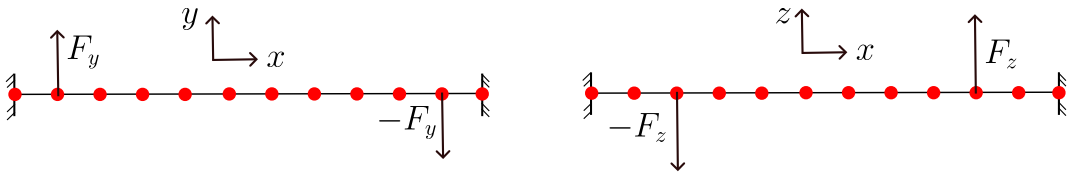


Figure 4.1: 1D schematic representation of the concentrated loads F_y and F_z applied for the verification of the global stiffness and forcing vector

Once the global stiffness matrix and external forcing vector have been checked, in the second verification step the global mass matrix is verified. Since $\boldsymbol{K_s}$ is now known to be correct, the mass matrix $\boldsymbol{M_s}$ can be verified based on the eigensolution that one obtains for the rod. This is because on one hand, the eigenmodes $\boldsymbol{\phi_s}$ and the circular eigenfrequencies ω_s of the fuel element can be obtained numerically starting from the eigenproblem (2.98):

$$\left(\boldsymbol{K_s} - \omega_s^2 \boldsymbol{M_s}\right)\boldsymbol{\phi_s} = \boldsymbol{0}, \tag{2.98}$$

but at the same time, for the simplified fuel element rod geometry, according to the Bernoulli beam model, the q$^{\text{th}}$ circular eigenfrequency and normalized eigenmode can be obtained as [143]:

$$\omega_{s,q} = \beta_q^2 \sqrt{\frac{EI}{\rho_s \frac{\pi D^2}{4}}}, \tag{4.1}$$

$$\boldsymbol{\phi}_{s,q}(x) = \left(\sinh \beta_q x - \sin \beta_q x\right) + \left(\frac{\sinh \beta_q L - \sin \beta_q L}{\cos \beta_q L - \cosh \beta_q L}\right) \cdot \left(\cosh \beta_q x - \cos \beta_q x\right), \tag{4.2}$$

where E, I, ρ_s, D, L refer to the physical and geometrical properties of the fuel element geometry, and β_q can be obtained as the q$^{\text{th}}$ solution of $\cos \beta_q L \cdot \cosh \beta_q L = 1$. Hence, one can use the analytical solutions in equations (4.1) and (4.2) to verify the results obtained when computing (2.98) with the NRG-beamFoam FEM solver. The matching between the two approaches therefore implies that a correct form of the global mass matrix is obtained.

With $\boldsymbol{M_s}, \boldsymbol{K_s}$, and $\boldsymbol{f_s}$ known to be correct, only the numerical time integration scheme needs to be verified. Similar to the second verification step of the FEM solver, this is done by comparing the results of the developed FEM solver to a (dynamic) analytical solution. For this verification test, one can obtain such a reference solution using the Mode Superposition technique. A short background of the approach is provided in Subsection A.2. This method is similar to the modal ROMs proposed in Subsections 2.4.3.2 to 2.4.3.4, in the sense that it is based on projecting the nodal displacements $\boldsymbol{u_s}$ using the eigenmodes of the structure, and solving for the modal coordinates, $\boldsymbol{a_{s,MS}}$. However, the Mode Superposition uses *all* the eigenmodes of a structure to redefine the problem, and it therefore offers an exact solution. However, instead of using a numerical integration technique, the values of the modal coordinates are obtained analytically, using the Duhamel integral formulation, equation (A.6). Since the correct values of the eigenmodes have been proven in the second test

of the FEM solver, they can be readily used within the Mode Superposition benchmark case.

To increase the dynamics of the rod's vibration, and also to simplify the analytical solution, two additional assumptions are made within the third verification test. First, contrary to the rest of this thesis, modal damping is added to the rod's vibration. Second, it is assumed that no external forcing is applied to the fuel element. These two decisions are explained in the following two paragraphs.

The addition of modal damping is justified by the fact that if one doesn't use any structural damping, for the third verification test there would be no source of dissipation of the vibration energy of the rod, since there is no external forcing, and the chosen structural numerical integration scheme conserves the energy of the system, as was already pointed out in Subsection 2.3.7. However, it is of interest for this thesis to ensure that the numerical integration scheme does not influence the total energy received by the structure, as this influences the total vibration amplitudes due to TIV, which is ultimately the main research interest of the FSI simulations.

The choice of not adding external forcing is made to avoid the need to solve the Duhamel integral, equation (A.7), with respect to time. Thus, by using this hypothesis, the analytical solution for the third verification test is greatly simplified. To trigger the vibration of the rod, an initial displacement is applied to the fuel element, after which it is allowed to vibrate freely. For simplicity, the spatial distribution of the displacement corresponds to the first eigenmode of the structure. Under these assumptions, the analytical solution for the nodal displacements $\boldsymbol{u_s}$ according to the Mode Superposition method is computed using:

$$\boldsymbol{u_{s,MS}}(t) = \sum_{i=1}^{N_{FEM}} a_{s,MS,i}(t)\boldsymbol{\phi}_{s,i} = \boldsymbol{\Phi_s}\boldsymbol{a_{s,MS}}(t), \text{ where} \tag{4.3}$$

$$a_{s,MS,i}(t) = e^{-\xi_{s,i}\omega_{s,i}t}\left(\zeta_{s,i}\sin\bar{\omega}_{s,i}t + \eta_{s,i}\cos\bar{\omega}_{s,i}t\right), \tag{4.4}$$

and $\zeta_{s,i}$ and $\eta_{s,i}$ can be determined from the initial conditions, stipulating a first eigenmode displacement, and no velocity:

$$a_{s,i}(0) = \begin{cases} 1, i = 1 \\ 0, i \neq 1 \end{cases}, \quad \dot{a}_{s,i}(0) = 0, \forall i. \tag{4.5}$$

By replacing (4.4) in (4.5), one obtains the transient analytical solution for the third verification test of the FEM solver as:

$$\boldsymbol{u_{s,MS}}(t) = \boldsymbol{\phi}_{s,1}e^{-\xi_1\omega_{s,1}t}\cos\omega_{s,d,1}t, \tag{4.6}$$

where $\omega_{s,d,1} = \omega_{s,1}\sqrt{1-\xi_{s,1}^2}$, and $\xi_{s,1}$ represents the modal damping associated with the first eigenmode [81]. For this test, it was assumed that $\xi_{s,1} = 0.013$, arbitrarily taken as approximately equal to the value of the modal damping obtained experimentally by Chen and Wambsganss [1]. Hence, for the third test associated with the FEM solver, it is desired that the Newmark numerical integration scheme can match the analytical solution given by (4.6), at each node of the rod's discretization, and at every considered timestep. To conduct the test, 5 and 20 beam elements were used. Similar to the rest of the thesis, a Newmark time integration scheme with $\alpha = \frac{1}{4}$, $\gamma = \frac{1}{2}$ was imposed.

4.1.2. Results

The results for the first verification test can be visualized in Figure 4.2, where an exact matching is obtained between Abaqus and the developed FEM solver when the same inputs and types of elements are used. This ensured that the assembling of the global stiffness and forcing vectors when both bending planes are included is correct.

Next, the outcome of the second verification test of the FEM solver is presented. For this, the first four eigenmodes of the structure are used as input. This choice is justified by the fact that the FSI simulations in this work are run at 10 m/s, and under these flow conditions, the first four modes of the brass beam are the most excited [38, 41]. For the test, between 5 and 30 beam elements were employed, because based on the available literature this was expected to be the general range of beam elements that one should use for axial flow TIV FSI simulations [29, 63]. To determine the convergence of the eigenfrequencies obtained using the FEM solver module towards the analytical solutions, the relative error $\chi_{r,\omega_{s,q}}$ is used:

$$\chi_{r,\omega_{s,q}} = \frac{|\omega_{s,q,num} - \omega_{s,q}|}{\omega_{s,q}}, \tag{4.7}$$

where $\omega_{s,q}$ is computed using (4.1), and $\omega_{s,q,num}$ is the circular eigenfrequency obtained using NRG-beamFoam to compute the generalized eigenproblem (2.98). Based on this monitor, the errors obtained for the first four eigenmodes when considering different levels of structural discretization are presented in Figure 4.3.

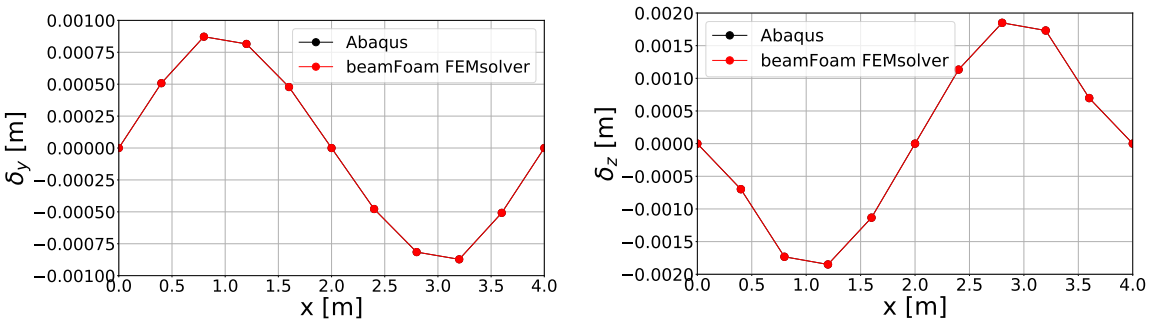


Figure 4.2: Static lateral displacements obtained for the $y-$ (left) and the $z-$ (right) direction, for the first verification test of the FEM solver

Analyzing Figure 4.3, a good match with the expected circular frequencies is visible. If using five elements to discretize the fuel element geometry, then an error of 2% can be expected for the fourth eigenfrequency, based on the black marker at the top right of Figure 4.3. On the other hand, for the same number of elements, if considering the first eigenfrequency, a relative error of less than 0.1% is found. Furthermore, another pointer that the FEM solver works correctly is that the prediction of each eigenfrequency is convergent with respect to the number of elements: looking at the errors for each of the four eigenfrequencies, the error is reduced when considering a higher number of elements. For each individual mode, the order of convergence seems to be constant, that is, the reduction in the order of the error is similar when going from 5 to 10 elements, or when going from 10 to 20. Finally, it is important to note that although Figure 4.3 can serve as an indicator of the eigenfrequency errors that can generally be expected from the NRG-beamFoam workflow, it is not clear what the dependence of the results are with regards to the different properties of the fuel element geometry considered, such as, for example, its slenderness ratio, which is significantly larger than that of the brass beam geometry that will be used in the remainder of this work. Hence, it cannot be readily concluded that one might expect the same number of errors as those in Figure 4.3 when considering the brass beam case, which will be used exclusively in the remainder of this thesis, but it is an indication that the global mass matrix is correctly implemented for the workflow.

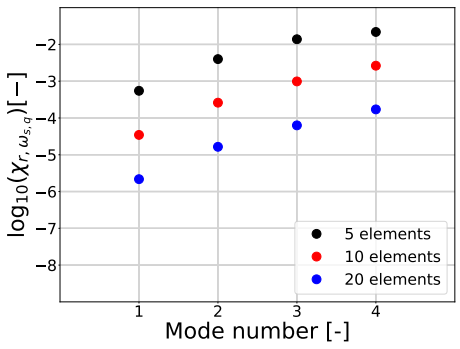


Figure 4.3: The variation in relative error of the first four eigenfrequencies between the `beamFoam::FEMsolver` module and the analytical solution, for different numbers of elements used to discretize the fuel element geometry, when considering one-plane bending

Apart from the results associated with the eigenfrequencies, another set of outputs of the second verification test of the FEM solver module is the eigenmode shapes obtained. It is of interest to observe whether these match the analytical solutions (4.2). This comparison is visually represented in Figure 4.4, where the eigenmodes were rescaled using the amplitude of the eigenmode, $A(\boldsymbol{\phi_s}$. There, it can be observed that when using 30 elements, the computed eigenmode virtually coincides with the analytical solution. The eigenmodes obtained for coarser discretizations are shown in Figure C.2, Appendix C.1. There, it can be seen that even for a low number of elements the expected eigenmode shapes are obtained, and that a clear convergence to the analytical solutions can be observed. These again indicate that the second verification test is passed.

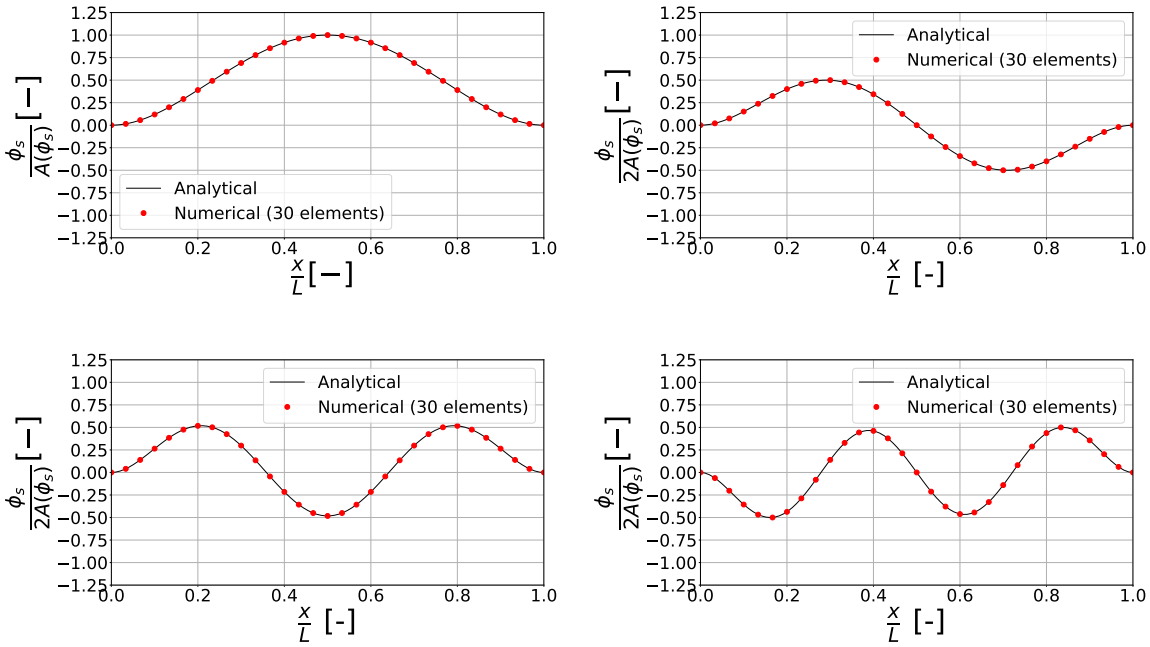


Figure 4.4: The comparison between the unitary amplitude vectors obtained analytically, and the discrete values obtained by the `beamFoam::FEMsolver` module, for 30 elements used to discretize the fuel element geometry

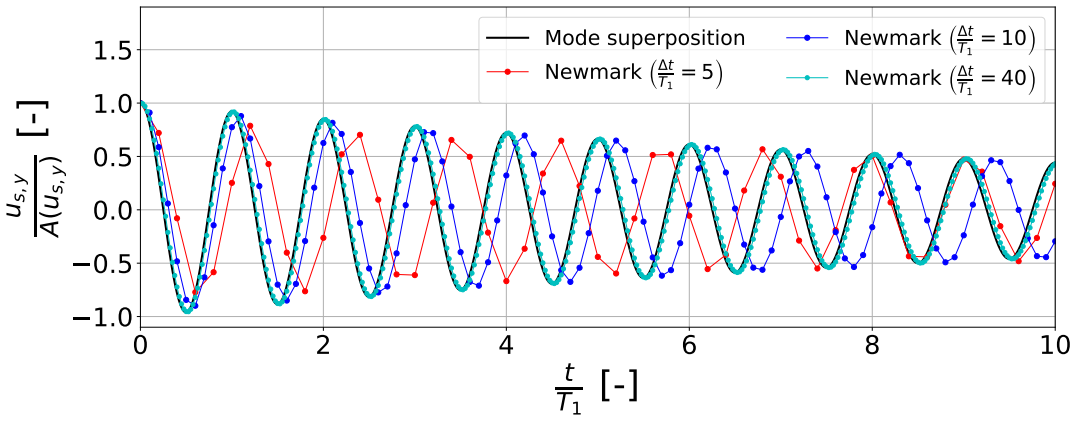


Figure 4.5: Comparison of the transient lateral midpoint displacements of the middle of the rod when using the Mode Superposition and the Newmark method for different numbers of timesteps used per period

Lastly, the results of the third verification test are presented. The transient midpoint displacement following a first eigenmode initial displacement is presented in Figure 4.5, where different numbers of timestep per period are used. For this Figure, the displacements are adimensionalized using the initial displacement, $A(u_{\mathbf{s},\,\mathrm{y}})$, and the time with the damped first period of the structure, T_1. It can be seen that with decreasing timestep size, the displacements obtained get increasingly closer to the Mode Superposition solution. Even if a small number of timesteps is used, the amplitude of the motion appears to be preserved throughout the transient. In this sense, it can be concluded that the time integration scheme used herein does not modify the energy of the vibration based on the timestep size, but only its phase shift. Indeed, this behaviour is supported by the literature [89], where it is shown that the dissipation of the vibration energy is only dependent on the value of γ in the Newmark time integration scheme, (2.95), while the phase shift depends on the chosen timestep size. Hence, the results obtained for the third verification test indicate a correct implementation of the Newmark time integration scheme.

Up to this point, the discussion of the FEM solver was only made by considering the XY bending plane. Given the cross-circular section of the fuel element geometry presents the same bending properties in all radial directions, the results for the XZ bending plane are expected to be the same as for the XY one. This is supported by the information in Appendix C.1: Figures C.1 - C.3 show that the same eigensolution is obtained for all discretization levels considered.

4.2. Load interpolator

Before describing the test cases, some preliminary observations are made. In contrast to the FEM solver, the load interpolator module directly communicates with the Fluid Solver in OpenFOAM to obtain its inputs. This implies that the individual testing of the load interpolator must be undertaken within a (pseudo-)fluid simulation. On the other hand, it is desired that the tests be kept cheap and simple, making use of only the load interpolator module, instead of the entire FSI workflow. As a consequence of these observations, the load interpolator tests consist of a single-timestep, single-iteration fluid simulation, where the structure is considered rigid, and the fluid equations are not solved. To this end, a modified version of the PIMPLE algorithm is used, where the fluid equations are not computed, and the displacements sent to the mesh motion solver for the FSI boundary are equal to 0. As was also pointed out in Section 3.5, the geometry used for the verification is that of the brass beam case.

Having presented the general framework of the verification of the load interpolator, the method of imposing the inputs for the load interpolator is presented. As can be seen in the Methodology flowchart, Figure 3.1, these consist of loads associated with the fluid mesh's boundary faces on the FSI boundary. Although the load interpolator testing should be conducted inside OpenFOAM, an exact forcing distribution on a given boundary is difficult to achieve by imposing a certain type of flow within the annular fluid volume of the brass beam case. One must therefore resort to the boundary conditions applied directly onto the surface of the brass beam. The loads obtained at the wall stem either from the local pressure forces or from the effective stresses that result from the modeled viscous and turbulent loads. The latter contributions are directly dependent on the initial configuration of the flow, making them difficult to manipulate reliably, as was already mentioned. Hence, the forcing applied onto the rod is best controlled by imposing a convenient pressure boundary condition on the outer surface of the brass beam. In practice, to avoid the uncertainties introduced by the flow, imposing a fixed value of the pressure as a Dirichlet boundary condition is most desirable. Within the OpenFOAM framework, a fixed pressure can be defined separately for each patch that the user defines in the preprocessing phase of the simulation before the mesh is generated.

Next, the characteristics required for the forcing inputs are discussed. One of the key qualitative characteristics of axial-flow TIV is that the forcing field imposed onto the structure is non-uniform in the axial direction. On the other hand, it was established that the forcing is imposed by prescribing the pressure values on the patches of the FSI boundary. Furthermore, the load interpolator has the role of reducing the 3D loads over the FSI boundary into a set of 1D distributed loads, by summing the forcing over the structure's cross-section at a given axial location. Given the circular cross-section of the brass beam case, imposing the same pressure value on the entire circumference would lead to a zero net load when mapping to the 1D structural domain. It is therefore important for verification purposes that the pressure field is not constant in the circumferential direction of FSI boundary. In this context, to impose an axially varying pressure field over the rod, one would

therefore have to simultaneously split its surface into multiple patches at different axial and circumferential positions. This would generate a large number of patches, and hence a more complicated setup of the problem, given that for each patch, all of the other boundary conditions of the problem will have to be imposed. A workaround to this approach is found, and it is presented in the following Subsection.

4.2.1. Tests description

In Chapter 3, it was pointed out that the working of the load interpolator can be conceptualized into three separate tasks: the computation of the 1D distributed loads for each segment, the coordinate transformation between the OpenFOAM and the beam reference system, and the integration of the 1D distributed loads over each beam element to obtain the associated nodal forces. It is therefore desired that in doing the individual verification of the load interpolator, all three of these features are shown to work correctly. In the previous Subsection, it was remarked that this is preferably done for an axially- and circumferentially- varying pressure field, and that within the framework of CFD simulations in OpenFOAM, defining such a varying pressure field using the boundary conditions is cumbersome. In this context, it is proposed that the verification of the load interpolator be undertaken in two separate tests, as will be described below. The two can be seen as complementary, and hence their requirements are satisfactory for proving the basic functioning of the load interpolator module.

The first test consists of imposing a non-zero pressure field over only a portion of the brass beam's circumference, while keeping it constant in the axial direction for the entire rod. In doing so, a nonzero distributed load is obtained at all axial positions of the rod. For simplicity, it is assumed that the non-zero load is applied over a quarter of the rod's circumference, such that at any point where the non-zero pressure is imposed the the generated stresses point in the positive direction of the lateral directions of the beam reference system. A visualization of the pressure field imposed onto the rod can be seen on the left of Figure 4.6. In the following, the constant, nonzero pressure force applied onto the rod will be denoted $p_{\frac{1}{4}}$, as used in the right of the same Figure.

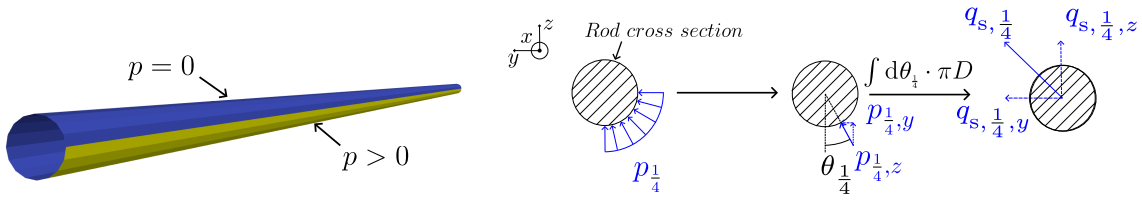


Figure 4.6: Visual representation of the first load interpolator verification case proposed; left: the pressure field imposed onto the rod; right: the equivalent distributed loads for each bending plane, $q_{\mathrm{s},\frac{1}{4},y}$ and $q_{\mathrm{s},\frac{1}{4},z}$

Compared to the unified approach initially proposed, this setup offers some advantages. For one, only two patches need to be defined for the brass beam's outer surface, one where the pressure is zero, and one where it isn't. Furthermore, one knows the analytical values of the distributed loads that are expected to be obtained for each segment, and each of the two bending planes. Using the notations on the right of Figure 4.6, it can be seen that based on the defined azimuthal angle $\theta_{\frac{1}{4}}$ the local $y-$ and $z-$ components of the stresses are equal to:

$$p_{\frac{1}{4},y} = p_{\frac{1}{4}} \sin\theta_{\frac{1}{4}}, \tag{4.8}$$

$$p_{\frac{1}{4},z} = p_{\frac{1}{4}} \cos\theta_{\frac{1}{4}}, \tag{4.9}$$

and thus the average stresses in each direction can be obtained through using:

$$\overline{p}_{\frac{1}{4},y} = \frac{\int_0^{\frac{\pi}{2}} p_{\frac{1}{4}} \sin\theta_{\frac{1}{4}} \, \mathrm{d}\theta_{\frac{1}{4}}}{\int_0^{2\pi} \mathrm{d}\theta_{\frac{1}{4}}} = \frac{1}{2\pi} p_{\frac{1}{4}} \tag{4.10}$$

$$\overline{p}_{\frac{1}{4},z} = \frac{\int_0^{\frac{\pi}{2}} p_{\frac{1}{4}} \cos\theta_{\frac{1}{4}} \, \mathrm{d}\theta_{\frac{1}{4}}}{\int_0^{2\pi} \mathrm{d}\theta_{\frac{1}{4}}} = \frac{1}{2\pi} p_{\frac{1}{4}} \tag{4.11}$$

In equations (4.10) and (4.11), $\overline{p}_{\frac{1}{4},y}$ and $\overline{p}_{\frac{1}{4},z}$ represent *stresses*, and therefore have units $\frac{\mathrm{N}}{\mathrm{m}^2}$. To obtain the distributed loads in each plane, measured in $\frac{\mathrm{N}}{\mathrm{m}}$, use is made of a load interpolator assumption made in Subsection 3.4.1.1, namely that the stresses are evenly distributed around the rod's circumference. Doing so, the distributed loads in each plane are obtained by multiplying the average stresses with the rod's circumference:

$$q_{\mathrm{s},\frac{1}{4},y} = \overline{p}_{\frac{1}{4},y} \cdot \pi D = \frac{D}{2} p_{\frac{1}{4}}, \tag{4.12}$$

$$q_{\mathrm{s},\frac{1}{4},z} = \overline{p}_{\frac{1}{4},z} \cdot \pi D = \frac{D}{2} p_{\frac{1}{4}}, \tag{4.13}$$

and therefore:

$$\boxed{q_{\mathrm{s},\frac{1}{4},y} = q_{\mathrm{s},\frac{1}{4},z} = \frac{D}{2} p_{\frac{1}{4}}}. \tag{4.14}$$

Hence, the objective of this first test is to observe whether in computing equation (3.13), the load interpolator module obtains the analytical solution (4.14). Should this test be passed, two key capabilities are proven: the capacity to correctly obtain the localized fluid forcing for each finite volume cell face on the FSI boundary, as well as the correct distribution of the fluid faces to each segment using the associated connectivity matrix. Additionally, since for the brass beam case, the $x-$ direction of the OpenFOAM reference system does not correspond to the axial direction of the rod, the coordinate transformation (3.18) is also verified. It is, however, important to point out a limiting additional assumption of the test: since it is difficult to prescribe stresses due to viscosity and turbulence, the contribution of $\boldsymbol{\tau}_{\mathbf{f},\mathrm{ik}}$ to (3.13) is momentarily neglected, and will be tested in Section 5.4, in the context of a transient FSI simulation.

Based on the above observations regarding the first test, it can be deduced that it proves the first two of the three main functionalities of load interpolator, that is, the computation of the distributed loads for each segment, as well as their transformation to the beam reference system. However, it is also important to assess whether given a set of distributed loads, the correct nodal forces are obtained using the integration formula (3.19). If one simply uses the first verification test to compute the distributed loads, the correct nodal equivalence would only be tested for uniform load distributions, which as already mentioned is not representative of axial-flow TIV. To improve trust in the module and to avoid over-complication in setting up the case, it is proposed that non-uniform 1D load distributions are *prescribed* for each of the segments, rather than transferred from the first test case. This approach is justified by the following: after passing the first verification test, it can be assumed that the load interpolator is capable of obtaining the correct values of the 1D distributed loads at its different axial locations. Hence, one can test the rest of the load interpolator's functionalities independently, using artificial intermediate inputs.

Next, the choice of distributed loads prescribed at the segments' nodes is discussed. Similar to the first verification test, these are chosen such that analytical solutions are known, and can be used to assess the model's accuracy. In the literature, the statically equivalent nodal loads can be found for simple internal load cases, such as for a concentrated load applied at the midpoint of the structure, or for a triangular distributed load [80, 84]. Part of these analytical solutions can be found in tabulated form in Figure C.4, Appendix C. Such results can be easily retrieved for the nodal loads in both the XY ($F_{1,y}, F_{2,y}, M_{1,z}, M_{2,z}$), as well as the XZ ($F_{1,y}, F_{2,y}, M_{1,z}, M_{2,z}$) bending plane.

For the verification test herein, uniform, linear, and parabolic distributed loads are imposed onto the segments' nodes. The shape of the distributed loads, along with the expected nodal loads in the XY bending plane, are given in Table 4.1. For the XZ bending plane, one considers the same value of the nodal forces, and the same absolute values of the moments, *but with a negative sign*, given the different conventions for the two bending planes, as was already described in Subsection 2.3.5. The verification test is undertaken separately for each beam element, as visualized for a parabolic load in Figure 4.7. For the simple load distributions being considered here, the local values for each node of the segments can be easily obtained based on their known positions within the local frame of reference of each element. For example, for a parabolic load, the local value of the 1D distributed load at a normalized position $\frac{x}{L_{\mathrm{el}}}$ of an arbitrary segment node inside the element is:

$$q_{\mathrm{s}}\left(\frac{x_{\mathrm{el}}}{L_{\mathrm{el}}}\right) = -4w\left(\frac{x}{L_{\mathrm{el}}} - \frac{1}{2}\right)^2 + w, \tag{4.15}$$

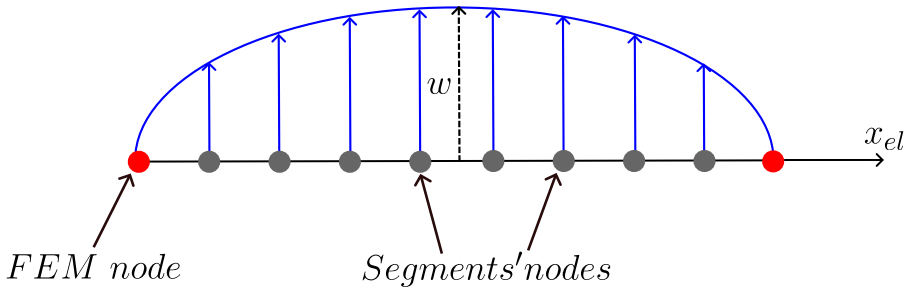


Figure 4.7: A visual representation of the second verification test for the load interpolator, where as an example a parabolic load distribution over the element is sketched

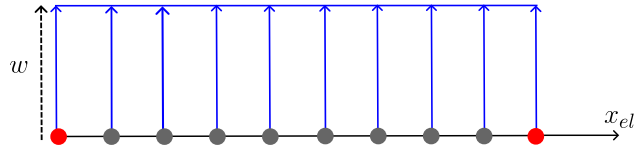
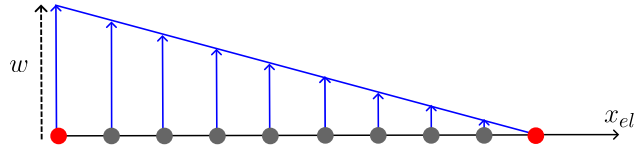
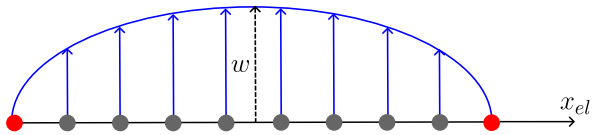
Considered distribution	$F_{1,y}$	$F_{2,y}$	$M_{1,z}$	$M_{2,z}$
	$\frac{wL_{el}}{2}$	$\frac{wL_{el}}{2}$	$\frac{wL_{el}^2}{12}$	$-\frac{wL_{el}^2}{12}$
	$\frac{7wL_{el}}{20}$	$\frac{3wL_{el}}{20}$	$\frac{wL_{el}^2}{20}$	$-\frac{wL_{el}^2}{30}$
	$\frac{wL_{el}}{3}$	$\frac{wL_{el}}{3}$	$\frac{wL_{el}^2}{15}$	$-\frac{wL_{el}^2}{15}$

Table 4.1: The load distributions considered for the second verificaton test of the load interpolator

where w represents the maximum value of the parabola, see Figure 4.7. Similarly, the values of the shape functions required to numerically compute equations (3.5)-(3.12) are also known for a given axial position in the local frame of reference of the system, see equations (2.73)-(2.76), and (2.84)-(2.87). The objective of the test is therefore to see whether the nodal forces and moments obtained by the load interpolator replicate the values seen in Table 4.1 for the different distributed loads considered, for both of the bending planes in which the structural displacements are computed.

4.2.2. Results

The results for the first verification test are presented. This was conducted at multiple levels of discretization for the fluid and the structure, to quantify the impact of both discretizations on the module's behavior. Within equation (4.14), the rod diameter of the brass beam was used, along with an arbitrarily chosen pressure force $p_{\frac{1}{4}} = 10^5 \ \frac{\mathrm{N}}{\mathrm{m}^2}$, obtaining:

$$q_{\mathrm{s},\frac{1}{4}} = 635 \ \frac{\mathrm{N}}{\mathrm{m}}. \tag{4.16}$$

As previously mentioned, it is of interest to determine how close the predictions of the load interpolator module are to this result. In the following, these will be denoted as $q_{\mathrm{s},\frac{1}{4},\mathrm{OF}}$. A unique distributed load is used as the module's output, since it was found to offer the same distributed load values in both bending planes, that is, $q_{\mathrm{s},\frac{1}{4},\mathrm{OF}} = q_{\mathrm{s},\frac{1}{4},\mathrm{OF},y} = q_{\mathrm{s},\frac{1}{4},\mathrm{OF},z}$. For quantifying the accuracy, the relative error $\chi_{\mathrm{rel},\frac{1}{4}}$ will be used:

$$\chi_{\text{rel},\frac{1}{4}}\,[\%] = \frac{q_{\text{s},\frac{1}{4}} - q_{\text{s},\frac{1}{4},\text{OF}}}{q_{\text{s},\frac{1}{4}}} \cdot 100 \tag{4.17}$$

The values of $\chi_{\text{rel},\frac{1}{4}}$ for the different considered discretizations of the brass beam case are presented in Table 4.2, where the number of fluid elements for each case is successively provided in the radial, circumferential, and axial direction. Since the verification test occurs at the outer surface of the brass beam, and the fluid equations are not solved within the test, the circumferential and axial discretizations are of most interest. For all cells on the rod's boundary, all elements have the same sizes, that is, the considered fluid meshes are locally uniform. It is worth noting that for any given configuration in Table 4.2, the same value of the distributed load was obtained for all of the segments. Additionally, the finest mesh discretization corresponds to the one used by van den Bos [38] in his simulations of the brass beam case using AniPFM. This is relevant for this work because the ultimate goal for the model developed herein is to function in combination with the AniPFM for axial-flow TIV-FSI simulations.

Fluid Discretization	Structural Discretization		
# of fluid nodes	# of elements	# of segments	$\chi_{\text{rel},\frac{1}{4}}$ [%]
(10 × **8** × 120)	20	3	0.647
		6	
(20 × **8** × 160)	10	3	
		6	
	5	3	
(50 × 40 × 400)	5	3	0.105
	10	3	
	20	3	

Table 4.2: The results for the first load interpolator test, conducted for different fluid and structural discretizations; for the first five tests, the same value of $\chi_{\text{rel},\frac{1}{4}}$ is obtained; the only parameter that remains constant for all five is the number of circumferential elements, bolded in blue; this suggests a link between $\chi_{\text{rel},\frac{1}{4}}$ and the circumferential discretization of the fluid mesh

Based on the obtained results, a few observations can be made. Firstly, the relative error does not depend on the number of beam elements, nor the number of segments considered for each. This result can be explained by the fact that the distributed loads are obtained as a summation of the cell forces assigned to each segment, *that are subsequently weighed with the corresponding summation of the cell face areas*, as stipulated in equation (3.13). This approach ensures that the load mapping is consistent for each segment.

Secondly, the accuracy in mapping the imposed uniform loads is within 1%, even when just eight fluid elements are used to discretize the circumference of the rod. In practice, this would be too coarse for any high-fidelity fluid simulation. More relevant in this regard is the finest computational grid, which was previously used for AniPFM simulations. There, it can be seen that the obtained error is approximately 0.1%. This is significantly smaller than the current uncertainty of the AniPFM module [41], therefore suggesting that the execution of the first two tasks of the load interpolation technique is not going to significantly reduce the accuracy of the proposed FSI workflow. In fact, it appears as if the small difference may be caused by the difference between the discretized surface over which the pressure $p_{\frac{1}{4}}$ is applied within OpenFOAM, and the idealized circular surface that was used to compute (4.14). In reality, in OpenFOAM, this circular surface is obtained using straight cell faces onto which the constant value of $p_{\frac{1}{4}}$ is applied. This hypothesis is additionally supported by the fact that for the first five tests in Table 4.2, all of which gave the same relative error, the only discretization parameter that remained constant was the number of fluid cells in the circumferential direction, as illustrated in bold blue. Additionally, the number of circumferential elements is also the main parameter that changes between the first five and the last three tests, the two blocks presenting different relative errors amongst each other, but constant within each of them. As expected based on this proposed argument, the relative error reduces as the circumferential discretization becomes finer.

Next, the results for the second verification case of the load interpolator are discussed. Those consisted of applying uniform, linear, and parabolic loads, using varying numbers of segments (between 3 and 24), to observe the convergence rate of the obtained nodal loads with respect to the load interpolator discretization. As a monitor, similar to the first load interpolator test, a relative error between the analytical values in Table 4.1 and those obtained in OpenFOAM are used. However, those will not be graphically presented using percentages, to avoid overcomplicating the interpretation of the logarithmic scale results.

First, the performance observed for the uniform load distribution is described. **For all levels of discretizations considered, the load interpolator obtained the exact values of the nodal forces and moments associated with the uniform load, as seen in Table 4.1**. This result was expected: if the load applied onto the element is uniform, then according to the static equivalence principle the nodal forces and moments are obtained by integrating a second- (for the moments), or a third- (for the forces) order polynomial over the element's length, as seen in equations (3.5)-(3.12). On the other hand, the Simpson 3/8 rule is based on approximating the integrand as a third-order polynomial, and using this to obtain a simplified quadrature rule. Hence, it can be expected that the Simpson 3/8 can integrate exactly a polynomial of the third order or lower.

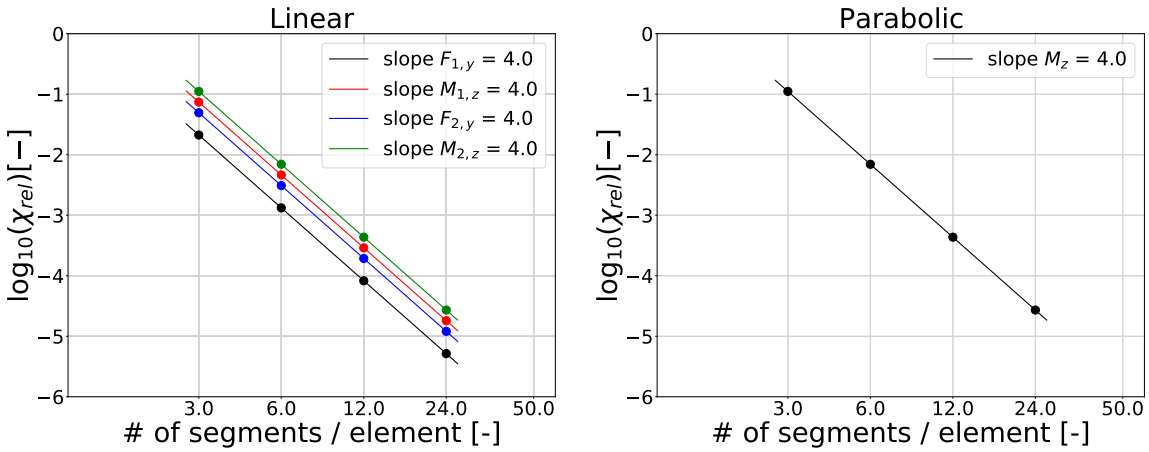


Figure 4.8: The logarithmic relative error obtained for the second verification test of the load interpolator, when considering a linear (left), and a parabolic (right) distributed load over the beam element

Next, the relative errors obtained when considering the linear and the parabolic distributed loads are presented in Figure 4.8. In particular for the parabolic load, given the symmetry with respect to the element's midpoint, the absolute values of the forces and moments are equal for the two nodes, as can also be seen in the last row of Table 4.1, $F_{1,y} = F_{2,y} = F_y$, and $M_{1,z} = -M_{2,z} = M_z$. This qualitative result is also maintained by the numerical computations of the load interpolator module, and hence in the following the discussion of the forces and moments for the parabolic load is made common for the two nodes.

Analyzing Figure 4.8, it can be seen that for all of the concentrated loads considered, the slope is constant, and equal to 4. This result is expected: the Simpson 3/8 rule has this exact convergence rate [135]. Furthermore, for all cases, when considering three segments, a relative error between 2% and 10% is obtained. This is one order of magnitude larger than the errors obtained within the first verification test, Table 4.2. When using six and twelve elements, however, this already reduces to below 1% and 0.1%, respectively, for all of the concentrated loads. Thus, it can be concluded that based on the requirement of obtaining nodal forces and moments that wouldn't increase the uncertainty of the FSI computations, one should use between six and twelve segments. Based on the left plot of Figure 4.8, if one compares $F_{1,y}$ to $M_{1,z}$, as well as $F_{2,y}$ to $M_{2,z}$, the accuracy of the load interpolator appears to be better when considering the lateral forces, rather than when computing the bending moments. This result is encouraging, since for the considered application it is expected that the deformations due to the lateral forces are significantly larger than those due to the bending rotation, given the high bending stiffness of nuclear fuel elements.

On the right of Figure 4.8, only the convergence of the moments is presented because **similar to the uniform distribution, exact values of the nodal forces were also obtained for the parabolic distribution when considering six elements or more**. Though this result is somewhat unexpected, it can be explained by the fact that the truncation error of the Simpson 3/8 rule is directly proportional to the fourth derivative of a point within the integration domain [135]. It can be shown that the fourth derivative of the integrand for the nodal forces when using a parabolic load reaches 0 for $\frac{x_{el}}{L_{el}} = \frac{1}{2}$, hence suggesting that the result is theoretically possible.

4.3. Displacement interpolator

The role of the displacement interpolator module is to map the underrelaxed displacements of the structural nodes to the fluid cells on the FSI boundary. To do so, the displacement interpolator makes use of equations (3.20)-(3.25), as presented in the Methodology part of this report. In practice, this requires the computation of different specific intermediate variables for the fluid nodes on the FSI boundary, such as their connectivity matrix with the structural elements, as well as their current and initial positions with respect to the beam axis, $\boldsymbol{r}_{\mathbf{beam}}$ and $\boldsymbol{r}_{\mathbf{beam,0}}$. Their correct calculation must be verified before any complete transient FSI simulation is run. At the same time, the displacement interpolator module is also the outer interface of the NRG-beamFoam workflow with the rest of the OpenFOAM architecture. It is therefore of interest to also see whether the correct displacements of the FSI boundary are sent to the mesh motion solver. Finally, provided that the module functions as expected, another objective of this Section is to determine the accuracy that one might expect from the displacement interpolator, for different numbers of structural elements relative to the fluid grid. This information can then be used by future users of the NRG-beamFoam workflow to obtain the desired level of accuracy within their simulations. In contrast to the other two main modules of the NRG-beamFoam workflow (the FEM solver and the load interpolator), the verification of the displacement interpolator is conducted using only one quantitative test. This is described in the following Subsection.

4.3.1. Test description

Similar to the tests proposed for the FEM solver and the load interpolator, it is desired that verification of the displacement interpolator occurs as the module is isolated from the rest of the FSI workflow. Because of this, the computation of the fluid equations was temporarily deactivated. Furthermore, to avoid any other potential biases, no methods of the `LoadInterpolator` or the `FEMsolver` classes were called during the mesh deformation. Since part of the displacement interpolator's functioning is tied to its correct link with the rest of the OpenFOAM framework, it is required that the verification is undertaken within OpenFOAM. Based on these assumptions, there can be no dynamic coupling between the fluid and the structure, and thus only one timestep can be computed. In summary, the verification test consists of a single-timestep simulation in OpenFOAM, where within the PIMPLE algorithm only the `DisplacementInterpolator` class will be called. Next, the data used within the verification case is discussed. Similar to the load interpolator verification, an artificial set of inputs is fed to the displacement interpolator, for which the analytical solutions of the outputs can be obtained and used as a benchmark. Given that most of the energy is expected to be transferred in narrow frequency bands around the eigenfrequencies of the structure [83], it can be supposed that the structural displacement can be reasonably described using a linear combination of the eigenmodes of the structure. Indeed, this is also the philosophy that led to the eigenmode-based ROMs presented in Subsections 2.4.3.2-2.4.3.4. Some of these ROM methods will be used later in this thesis, and it is therefore of additional interest to see the performance of the displacement interpolator in mapping the eigenmodes of the structure. Hence, as inputs to the verification test of the displacement interpolator, the eigenmodes obtained for the coarse structural grid consisting of beam elements will be used. The role of the displacement interpolator is to then map the said eigenmodes to each of the fluid nodes on the FSI boundary, where the fluid discretization is expected to generally be significantly finer, given the different numerical methodologies used to solve the two domains.

To conduct the verification case, the brass beam geometry is used. Given the practical objective of determining the accuracy of the displacement interpolator for future reference, the AniPFM discretization of van den Bos [38] is used, with 400 elements used in the axial, 50 in the radial, and 40 in the circumferential directions. This mesh was also employed in the first verification test of the load interpolator case, for similar practical purposes. To obtain the expected analytical solution at the fluid nodes, the eigenmodes were obtained with a matching structural grid. For the fluid mesh used herein, the considered cells have constant lengths in the axial direction, and since this assumption is also used by the structural elements, to obtain a matching struc-

tural grid for the analytical solution, the only requirement is to also use 400 beam elements. For this test, both bending planes are active, to observe the displacement interpolator's performance in both considered directions.

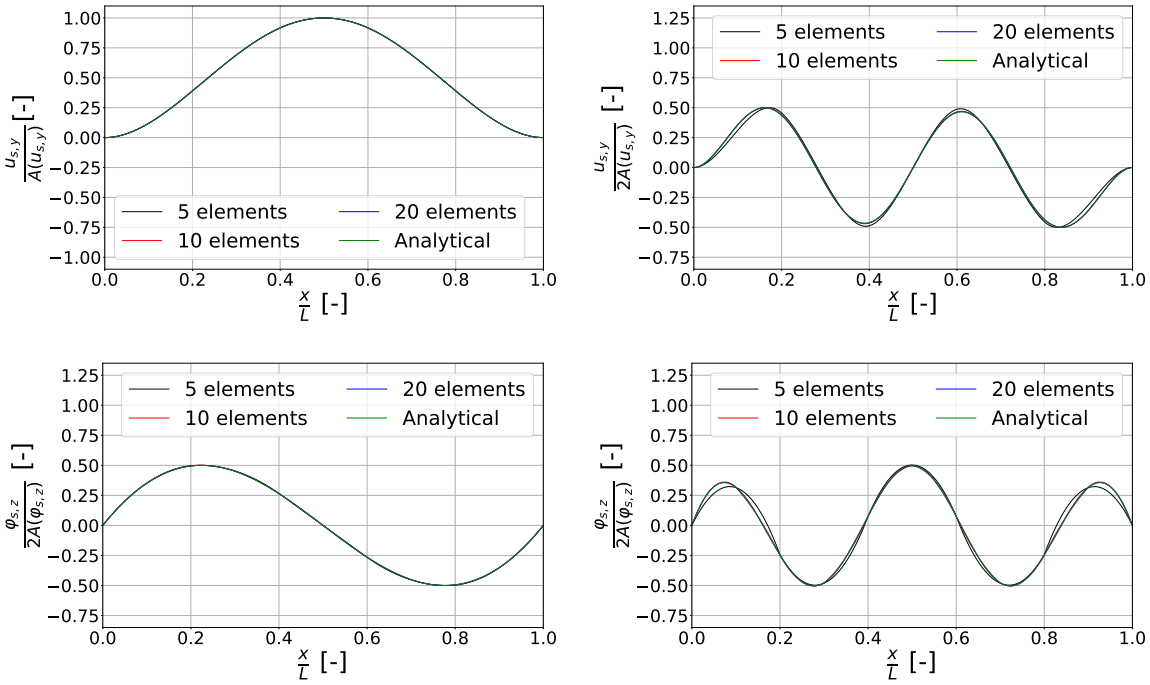


Figure 4.9: The axial distributions of the unitary amplitude $u_{s,y}$ (top) and $\varphi_{s,z}$ (bottom) degrees of freedom obtained for the first (left) and the fourth (right) eigenmode, at different levels of structural discretizations, compared with the analytical solution

4.3.2. Results

To verify the displacement interpolator's behaviour, the first four eigenmodes of the structure are used as input. For this, the same motivation as provided for the second verification test of the FEM solver stands: for the FSI simulations that will be considered later in this work, these are the most excited modes. For the structural discretization, between five and 20 elements will be used. The results presented herein were obtained starting with the displacements imposed onto all of the fluid nodes on the FSI boundary, along with their corresponding axial position in the global beam reference system. Based on the proposed beam element structural model, it was expected that all fluid nodes located at a certain axial position would have the same values of the displacements and rotations. This was indeed confirmed by processing the data from the entire outer surface of the rod. Consequently, the fluid node deformations can be uniquely represented along the 1D beam axis. Hence, for the plots in this Subsection, axial distributions are presented, where the duplicates at the same axial position have been already filtered out.

First, a visual representation of the displacements of the fluid nodes is provided. For illustration purposes, the obtained $u_{s,y}$ and $\varphi_{s,z}$ displacements for the first and the fourth eigenmode are presented in Figure 4.9. There, good matching between the expected solution and the one obtained using a limited number of elements is seen. Furthermore, for both translations and rotations, the fidelity of the displacement interpolator appears to be better for the first eigenmode than for the last. This result is expected, given that the higher the eigenmode is, the less satisfying the second (for the rotations) and third (for the displacements) order polynomial approximations (3.20) and (3.22) within an element are for the large local variations of the actual eigenmode computed on the matching structural grid. In Appendix C.3, Figures C.5-C.8, one can further analyze the spatial distributions of the first four eigenmodes obtained for all degrees of freedom considered by the structural model, that is, $u_{s,y}, u_{s,z}, \varphi_{s,y}$, and $\varphi_{s,z}$. The results there support the two claims made in this paragraph, while at the same time proving that the behaviour of the displacement interpolator is the same in both bending planes, including the difference in sign between $\varphi_{s,y}$, and $\varphi_{s,z}$ for similar distributions in

$u_{s,y}$ and $u_{s,z}$, as suggested by the kinematic relations (2.62) and (2.82) of the proposed beam element model.

Having qualitatively verified that reasonable fluid node displacements are prescribed to the FSI boundary, it is desired to quantify the module's accuracy. At this point of the NRG-beamFoam's workflow development, it is of interest to see the general accuracy across the entire FSI boundary. Thus, to assess the convergence, the relative error is computed using the norms of the eigenmode displacements obtained for the matching grid solution, $\boldsymbol{\phi}_{\mathbf{s,match}}$, and the (axially unique) output of the displacement interpolator based on the inputs generated by the eigenmodes sampled for a coarse structural grid, $\boldsymbol{\phi}_{\mathbf{s,coarse}}$:

$$\chi_{\text{rel}} = \frac{\left\| \boldsymbol{\phi}_{\mathbf{s,match}} - \boldsymbol{\phi}_{\mathbf{s,coarse}} \right\|}{\left\| \boldsymbol{\phi}_{\mathbf{s,match}} \right\|}, \tag{4.18}$$

where the vectors $\boldsymbol{\phi})_{\mathbf{s}}$ can refer to any of the nodal displacements $u_{s,y}, u_{s,z}, \varphi_{s,y}, \varphi_{s,z}$, for any of the first four eigenmodes. Based on this criterion, the convergence of the displacement interpolator module is shown for the first four eigenmodes of $u_{s,y}$ and of $\varphi_{s,z}$, on the left and the right of Figure 4.10. For extrapolation purposes, the results are presented with respect to the ratio between the number of structural and fluid nodes. The analogous results for the $u_{s,z}$ and the $\varphi_{s,y}$ degrees of freedom are presented in Appendix C.3, Figure C.9, where it can be observed that the results are identical to those obtained for $u_{s,y}$ and $\varphi_{s,z}$, respectively. This indicates that the displacement interpolator works equivalently for the XZ bending plane and that the observations made in the following apply to both bending directions.

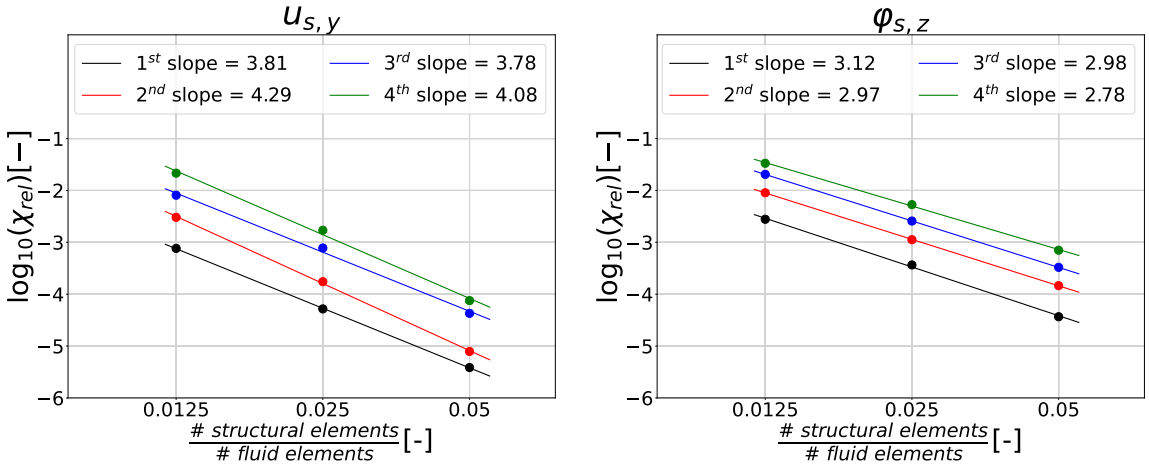


Figure 4.10: The logarithmic convergence of the displacement interpolator relative error defined in (4.18) for the $u_{s,y}$ (left) and the $\varphi_{s,z}$ (right) degrees of freedom, at varying levels of structural discretization

Based on Figure 4.10, a few conclusions can be drawn. First, for a given eigenmode number, the translations are more accurately mapped than the rotations are. What's more, the order of convergence for $u_{s,y}$ is approximately equal to 4, while for $\varphi_{s,z}$ it is roughly 3. This difference in the order of accuracy can be attributed to the fact that the translations are interpolated to the inner locations of the beam elements using a third-order polynomial, as shown in (3.20), while the rotations using a second-order one, (3.22). Secondly, as expected based on the comments made regarding Figure 4.9, for a given discretization level the error monotonously increases with the eigenmode number being considered.

Next, the different convergence rates obtained for each eigenmode mapped by the displacement interpolator are discussed. In contrast to the load interpolator test in Figure 4.8, the slopes obtained for the displacement interpolator are not constant. Furthermore, the variations in the slope values are larger for the translations than for the rotations. In particular, the displacement interpolation of $u_{s,y}$ appears to converge faster for even eigenmodes. A possible explanation for these differences can be the changing sampling locations of the eigenmodes when considering different numbers of structural nodes. As a simple example, if one uses an odd number of elements, the midpoint of the rod is not included among the structural nodes, and the peak of the first eigenmode for the translations $u_{s,y}$ or $u_{s,z}$ is not captured. This is visible, for example, by considering the

displacements obtained for the first eigenmode of $u_{s,y}$ with 5 elements, on the top left of Figure C.3. Hence, based on equation (3.20), the displacement interpolator wouldn't be able to reconstruct the peak, leading to a higher value of χ_{rel} according to (4.18). A similar argument could also be made for the higher eigenmodes, where the coarse structural grid samples the location on the left and the right of the local peaks of each eigenmode, without being able to reproduce the local amplitude.

The accuracy that one might expect from the displacement interpolator module is discussed next. The performance in terms of the translations $u_{s,y}$ is analyzed first, given that these are more relevant for fuel elements in axial-flow TIV. The results in Figure 4.10 show that when considering 80 times fewer structural elements than fluid ones, an average error of less than 0.1% can be expected for the translations mapped at the fluid nodes. For the displacements of the fourth eigenmode, this increases to approximately 2%. On the other hand, when increasing the discretization ratio to one structural element for every 20^{th} fluid one, this error for the fourth eigenmode translations already reduces to an average of below 0.1%. For the AniPFM discretization used by van den Bos [38], the two example cases correspond to using 5 and 20 beam elements, respectively. In practice, the computational difference in using the latter over the former is quite small. It can therefore be concluded that the displacement interpolator is generally able to accurately map all of the relevant eigenmodes of the structure to the fluid nodes while maintaining a reasonable computational cost. Of course, apart from a low average value of the error, it is also required that the mapping is locally accurate, that is, the module shouldn't predict large variations from the analytical solution at any of the fluid nodes. Due to time constraints, and the fact that the pointwise convergence is better assessed within the context of displacements generated by TIV forcing, this aspect was not quantified within this verification test.

Finally, the accuracy in determining the rotations $\varphi_{s,z}$ is briefly discussed. For the considered application, those are of secondary importance, given their relatively small effect on the translations imposed onto the fluid nodes, and the subsequent low amplitude of the TIV. Nevertheless, their accuracy is important to ensure the smoothness of the FSI boundary displacements, which can influence the stability of the numerical problem. For the coarsest considered structural discretization, the average relative errors are between 0.28% and 3.3%, for the mapping of the first and the fourth eigenmode, respectively. For the highest number of beam elements, the average error drops to below 0.1%. Though these variations appear to be quite small, whether this level of accuracy of the rotations is satisfying is best understood when analyzing the performance and the stability of the workflow for complete FSI simulations.

5

Verification of NRG-beamFoam using FSI simulations

The results in Chapter 4 attest to the correct basic functioning of the main modules within the proposed workflow. However, all of the tests presented therein present a major limitation: they have only been conducted for the isolated modules, and most have also only been verified statically. Hence, it is still not clear whether the transfer of information between the different modules is correctly carried out dynamically. Furthermore, in Chapter 4 only the main structural solver and the mapping routines between the fluid and the structure have been tested. However, to efficiently solve the coupling between the two domains at each timestep the proposed subiteration algorithm must also be implemented and tested.

Once the subiteration algorithm is implemented, a functional FSI workflow is obtained. It is then the target for the remainder of this Chapter to assess its accuracy and computational costs when compared to the previous NRG-FSIFOAM workflow. Both of these objectives require that fluid and structural simulations be run in a coupled manner. Given that NRG-beamFoam is still in its testing phase, the simulations should be made as computationally cheap as possible, to enable quick analyses and debugging. Since the computation of the fluid solution was found to be the biggest contributor to the computational costs of van den Bos' FSI simulations using AniPFM [38], the cheapest fluid model available is initially used for coupled simulations by assuming a laminar flow. Doing so not only eliminates the need to solve the additional transport equations for the turbulent quantities, but it also permits the use of a coarser fluid grid, and therefore a smaller dimension of the fluid systems of equations that need to be solved for each FSI loop. Once the behavior of the workflow for laminar simulations was found to be satisfying, URANS simulations were also run to observe whether the workflow could correctly include the stresses associated with turbulence. Given that both of these fluid models underestimate the amplitudes of TIV [35], an initial deformation is applied to the beam to give rise to dynamic FSI coupling. For comparison purposes, the FSI simulations are run using both the newly implemented NRG-beamFoam workflow and the NRG-FSIFOAM one that it desires to replace. In Section 5.1, a comprehensive description of the fluid and structural settings used for the laminar FSI cases is provided. Section 5.2 presents the results that confirm the correct functioning of the subiteration algorithm of the NRG-beamFoam workflow, while Section 5.3 presents the results that are obtained for the two methodologies for a laminar fluid model. Section 5.4 is dedicated to the URANS FSI cases.

5.1. Laminar FSI case setup

As mentioned at the start of this Chapter, the FSI simulations will first be run using a laminar fluid model. Though this was generally done by preserving the setup that was already laid out within the NRG-FSIFOAM workflow, a new, coarser fluid mesh could be created, thanks to the less demanding fluid model. The topic of this Section is hence to present the fluid discretization for the laminar simulation, as well as an overview of the numerical settings, since this is the first instance in this research where the fluid solver is actively used. This will be done in Subsection 5.1.1. On the other hand, this is also the first occasion when the NRG-beamFoam algorithm is used in its entirety, and hence its inputs must be defined. The inputs are presented by taking into account the earlier modelling choices, as well as the verification results obtained in Chapter 4. The settings

for the NRG-beamFoam structural solver, as well as for the deal.II used within NRG-FSIFOAM, are presented in Subsection 5.1.2. Unless specified otherwise, the same setup was used for both the NRG-beamFoam and the NRG-FSIFOAM workflows.

5.1.1. Fluid domain

First, the new fluid mesh is described. For its generation, 20 elements were used in the radial direction of the annulus, 24 around its circumference, and 120 in its axial direction. This configuration was obtained starting from the discretization grid of van den Bos [38] that he used for a URANS fluid model. Subsequently, the number of cells in all three directions was reduced such that the curvature of the surfaces is still well represented. The final grid uses 56 thousand fluid cells, reducing the count by approximately 14 times compared to the original mesh of van den Bos. A visual representation is given in Figure 5.1.

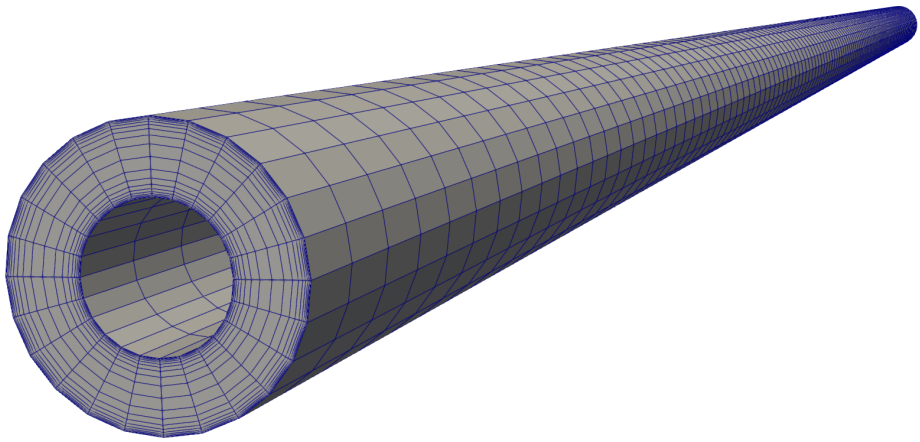


Figure 5.1: The fluid mesh used for the laminar FSI simulations in this Chapter

For the laminar FSI simulations, the discretization schemes were the same as the ones used by van den Bos [38] for URANS and PFM FSI simulations. More specifically, a second-order implicit time integration was used for the time derivatives, while for the gradient, the divergence, the Laplacian, and the cell face interpolation, linear schemes were chosen. For the wall distance, the `meshWave` method is used, wherein for each boundary and cell center, the distances to each wall are calculated, and only the minimum value is considered. The `fvSchemes` dictionary used for laminar FSI simulations can be analyzed in Appendix D, Subsection D.1.1.

The solver settings for the laminar FSI case are presented in Appendix D, Subsection D.1.2. Those were also taken from the setup of van den Bos [38] for an AniPFM fluid model, and they were found to work well also for laminar cases herein, when using both NRG-beamFoam and NRG-FSIFOAM. However, three differences in the `fvSolution` files used for the two different methodologies exist. First, for the NRG-beamFoam workflow to work in a subiterative manner, the `moveMeshOuterCorrectors` setting in the `PIMPLE` subdictionary must be set to `true`. This signals the solver that the mesh motion must be executed at each PIMPLE outer corrector, which is the essence of the strongly coupled algorithm proposed in Subsection 3.2. Furthermore, for NRG-beamFoam the number of subiterations at each timestep is controlled using the `nOuterCorrectors` entry in the same `PIMPLE` subdictionary, as opposed to the dedicated preCICE configuration file for the NRG-FSIFOAM workflow. Finally, to impose the same target residuals for both NRG-FSIFOAM and for NRG-beamFoam, one should consider that all NRG-FSIFOAM runs by van den Bos [38] were executed by using a single outer corrector for the PIMPLE algorithm. As a consequence, for the residual convergence only the `final` constraints, associated with the last outer corrector loop, were active. By contrast, NRG-beamFoam uses multiple outer corrector loops, which are controlled by both the conventional and the `final` subdictionary entries. Hence, to impose the same residuals for an arbitrary subiteration of NRG-beamFoam as those for NRG-FSIFOAM, one should use the `final` settings of NRG-FSIFOAM for the conventional entries in NRG-beamFoam. Indeed, this is what was also done for the `fvSolution` dictionary in Subsection D.1.2.

Next, the fluid properties are discussed. The benchmark experiment of Chen and Wambsganss [1] was conducted using water at atmospheric conditions. To obtain a laminar flow using water as a working fluid for the brass beam geometry, the velocity used would have to be equal to approximately 0.18 m/s. The use of this velocity is not desirable for two reasons. For one, it is not representative of axial flows in nuclear reactor cores, where velocities are usually one to two orders of magnitude higher [144]. As a consequence of this, the forces generated by the dynamic pressure of the flow over the rod would be much smaller, leading to the prevalence of the added mass effect for the FSI. This leads to the second reason for not choosing such a low velocity: the difficulty in converging simulations where the added mass effect is dominant, due to the large fluctuations induced in the fluid fields by the need to balance the mass contained within the system for incompressible solvers. To avoid the disadvantages of using water at a low velocity, it was decided to use it at 10 m/s, but keep the fluid model laminar, and thus not resolve the turbulence that naturally occurs at this axial velocity. Although this computational model is not representative of the real-life physics, this approach can be used for verification purposes, with the caveat that the effect of the turbulent stresses is not well captured. Alternatively, one could have also used a liquid that is much more viscous than water, but has a similar density, such that its flow at 10 m/s would remain laminar. However, it was feared that this would lead to a quick damping of the rod's movement, thereby hindering a long enough simulation time of the FSI phenomenon. Hence, the working fluid for the laminar FSI cases is water, with a kinematic viscosity $\nu_f = 10^{-6}\ \frac{m^2}{s}$, and a density $\rho_f = 1000\ \frac{kg}{m^3}$.

In the following paragraph, the boundary conditions imposed for the fluid are described. For clarity, a summary of the chosen boundary conditions can be seen in Table 5.1. For this work, the boundary of the brass beam case is split into four: the annular inlet and the outlet of the domain (called `inlet` and `outlet`), the surface of the rod (`inner`), and the rigid outer rod (`outer`). Typical BCs are imposed for the `inlet` and the `outlet` patches in terms of the velocity and the pressure. For the former field, a prescribed fixed value of the inflow is provided to the `inlet`, while at the `outlet` a zero-gradient BC is imposed. At the inlet, the velocity profile is assumed to be uniform across the annulus, similar to the choice of van de Bos [38] and of Freitas [41] for AniPFM simulations. In terms of the pressure, a zero operating value is prescribed at the outlet, with a zero Neumann boundary condition for the inlet. The `inner` and the `outer` patches represent solid walls, and are therefore also modeled using a zero gradient boundary condition for the pressure. For the velocity, the outer wall is assumed stationary, while for the `inner` patch the `movingWallVelocity` boundary condition is applied. This essentially represents a no-slip condition for the deforming solid wall, where the velocity is computed such that the fluxes normal to the moving wall are zero, and the tangential component of the velocity at the wall is 0. Apart from the boundary conditions for the pressure and velocity, one must also impose boundary conditions for the mesh motion solver. Here, it is assumed that for the inlet, outlet and outer surface of the annulus, the displacements are fixed and equal to 0. This is justified by the assumption that the outer rod is perfectly rigid and that the brass beam is clamped at its ends. For the lateral surface of the brass beam, it is assumed that the rod is initially stationary. However, the exact boundary condition that is imposed depends on the workflow being used: for NRG-FSIFOAM, a `fixedValue` BC is provided, similar to the rest of the boundary patches. On the other hand, for the NRG-beamFoam workflow, the newly implemented `BeamFoamDisplacement` BC must be provided, as was explained in Sections 3.2-3.3.

patch	U		p		pointDisplacement	
	type	value $\left[\frac{m}{s}\right]$	type	value $\left[\frac{m^2}{s^2}\right]$	type	value [m]
inlet	`fixedValue`	(0 0 10)	`zeroGradient`	-	`fixedValue`	(0 0 0)
outlet	`zeroGradient`	-	`fixedValue`	0	`fixedValue`	(0 0 0)
inner	`movingWallVelocity`	(0 0 0)	`zeroGradient`	-	`BeamFoamDisplacement` (for NRG-beamFoam)	-
					`fixedValue` (for NRG-FSIFOAM)	(0 0 0)
outer	`noSlip`	-	`zeroGradient`	-	`fixedValue`	(0 0 0)

Table 5.1: The boundary conditions imposed for the fluid domain for the laminar FSI simulations

Next, the initial conditions for the fluid domain are discussed. Those were obtained by simulating only the fluid flow inside the brass beam geometry, and assuming that the structure is rigid. Hence, the mesh motion solver is also deactivated. The simulation time was one second, until the residuals of the fluid equations reached low enough values that the latest solution could be considered steady state. Although this is not a necessary step for the FSI computation, that is, one can simply assume uniform pressure and velocity fields as initial conditions for the FSI problem, running a fluid-only simulation before activating the structural solver and subiterating is recommended as it leads to the convergence of the FSI coupling in a small number of subiterations from the beginning of the simulation. As for the initial conditions of the mesh motion solver, since the flow solution was obtained for a fixed mesh, no displacements are initially imposed. For comparison purposes, the same initial fields were used for both workflows.

As was mentioned at the beginning of this Chapter, a disadvantage of using a laminar or a URANS fluid model is that the TIV amplitudes are largely underestimated, and therefore to obtain a large enough forcing over the rod an initial displacement must be imposed onto the structure, after which the system is allowed to vibrate freely. For both FSI workflows considered, this is done by applying an initial uniform mechanical force over the entire span of the rod in its lateral directions, and then suddenly halting its action. For the NRG-FSIFOAM workflow, this is done using a deal.II built-in option that enables the user to apply body forces to the structure. In NRG-beamFoam, a similar functionality is controlled by the user based on the inputs to the `BeamFoamDisplacement` boundary condition. However, using NRG-beamFoam the input of the user is the force that is applied at each of the structural nodes. Given the different working principles of deal.II and NRG-beamFoam, the same exact value of the amplitude of the deformation was not immediately obtained. However, two close amplitudes were quickly imposed using the manual modification of the inputs to each models. Both displacements are small enough compared to the diameter of the annulus to assume that no additional coupling effects between the fluid and the structure arose because of the difference in initial deformation amplitude. What is more important for the numerical experiment developed herein is to observe whether the two workflows exhibit similar damping ratios and vibration frequencies. The comparison between the deformations for each methodology can be seen in Figure D.1, Appendix D.

5.1.2. Structural domain

First, the discretization for the two workflows is discussed. For NRG-FSIFOAM, the same structural grid used by van den Bos [38] was maintained, since this was already obtained at the end of an FSI mesh convergence study using a URANS fluid model with the same fluid grid as that used for this study. His findings therefore eliminate the need to conduct a standalone structural grid study for the NRG-FSIFOAM methodology as part of this research. The structural mesh used by van den Bos within NRG-FSIFOAM consists of 3D Lagrange Q FEM elements [145], where two nodes are used for each element in every direction of the Cartesian frame of reference. In terms of element count, 50 elements are used in the axial direction, and 25 in the transverse plane of the rod. On the other hand, for NRG-beamFoam, a 1D FEM element approach is used, where the user has to define the number of elements used in the axial direction of the structure and the number of segments used to map the loads from the fluid mesh to the FEM nodes. For the FSI test that is going to be conducted, where the rod is initially deformed and allowed to vibrate freely, it is expected that the rod will quickly start to vibrate in its first eigenmode, with the higher frequency excitations quickly damped by the fluid. This is also visible when considering the results of van den Bos [38] for a URANS FSI case, where a clear vibration pattern in the first eigenmode is seen. Hence, the number of beam elements and segments should be chosen so that the first eigenmode is accurately handled by the worfklow, while remaining stable during the short transient in which higher eigenmodes are also active. As was presented in Chapter 4, the accuracy in determining the deformation pattern of the rod, as well as the resulting mesh deformation of the fluid domain, is dependent on the number of elements that one uses. On the other hand, the accuracy of the concentrated loads at the FEM nodes is impacted by the number of segments being considered. Based on these criteria, a choice for the two is made in the following paragraphs.

The number of elements required is first discussed. As far as the deformation resolution of the FEM solver is concerned, the first eigenmode deformations are essentially converged when reaching 20 elements, as can be seen at the top left of Figure C.3. For the fluid grid in this laminar FSI test, using 20 beam elements leads to a ratio between the structural elements and the fluid ones that is approximately equal to 0.17. In Figure 4.10, it was shown that already for a value of 0.05, the fluid nodes' displacements imposed by the displacement interpolator are very close to the solution obtained when having a structural grid that is matching with the fluid

one. Thus, using 20 elements appears to be satisfying in terms of the requirements of both the FEM solver and the displacement interpolator modules, and will be used in the following for the laminar FSI simulation.

Next, the number of segments chosen for the laminar FSI simulations is motivated. Given the large number of elements chosen to represent the quasi-first eigenmode structural response, the variation of the loads over the elements can be modeled as linearly varying between the nodes of each element. In this context, the closest load interpolator verification case that was done is a linear variation of the distributed loads. The difference between the test and the load case in the FSI simulation is that within the verification test, the value of the distributed loads at the leftmost nodes was zero, while for the FSI verification, this will only be the case for the nodes at the clamped ends of the rod. Nevertheless, the results for the previous verification test, presented in Figure 4.8, can be used as an indicator. Hence, for a linearly varying axial load, errors of below 1% were obtained for all concentrated nodal loads when using 6 segments. The difference is particularly small for the lateral forces. This was considered a satisfying accuracy, and hence 6 segments will be used for laminar FSI simulations.

Finally, the settings regarding the time marching are presented. The two methodologies use an implicit trapezoidal scheme to advance the structural solution. For NRG-beamFoam, this corresponds to setting $\alpha = \frac{1}{4}$, $\gamma = \frac{1}{2}$ in (2.95). Within NRG-FSIFOAM, deal.II uses a θ scheme [43], where to obtain the trapezoidal scheme one must use $\theta = 0.5$. In terms of the timestep, for NRG-beamFoam, as a consequence of the chosen FSI subiteration scheme, the same value must be used for both the structure and the fluid. Hence, the value of the timestepping Δt needs to be chosen so that the dynamic accuracy obtained for both domains is satisfying. For the fluid domain, this was considered to be the case if the maximum Courant number is below 1. This requirement is mainly used as a stability criterion for time integration schemes, but for the backward implicit scheme used herein the Courant number criterion doesn't strictly apply. Nevertheless, it was still considered an indicator of the accuracy of the problem, given the linear discretization schemes used throughout, which interpolate the information on the mesh faces by only using information on the adjacent fluid cells. Furthermore, for the structural accuracy, as shown in the third FEM verification test, Figure 4.5, the number of timesteps per main period of the vibration has a clear influence on the accuracy of the displacements. Based on the findings in the third verification test of the FEM solver, it was also decided to use 100 timesteps per main period of the rod's vibration. Based on these criteria, a constant timestep $\Delta t = 3 \cdot 10^{-4}$ s is used for the laminar FSI simulations. The same timestep is also used for the NRG-FSIFOAM, for both the structural and the fluid domain, for comparison purposes.

5.2. Verification of the subiterator

Having described the settings for the laminar FSI simulation in Section 5.1, the verification of the subiteration algorithm is now presented. This constitutes the last module of NRG-beamFoam that needs to be verified before FSI simulations can be run using the beam-element FEM solver. In this context, it is of interest to see whether the module works as expected. Once this is established, it is also worthwhile to determine the approximate number of subiterations that the workflow needs, as well as the improvements of using the Aitken subiteration algorithm over using a fixed subiteration value. For all of the FSI cases in this Section, the settings in Section 5.1 are used, for a total simulation time of approximately 0.2 seconds. For the brass beam geometry, this represents between four and five vibration periods. In this Section, the transient lateral displacements are represented in adimensionalized form, where the rod diameter and the period of the vibration are used as characteristic lengths and timescales, respectively.

As was concluded in Section 2.1, given its good balance between efficiency and computational cost, for this thesis an Aitken subiteration algorithm is implemented. However, since simple benchmark tests for the Aitken algorithm were not readily found, it was decided to create one for this thesis. To this end, FSI simulations were first run using fixed values of the URF. For this subiteration method, the implementation is more straightforward than for Aitken, thereby enabling a quick implementation. What's more, for converged FSI residuals, for the same sets of settings used for the structural and the fluid solver, one should reach the same results using both subiteration methods. Thus, the results obtained with the fixed URF subiterations can be used as a benchmark for the Aitken algorithm.

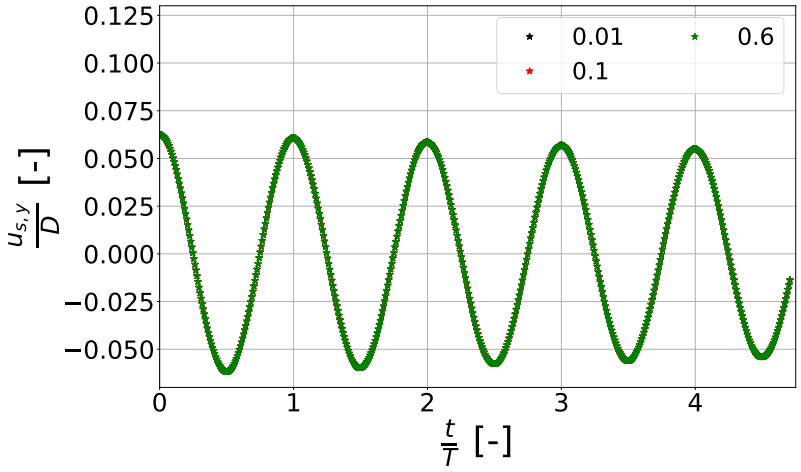


Figure 5.2: The y component of the lateral midpoint displacement of the rod for the laminar FSI case for the fixed URF verification test, equation (2.8), where different values of $\omega_{\mathrm{UR},0}$ are used for 2500 OC at each timestep for the NRG-beamFoam workflow, in combination with a laminar fluid model

To verify that the fixed URF subiteration was correctly implemented, FSI simulations with 2500 OC at each timestep were run. This number of FSI subiterations is exceedingly high, and should ensure that a converged solution is obtained for all values of the URF. To certify the convergence of the fixed URF algorithm, the $y-$component of the rod's displacement in the beam reference system was used as a monitor, and results are presented in Figure 5.2. Indeed, for the two cases, the exact same results are obtained at each timestep.

Since the fixed URF approach was found to work correctly in solving the coupling, the results in Figure 5.2 can be further used as a benchmark for the Aitken implementation. Initially, it was desired to use the same number of OC also for the verification of the Aitken subiteration algorithm. In practice, however, this led to crashes of the FSI simulations, since the residuals determined for two consecutive subiterations were eventually found to be zero. Hence, based on (2.9), the value of $\omega_{\mathrm{UR,i}}$ was obtained by diving by 0, leading to a solver error. Through trial and error, the highest number of subiterations for which the Aitken simulations remained stable was 10. This is the OC value used for the results in Figure 5.3, where the outputs of the Aitken method for the midpoint displacement are compared with the reference values in Figure 5.2. It can be seen that the same results are obtained, thereby showing that the implementation of the Aitken algorithm is also correct.

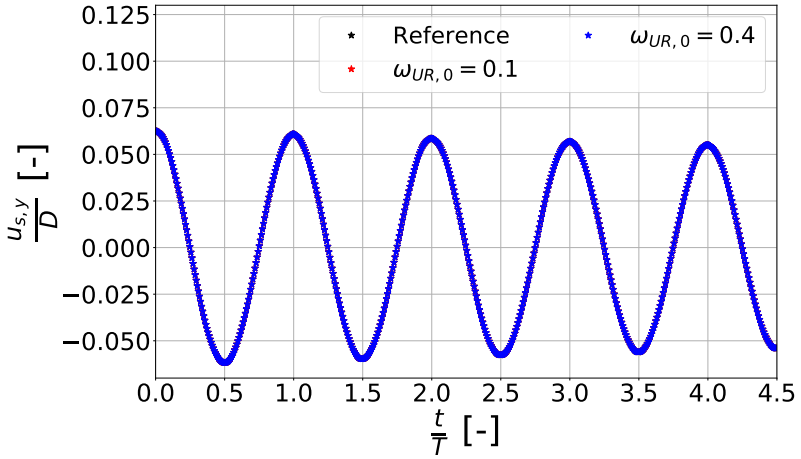


Figure 5.3: The y component of the lateral midpoint displacement of the rod for the laminar FSI case for the Aitken subiteration algorithm verification test, where two different values of the intial URF is used for 10 OC at each timestep for the NRG-beamFoam workflow, in combination with a laminar fluid model

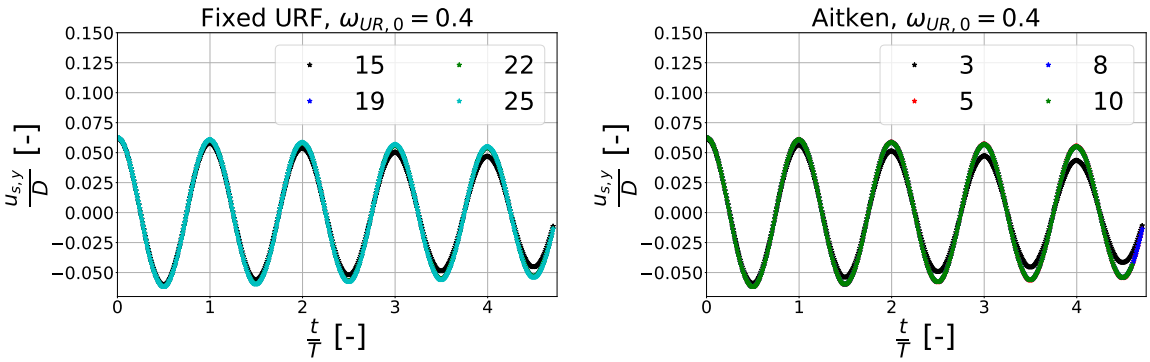


Figure 5.4: Convergence study for the number of FSI subiterations when using the fixed URF (left) and the Aitken (right) subiteration algorithms, with an initial URF $\omega_{UR,0} = 0.4$, using the y component of the lateral midpoint displacement of the rod, where two different values of the intial URF is used for 10 OC at each timestep for the NRG-beamFoam workflow, in combination with a laminar fluid model

Having demonstrated the convergence of both the fixed URF and the Aitken subiteration algorithm, it is now of interest to determine the actual number of FSI loops that each method needs to reach convergence. This is dependent on the initial value of the URF. For the fixed URF method, $\omega_{UR,0}$ remains constant throughout the subiterations, while for the Aitken method, it will adapt based on the residuals obtained throughout the subiteration process. Hence, the value chosen for the URF particularly impacts the number of subiterations of the fixed URF approach, which cannot adapt itself as the Aitken method does. Having made these preliminary observations, for this example, a value of $\omega_{UR,0} = 0.4$ is considered. In practice, this represents a conservative value that a user would choose for unstable computations. For this value of the URF, the values are represented in Figure 5.4. There, it can be seen that the fixed URF needs 19 subiterations to reach converge, while the Aitken algorithm only 8. This implies that under the conditions considered in this test, using the Aitken algorithm reduces the computational cost to approximately half. Of course, this benefit would decrease if a higher value of the initial URF were used. The test nevertheless indicates that the use of the Aitken algorithm brings noticeable computational benefits, and those are expected to be particularly visible for strongly coupled FSI scenarios in which NRG-beamFoam will be used. Of course, one might improve the computational benefits further by implementing a more efficient subiteration algorithm, such as IQN-ILS.

Throughout this Section, only the $y-$ components of the lateral displacements have been presented. However, it is worth noting that an analogous behaviour of the subiterator was also found for the $z-$ component of the lateral displacements. To show this, the same verification tests as shown in Figures 5.2 and 5.3 are presented for the $z-$ component in Figures D.2 and D.3, Appendix D, where one can observe that identical results are obtained. This is expected, since the same initial forcing is applied to both bending planes of the rod. Hence, both the fixed URF and the Aitken subiterators can accomodate for the two bending plane structural model used for this thesis.

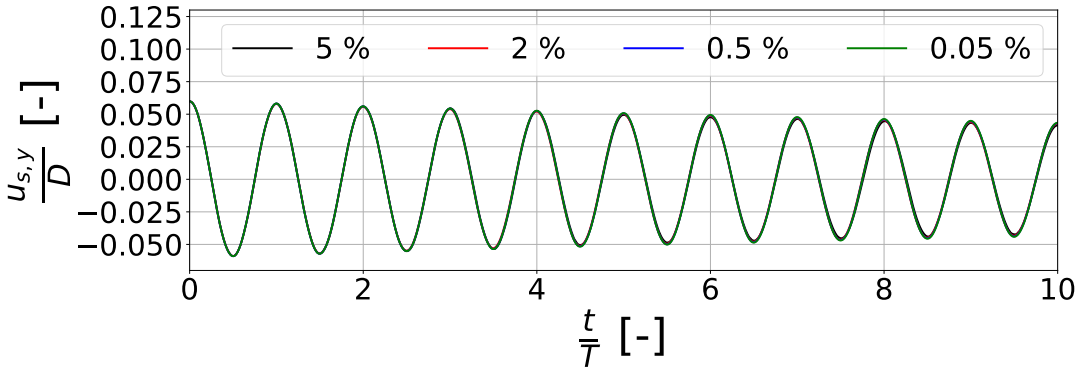


Figure 5.5: The midpoint lateral displacement obtained by the NRG-FSIFOAM workflow for different values of the relative convergence parameter Δ^k, defined in equation (5.1)

5.3. Laminar FSI results

The results presented in Section 5.2 indicate that the subiterator is working as expected, and therefore in combination with the findings in Chapter 4 all components of NRG-beamFoam appear to be functional. This means that FSI simulations can now be run using the newly proposed workflow. This Section presents a comparison of the results obtained with NRG-FSIFOAM and with NRG-beamFoam for a laminar FSI case of the brass beam geometry at a flow velocity of 10 m/s.

First, it is mentioned that although no visualisation of the flowfield is offered for the laminar FSI case, the pressure field qualitatively resembles that observed for the mean pressure in Figure 6.1 when using an AniPFM fluid model. Of course, due to the laminar flow, the skin friction is reduced, and thus one can expect a lower pressure gradient than the one that is observed there.

The setup of the case was already presented in Section 5.1, where the settings for the fluid and the structural domain were presented in detail. However, the final option that the user needs to pick is the convergence criterion for the FSI loop at a given timestep. For NRG-beamFoam, no criterion can be used for the FSI convergence, except for the fixed number of FSI subiterations at each timestep, for reasons explained in Subsection 3.4.3. Based on the results in Section 5.2, for a representative case of the NRG-beamFoam workflow, 8 FSI subiterations are used in combination with the Aitken subiteration algorithm, with a starting URF $\omega_{\mathrm{UR},0} = 0.4$. On the other hand, within NRG-FSIFOAM, the number of subiterations at each timestep can be controlled by computing the following Δ criterion at each k^{th} FSI loop:

$$\Delta^{\mathrm{k}} = \frac{\left\| \mathtt{S} \circ \mathtt{F}\left(\boldsymbol{d}^{\mathrm{k}}\right) - \boldsymbol{d}^{\mathrm{k}} \right\|}{\left\| \mathtt{S} \circ \mathtt{F}\left(\boldsymbol{d}^{\mathrm{k}}\right) \right\|}, \tag{5.1}$$

where $\boldsymbol{d}$ represent the displacements on the FSI boundary, and S and F the structural and the fluid solver operators. To control the FSI loop, both van den Bos [38] and Freitas [41] used a value of $\Delta^{\mathrm{k}} = 0.005$, with an initial URF $\omega_{\mathrm{UR},0} = 0.2$. However, this was never explicitly shown to be a value that ensures FSI convergence. Hence, it was decided to study the dependency between the results of the NRG-FSIFOAM workflow and the value of Δ^{k} imposed in (5.1) in a laminar FSI setting. Results are shown in Figure 5.5, which indeed confirm the validity of the value of Δ^{k} that was previously used. Hence, the same value will also be used for this study. On top of this, a second convergence criterion is imposed: if the desired value of the criterion (5.1) is not reached in 10 subiterations, then the solver moves to the next timestep.

Using the FSI convergence criteria defined above, the comparison between the NRG-FSIFOAM and the NRG-beamFoam results is made using the lateral midpoint displacement. Since the initial amplitude varies between the two workflows due to the different amplitudes of the initial forces applied onto the rod, as was explained at the end of Subsection 5.1.1, for the remainder of this thesis the analysis is made by adimensionalizing the lateral displacements for each of the two setups with their respective initial amplitude value. Since both describe a decaying vibration starting from an initial deformation, their output can be reduced to the vibration frequency and the damping ratio for each case. Hence, these monitors are mainly used to compare the workflows. To determine the vibration frequency and damping ratio, a fit function $u_{\mathrm{s,fit}}$ is applied to each signal, as defined in equation (5.2). Using the midpoint displacements at different timesteps for the two methodologies, the values of the fitting constants $A_{\mathrm{fit}}, y_{\mathrm{fit}}, \omega_{\mathrm{fit}}$, and ϕ_{fit} in (5.2) can be determined. Based on these, the frequencies and damping ratios can be computed using equations (5.3) and (5.4). For the curve fitting, only the displacements after the application of the initial force are used, given that the sinusoidal decay assumption of equation (5.2) does not necessarily correspond to the forced excitation that initially occurs within the system.

$$u_{\mathrm{s,fit}}(t) = A_{\mathrm{fit}} e^{-y_{\mathrm{fit}} t} \sin(\omega_{\mathrm{fit}} t - \phi_{\mathrm{fit}}) \tag{5.2}$$

$$f_{\mathrm{fit}} = \frac{\omega_{\mathrm{fit}}}{2\pi} \tag{5.3}$$

$$\xi_{\mathrm{s,\,fit}} = \frac{y_{\mathrm{fit}}}{\sqrt{y_{\mathrm{fit}}^2 + f_{\mathrm{fit}}^2}} \tag{5.4}$$

Having described the monitors used for comparing the two workflows, the transient midpoint displacement is presented in Figure 5.6, and the corresponding frequency and damping in Table 5.2. It is worth mentioning

that the uncertainty of the results was not quantified, and therefore the differences may also stem from here. Nevertheless, a good matching between the two methods can generally be observed: for both the frequency and the damping ratio the relative differences between the NRG-FSIFOAM and the NRG-beamFoam outputs are within 1.5% of each other.

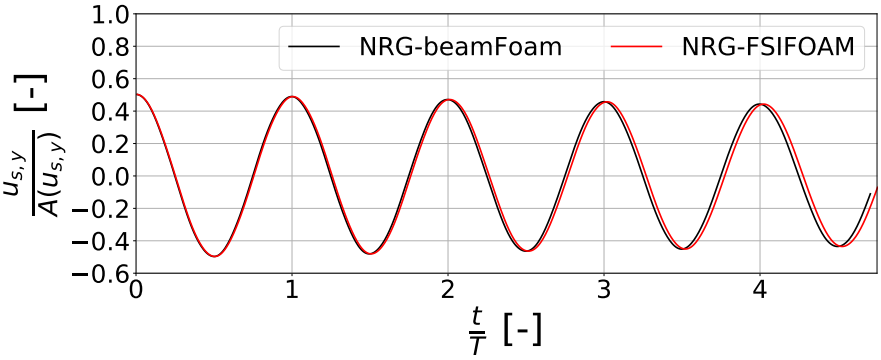


Figure 5.6: The y component of the lateral midpoint displacement of the rod for the laminar FSI case, for the proposed NRG-beamFoam workflow, compared with the current NRG-FSIFOAM one

	f_{fit} [Hz]	$\xi_{\text{s, fit}}$ [-]
FSIFOAM	25.645	0.00509
beamFoam	25.806	0.00516

Table 5.2: Comparison between the frequency and the damping ratio obtained for the two FSI methodologies for a laminar fluid model

In Figure 5.6, the difference in the frequency between the two methodologies is more visible, since a phase shift can be observed between the outputs for high values of $\frac{t}{T}$. This can be caused by multiple differences between the two workflows, such as the mapping routines. A particular cause of the change in vibration frequency could also be inherently related to the different structural solvers used within NRG-FSIFOAM and NRG-beamFoam. The former uses a Lagrange Q Finite Element approach, where each cell has two nodes in every direction of the Cartesian frame of reference, and where Lagrange polynomials are used as shape functions. The local stiffness matrix of each element is obtained based on the said Lagrange polynomials. To compute the mass matrix, a Gauss quadrature rule is used, based on the same Lagrange shape functions as used to build the stiffness matrix [146]. In this regard, the mass matrix is consistent. By contrast, NRG-beamFoam uses the Hermitian shape functions (2.71)-(2.76) and (2.84)-(2.87), by assuming that the bending planes are independent of one another, and by computing the stiffness and mass matrices for each bending plane using a consistent approach. Since the shape functions and the quadrature rules for the two structural solvers are completely different, one could also expect that the stiffness and mass matrices would vary. Hence, the eigenfrequencies and eigenmodes computed with each of the two solvers should also be slightly different, given that they form the eigensolution to the generalized eigenproblem formed by the mass and the stiffness matrix, equation (2.98). This hypothesis can be quantified by using the NRG-FSIFOAM and the NRG-beamFoam workflows in a similar FSI setting, but with air as a working fluid. For such a situation, the FSI coupling would be non-existent given the low ratio of density between the air and the rod, and the potential differences in the frequency obtained could only originate from the inherent behaviour of the structural solvers within the two methodologies.

In conclusion, the newly implemented NRG-beamFoam workflow offers very similar results to the NRG-FSIFOAM methodology for a short-period laminar FSI case of the brass beam. The differences in frequency and damping of the rod's motion are small and can be attributed to, for example, the different structural solvers employed within the two methodologies.

5.4. URANS FSI

The previous Section proved that NRG-beamFoam works well when using a laminar fluid model. It is of more engineering interest, however, to determine its performance when using a turbulent fluid model, since this is the regime that is almost exclusively found in nuclear reactors' core axial flow. For the NRG-beamFoam methodology, the only change compared to the laminar case is the nature of the inputs to the load interpolator: now, the turbulent stresses are also included. In practice, the use of a turbulent fluid model also required other modifications to the setup of the case, as will be detailed in Subsection 5.4.1. Similar to the laminar FSI case, it is of interest to not only present the numerical settings, but also to investigate the reduction of

the numerical errors, as well as to compare the results obtained with the two workflows. This is presented in Subsection 5.4.3. Since the URANS case is also of practical interest for axial flow TIV investigations, in Subsections 5.4.2 and 5.4.4 the method and the associated results obtained by comparing the computational costs of NRG-beamFoam and NRG-FSIFOAM are presented.

5.4.1. Case setup

Given that both workflows performed well for a laminar FSI case, the same settings presented in Section 5.1 are maintained for the URANS case wherever possible. However, for the turbulence modelling, the k and the ω fields need to be additionally computed, and thus, new discretization schemes and solver settings are provided in Appendix D, Subsections D.2.1 and D.2.2, respectively. These are the same as used by van den Bos [38]. Furthermore, the new boundary conditions are presented in Table 5.3. At the inlet, the values of both are provided as constant for the entire patch. The values were calculated by assuming a turbulent intensity of 5% and a turbulent length scale of 0.1 cm [38]. For the outlet, zero Neumann boundary conditions are prescribed. Though the value of k exactly at the solid inner and outer walls is 0, a slightly higher value of 1e-12 is imposed for stability. For the specific dissipation rate ω, a specific `omegaWallFunction` BC is provided, which computes its values at the center of the fluid cells adjacent to the wall based on the local value of y^+. The initial conditions for the problem are obtained similarly to the laminar case, that is, a fluid-only simulation is run until convergence, before the coupling between the fluid and the structure is switched on. Of course, the difference with the laminar case is that now for the fluid-only case a URANS fluid model is used. For all pure URANS FSI simulations in this thesis, the same fluid model and the same initial forcing are used as for the laminar case.

The discretization settings for the structure in the two workflows also remain the same as they were for the laminar case: since only the turbulent loads are added to the laminar case, it is expected that the load pattern in the axial direction of the rod will not change significantly compared to the laminar case. Hence, the same number of elements and segments can be used for the NRG-beamFoam workflow, and the same structural grid for the NRG-FSIFOAM methodology. Another consequence of the similar fluid conditions is also that the same timestep can be used, since neither the Courant number nor the rod dynamics are expected to be significantly affected by the turbulent stresses.

patch	k		ω	
	type	value $\left[\frac{m^2}{s^2}\right]$	type	value $\left[\frac{1}{s}\right]$
inlet	`fixedValue`	0.375	`fixedValue`	612
outlet	`zeroGradient`	-	`zeroGradient`	-
inner	`fixedValue`	1e-12	`omegaWallFunction`	-
outer	`fixedValue`	1e-12	`omegaWallFunction`	-

Table 5.3: The additional boundary conditions imposed for the fluid domain for the URANS FSI simulations

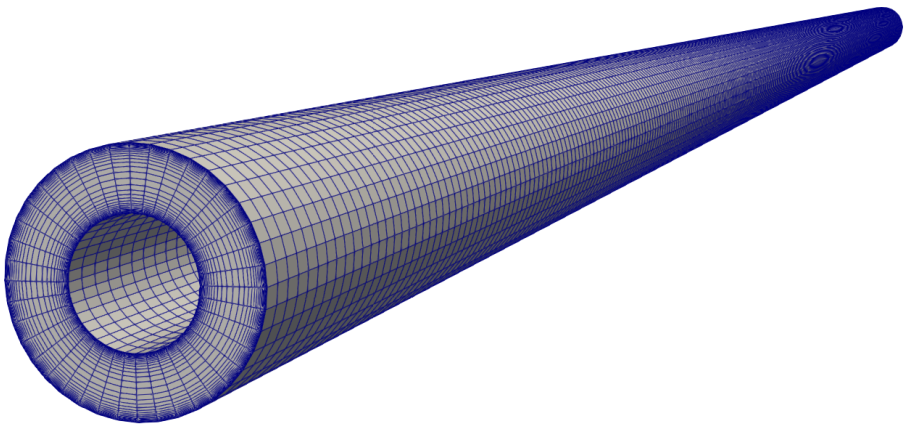


Figure 5.7: The fluid mesh used for the URANS FSI simulations in this Chapter

However, three differences compared to the laminar FSI case need to be pointed out. First, given the need to now model the turbulence, the resolution of the mesh had to be increased. The fluid mesh of van den Bos [38], which he generated for both pure URANS and for URANS-AniPFM FSI simulations, was obtained at the end of a mesh convergence study using the same settings for the URANS fluid model as the ones in this study. Hence, the same fluid grid is used for the study herein. This consists of a discretization where 400 uniform axial elements are used, along with 50 in the radial and 40 in the circumferential direction. A visual representation of the mesh is given in Figure 5.7. Additionally, for the NRG-beamFoam methodology, stability issues were observed if the turbulent transport equations for k and ω were not solved at all subiterations. No explanation for this could be formulated. Nevertheless, to ensure that the turbulent fields are recomputed for each outer corrector of the PIMPLE loop, when using the NRG-beamFoam workflow with a URANS fluid model, `turbOnFinalIterOnly` must be set to `false` in the `PIMPLE` subdictionary of `fvSolution`. Finally, since the URANS case is closer to the phenomena encountered in real-life applications in nuclear reactors, a longer simulation time was used than for the previous laminar case, that is, 0.5 seconds instead of 0.2. This corresponds to approximately 12 periods of the rod's vibration.

5.4.2. Determination of the computational times for the two workflows

Given that the main practical objective of the NRG-beamFoam workflow is to reduce the computational cost of NRG-FSIFOAM while keeping the same level of accuracy, this thesis' outcome should contain a comparison of the computational costs of the two methodologies. How the CPU time data is collected for the comparison between NRG-FSIFOAM and NRG-beamFoam is presented in this Subsection. It is worth reminding that the NRG-beamFoam workflow is intended to replace all steps of the FSI loop, except for the computation of the fluid solution and the mesh motion solver, as can be seen in Figure 3.1. Thus, for a fair comparison of the two workflows, one should use the entire computational time spent inside the NRG-beamFoam workflow with the time that is spent outside of the fluid solver and the mesh motion solver within NRG-FSIFOAM.

For the NRG-FSIFOAM workflow, use was made of the performance analysis features of preCICE. The software creates a standalone `.json` file for both the fluid and the solid domains, based on which the user can visualize a timeline of the different operations that take place for each solver from the coupling algorithm's perspective: the communication of raw information between the fluid and the structural solvers, the associated computation of the mapping between the two grids, and the actual computation of the solution for each of the two domains. Starting from this, at the end of the simulation, a summary file is generated for each solver, where one can find statistics about the number of occurrences and the total computational time associated with the solver solution at each subiteration, as well as the associated mapping and communication with the other mesh. It is worth mentioning that different timelines are obtained for each of the processors participating in the parallel solution of the structural or fluid problem. The summary file is obtained based on the CPU time of the master processor of the simulation. An example of both outputs can be seen in Figure D.4, Appendix D. Based on the entries in the summary file, the total time spent throughout the transient computations can be split into two categories: the total time spent within OpenFOAM, contained in the `solver.advance` entry, and the CPU time spent advancing the FSI coupling, `advance`. Since both the computation of the fluid fields and the mesh motion are executed in OpenFOAM, the entry `advance` represents the CPU time that is directly comparable to the set of routines that are desired to be optimized using NRG-beamFoam, for the master processor of the OpenFOAM computation.

For the NRG-beamFoam workflow, the CPU time is obtained by using an instance of the `clockTime` OpenFOAM class, which has a precision of $2 \cdot 10^{-6}$ s for timing the execution of given routines within the developed code. Four different entries are created for the CPU times: fluid solver, load interpolator, FEM solver, and displacement interpolator. To keep the results comparable with NRG-FSIFOAM, all of these are only computed on the master processor within the OpenFOAM parallel run. Throughout the transient, time is added to each of these entries by using the `timeIncrement()` method of the `clockTime` class after the routines associated with the load interpolator, the FEM solver, and the displacement interpolator have been executed inside the `BeamFoamDisplacement.updateCoeffs()` method. In particular, the timing of the fluid solver time starts at the end of the `BeamFoamDisplacement.updateCoeffs()`, and it ends once a new call of the same function is made. In this regard, similar to the outputs of preCICE, inside the fluid solver entry of NRG-beamFoam's CPU time both the mesh motion and the computation of the fluid domains are computed. Thus, to compare NRG-beamFoam to its predecessor, the summation of the CPU times spent inside the load interpolator, the FEM solver, and the displacement interpolator will be used.

5.4.3. Frequency and damping ratio comparison

For the main URANS results, the same monitors as for the laminar case will be used. However, to offer a fair comparison between the two workflows, the FSI convergence is first analyzed for each. Given the small variation found for the NRG-FSIFOAM results, Figure 5.5, a similar study was not conducted for the current URANS case. Instead, the same convergence criteria were used for the NRG-FSIFOAM workflow as for the laminar case ($\omega_{UR,0} = 0.2, \Delta^k = 0.005$, and no more than 10 subiterations for each timestep). However, for NRG-beamFoam, a similar study of the convergence of the results was conducted, for varying numbers of OC for the PIMPLE algorithm. Results will be presented for 10 to 40 subiterations at each timestep, where an initial URF $\omega_{UR,0} = 0.4$ is used. Though a simulation with 5 outer correctors was also attempted, it led to a premature crash of the simulations and is therefore not presented here. To better quantify the differences between the NRG-beamFoam results with different numbers of OCs, the relative difference in the Root-Mean Square (RMS) of the damped signal compared to the case when 40 OCs is used:

$$\Delta\text{RMS} = \frac{\text{RMS} - \text{RMS}_{40\text{OC}}}{\text{RMS}_{40\text{OC}}}. \tag{5.5}$$

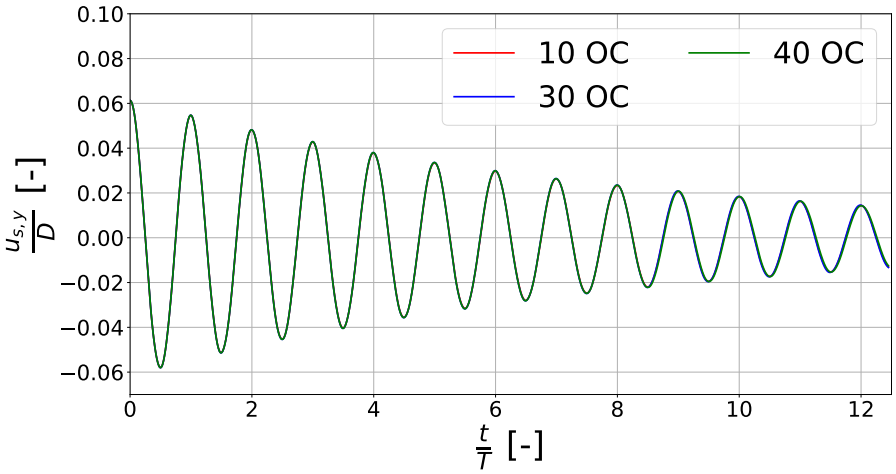


Figure 5.8: Convergence study for the number of FSI subiterations when using the Aitken subiteration algorithm, with an initial URF $\omega_{UR,0} = 0.4$, using the y component of the lateral midpoint displacement of the rod, for the NRG-beamFoam workflow, in combination with a URANS fluid model

# of OC	Δ RMS [%]
10	0.3
20	0.25
30	0.08
40	0

Table 5.4: The RMS amplitude variation for the URANS OC convergence test

The OC convergence study for NRG-beamFoam is presented in Figure 5.8, with the variation in the RMS of the damped signal in Table 5.4. A small variation between using 10 and 40 OC is seen in the results, namely, a 0.3% variation in the RMS of the signal. This marginal improvement is obtained by increasing the computational cost approximately fourfold. Hence, in the following, 10 OCs are used for the NRG-beamFoam workflow, as a good balance between accuracy and computational efficiency.

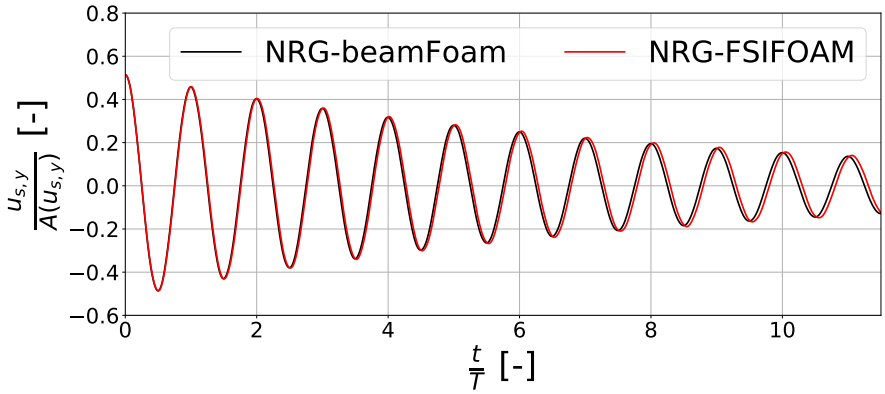


Figure 5.9: The y component of the lateral midpoint displacement of the rod for the URANS FSI case, for the proposed NRG-beamFoam workflow, compared with the current NRG-FSIFOAM one

	f [Hz]	$\xi_{s,\text{fit}}$ [-]
FSIFOAM	25.676	0.01937
beamFoam	25.808	0.01928

Table 5.5: Comparison between the frequency and the damping ratio obtained for the two FSI methodologies for a URANS fluid model

Having performed the FSI convergence study, the comparative results between NRG-FSIFOAM and the NRG-beamFoam workflows are presented in Figure 5.9 and Table 5.5. Compared to the laminar case, the changes in frequency are minor, approximately 0.02 Hz for NRG-FSIFOAM and 0.002 for NRG-beamFoam. This is expected, given that the frequency is mainly influenced by the eigenfrequencies that are inherent to each structural solver of the two methodologies, and by the added mass effect of the fluid that is obtained through the coupling. Neither of these was expected to be influenced by the effects of turbulence developed within the fluid flow. On the other hand, the damping ratio is almost quadrupled when going from a laminar to a URANS fluid model. This can be attributed to the larger force components that are developed by the turbulent structures normal to the wall [140], and hence to the larger energy transfer between the two media. For the URANS FSI case, the damping ratios for the two considered methodologies remain similar to one another, with a relative difference of less than 0.5%. In contrast to the laminar FSI case, however, the damping ratio obtained for NRG-FSIFOAM is larger than the one for the NRG-beamFoam. These differences could be attributed, for example, to the changes in total energy within each FSI system due to the mapping of stresses and displacements on the FSI boundary. At the same time, the slightly different results may be a consequence of the uncertainty of using the fit function (5.2).

5.4.4. Computational comparison

First, an analysis of the computational time within the NRG-FSIFOAM workflow is given. This is presented in Figure 5.10, where the outputs of preCICE for the Fluid solver are presented. First, the left chart of Figure 5.10 is commented. The entries "Fluid solver" and "FSI advance" correspond to `advance` and `solver.advance` in the preCICE summary files of the fluid domain, as introduced in Subsection 5.4.2. Hence, the chart on the left of Figure 5.10 shows that the fluid solver of NRG-FSIFOAM spends approximately half of the time (51.1%) computing the solution for the mesh motion and the fluid fields. The rest of the time (48.89%) is spent on waiting for preCICE to communicate the boundary displacements based on the outputs of the structural solver that are required for the next subiteration. Although the fluid grid is significantly larger than that of the structure, as was described in Subsection 5.4.1, the former domain was computed using 60 cores, and the latter using only 1. It was therefore expected that the simulation time for the two domains would be comparable, with the initial mapping of the displacements from the structure taking up a small portion of the processing power. Upon inspecting the timeline of the different routines executed by the fluid solver in contrast with those of the structural solver inside NRG-FSIFOAM, it was found that the computation of the structural solution is longer than that of the fluid. This is visible in Figure D.5, Appendix D, for a random subiteration of the FSI transient. Hence, most of the time associated with the "FSI advance" entry on the left of Figure 5.10 consists of the fluid solver waiting for the structural displacements on the FSI boundary to be computed.

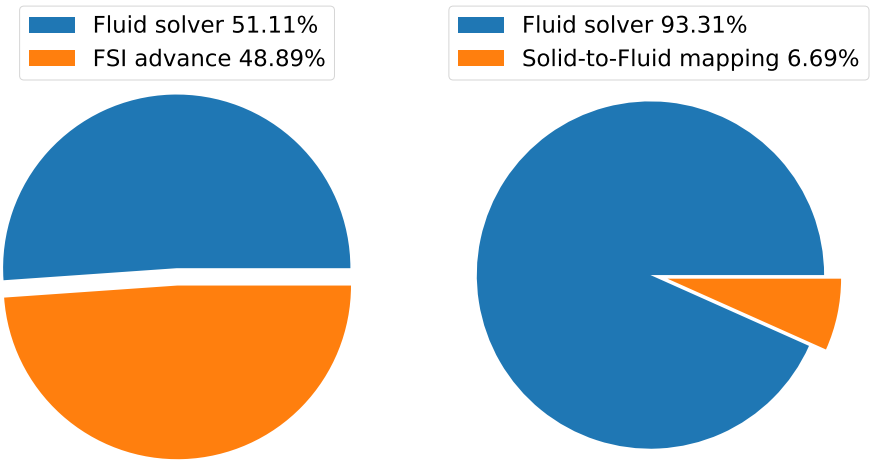


Figure 5.10: Left: the CPU time spent within the NRG-FSIFOAM workflow on the fluid and the rest of the FSI routines; right: comparison between the CPU time spent on obtaining the fluid solution, and that on mapping its necessary inputs

The practical implications of the high computational costs for the structure in NRG-FSIFOAM are now given. It is important to note that the deal.II solver used for the NRG-FSIFOAM methodology can only be run in serial. Thus, the structural solver in deal.II represents a bottleneck of the entire worfklow: if one wanted to speed up the computations by increasing the number of cores with which the fluid domain is decomposed, then each of these would have to wait for the structural solution to be computed by the single core allocated to the structural solution first, before moving on to the next subiteration. Hence, for the current setup, if the number of allocated processors were increased, the total computational time would remain the same. Despite this, to be able to compare the general performance of a 3D Lagrange element FEM solver, as used by deal.II within NRG-FSIFOAM, to the 1D beam element FEM used by NRG-beamFoam, it is assumed that the Lagrange-element structural solver within NRG-FSIFOAM is executed faster than the fluid solver for the considered problem. In such a situation, the "FSI advance" entry on the left of Figure 5.10 would be approximately equal to the time that is spent on mapping the solution from the structural domain. Within the preCICE summary files, the total time specifically spent on transferring this coupling information is also provided. Hence, this hypothetical situation is presented in Figure 5.10, where for the considered simulation a faster structural solver would imply that approximately 6.7% of the computational time would be spent on mapping the solution from the brass bream to the fluid domain surrounding it. This will be used in the following for a comparison with the NRG-beamFoam workflow.

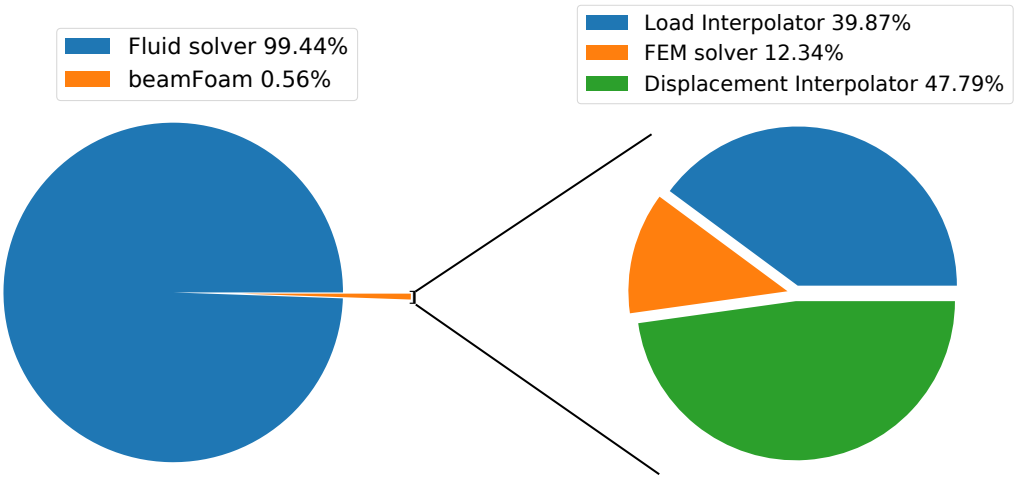


Figure 5.11: Left: the CPU time spent within the NRG-beamFoam workflow on the fluid and the beamFoam routines; right: the breakdown of the CPU time spent for the different classes inside beamFoam

Having discussed the NRG-FSIFOAM performance for a URANS FSI case, the same is done for the NRG-beamFoam workflow, where the associated results can be visualized in Figure 5.11. On the left chart of Figure 5.11, it can be seen that all of the routines within NRG-beamFoam only take up 0.55% of the computational time. This implies a speed-up of more than 48% compared to the nominal NRG-FSIFOAM settings provided by van den Bos [38], provided that the same number of FSI subiterations are executed. Assuming that the structural solver were faster than the fluid one, it can be observed that the use of NRG-beamFoam still reduces the computational costs by approximately 6% for a URANS FSI case, while offering very similar results of the rod's vibration, as seen in Figure 5.9 and Table 5.5.

In light of the newly discovered bottleneck of the NRG-FSIFOAM workflow, it is of interest to also understand whether the same could occur for the NRG-beamFoam workflow. As was already mentioned in Subsection 3.3, the beamFoam part of the methodology works in parallel on each core that is used to decompose the fluid domain, and the same operations are executed by each core. In principle, this implies a similar situation as for the NRG-FSIFOAM workflow, where the fluid solver is stopped until the structural solution is computed. However, given the small number of unknowns for the structural problem, the computational overhead should remain reduced, even once more complex configurations are tested.

It is also of practical interest to observe the relative computational cost of each of the three main modules of NRG-beamFoam, as seen on the right of Figure 5.11. For the current case, the FEM solver only accounts for 12% of the computational costs, while the rest of the processing power is spent on the mapping routines. This implies that the ROM solver, which is intended to reduce the computational costs of the FEM solver, will have a very marginal impact on the total computational cost of the FSI simulations using NRG-beamFoam. It is then important to observe how the shares between the different modules, as seen on the right of Figure 5.11, will vary when using more complex configurations to compute axial-flow TIV behaviour.

As an ending to this Chapter, it is important to point out the current limitations of the NRG-beamFoam workflow. In comparing the two workflows, it was assumed that the same fluid solver costs are present on the right of Figure 5.10 and on the left of Figure 5.11, that is, the same fluid solver costs are incurred in both the NRG-FSIFOAM and the NRG-beamFoam methodologies. However, this is not currently true. Although the same settings are essentially used for the fluid models within both workflows, the FSI convergence criteria for the two are different, where NRG-beamFoam always used 10 OC, while NRG-FSIFOAM adapted its number of subiterations for each timestep based on the criterion (5.1). In practice, this led to an average of between 7 and 8 subiterations. The computational benefits of NRG-beamFoam presented in this Chapter are valid under the condition that the same number of subiterations would be executed using the two workflows. This is possible if a similar criterion (5.1) is used for the control of the PIMPLE algorithm within the NRG-beamFoam workflow, and if no difference in the FSI convergence occurs because of the different vectors that are used for the displacement convergence. More specifically, NRG-FSIFOAM uses the displacements for the Lagrange-element grid as $\boldsymbol{d}^{\mathrm{k}}$ in (5.1), while NRG-beamFoam uses either the nodal displacements (for the FEM element solver), or the generalized modal coordinates (for the ROM solver). On top of this, differences in the number of subiterations may appear due to difference in performance between the IQN-ILS (used by NRG-FSIFOAM), and the Aitken algorithm (implemented in NRG-beamFoam), where the former is known to be better performing. Nevertheless, the results in this Subsection show that the mapping routines and the structural solver within NRG-beamFoam present inherent advantages over those in NRG-FSIFOAM thanks to formulating the structural problem using a small number of unknowns.

6

Towards a robust integration of AniPFM with NRG-beamFoam

Chapter 5 showed that the proposed NRG-beamFoam workflow can be reliably used for a URANS fluid model, and that it can potentially reduce the computational cost of the currently used NRG-FSIFOAM workflow. However, using a URANS fluid model is generally not satisfying for simulating TIV phenomena, since this leads to an underestimation of the amplitude of the structural vibrations [35]. Because of this, within NRG a novel AniPFM fluid model was developed, whose mathematical model was described in Subsection 2.2.3. In the current Chapter, a description is offered of the attempts that were made in this thesis to combine the most recent iteration of the AniPFM solver with NRG-beamFoam. In Section 6.1, a description of the settings used to run FSI simulations with the AniPFM model is offered for both the NRG-FSIFOAM and the NRG-beamFoam workflows, with the associated results reported in Section 6.2.

6.1. Case setup

Similar to all previous FSI results presented in this work, the brass beam geometry is used, for an axial flow velocity at 10 m/s. What's more, the transport properties for the fluid remain the same as for the rest of the FSI cases in this work. For this Chapter, it is generally attempted to use the same settings as those of Freitas [41], who presented the most recent advancements of the AniPFM model. Thus, for the fluid discretization, the mesh that was most recently employed for NRG-FSIFOAM cases is also used here. This specific discretization uses the same radial and circumferential discretization as the mesh used in Chapter 5 for the URANS cases, but with 300 uniform axial elements instead of 400. This reduces the computational costs, and it was shown to cause a small change in the uncertainty of the results for the specific settings of the AniPFM model that will be used in this Chapter [41]. More specifically, 256 modes are used for the Fourier decomposition of the non-dimensional velocity fluctuations. Unless specified otherwise, the random seed used to generate the wavenumber vectors for each of these modes is equal to the default value, 0. As was already mentioned in Subsection 2.2.3.2, the C&EC time correlation method is used for the computation of the non-dimensional velocity fluctuations at the current timestep, with a value of the control parameter $f_\tau = 25$, as for higher values the two point-correlation of the velocity fluctuations doesn't significantly change for a turbulent channel flow geometry, thereby suggesting a similar decorrelation for the brass beam case [41]. To model the TKE spectrum, the wavenumber space was discretized using 128 intervals, and the maximum wavelength l_{user} was taken as equal to the hydraulic radius of the geometry. The wavelength at which the TKE is highest, k_{e}, is computed using the cubic cut-off formulation, (2.34), with $C_{\text{l}} = 3$.

The additional discretization schemes and solver settings associated with the AniPFM model are presented in Appendix E, Subsections E.1 and E.2. For the former, the same settings as used by Freitas [41] are also imposed for this work. For the latter, slightly more stringent residual criteria are imposed for the fluctuating fields, in the attempt of stabilizing the computations.

patch	Ufluct		pfluct	
	type	value $\left[\frac{\text{m}}{\text{s}}\right]$	type	value $\left[\frac{\text{m}^2}{\text{s}^2}\right]$
inlet	mapped	(0 0 0)	fluctuatingPressure	-
outlet	mapped	(0 0 0)	fluctuatingPressure	-
inner	noSlip	-	zeroGradient	-
outer	noSlip	-	zeroGradient	-

Table 6.1: The additional boundary conditions imposed for the fluid domain for the URANS AniPFM simulations

The use of the fluctuating pressure and velocity fields within the AniPFM model also leads to the requirement of prescribing boundary conditions for both. For the current setup, these are presented in Table 6.1. At the walls, both fluctuating fields should satisfy the same boundary conditions as for the mean fields computed for URANS simulations, that is, a zero Neumann boundary condition for the pressure fluctuations (corresponding to $\frac{\partial p'}{\partial n} = 0$), and a no-slip condition for the velocity fluctuations ($\mathbf{u_t} = \mathbf{0}$). This is in agreement with the results that one obtains, using, for example, the Reynolds decomposition [34]. For the inlet and the outlet, the velocity fluctuations are obtained by mapping fluctuations that already exist inside the domain, as explained in Subsection 3.4.6. For the pressure flucutations, the `fluctuatingPressure` BC is used. This uses equation (3.26) to obtain the pressure fluctuations on the patch starting from the mapped velocity fluctuations.

The strategy for obtaining the initial conditions is similar to the previous URANS and laminar FSI cases, namely, a fluid-only simulation is run before the FSI coupling is activated. For this preliminary simulation, first, a URANS simulation is run until acceptable convergence is achieved. Subsequently, the latest URANS results are used as initial conditions for an AniPFM fluid-only calculation. This is simulated for approximately 5 seconds, that is, approximately 40 flow passes, until the pressure fluctuation RMS field has stabilized. Once the FSI coupling is switched on, in contrast to the laminar and URANS cases, no initial forcing is applied to the rod. Instead, the pressure fluctuations on the FSI boundary are mapped to the structural domain and produce the dynamic response of the rod, giving rise to FSI.

Next, the settings for the structural domains are discussed. The same structural grid was used for the AniPFM simulations using the NRG-FSIFOAM methodology as those for the URANS case. This corresponds to the same FEM mesh used by van den Bos [38] and by Freitas [41] for their AniPFM FSI results using NRG-FSIFOAM. Similarly, for the NRG-beamFoam workflow, the same discretization as for the laminar and the URANS cases is initially used, given its satisfying previous accuracy. Hence, unless specified otherwise, for the results presented in this Chapter using the NRG-beamFoam methodology, 20 beam elements are used, along with 6 segments for each of these to map the fluid loads to the structural solver. For the time marching of the problem, a timestep $\Delta t = 2 \cdot 10^{-4}$ is used for both the structural and the fluid solvers. This corresponds to the settings used by Freitas [41], and is motivated by the stability requirements of the C&EC method of the non-dimensional velocity fluctuations.

6.2. Results

To correctly run the NRG-beamFoam workflow with the AniPFM model, it was required to determine the number of FSI subiterations, as well as how the AniPFM should be run. Apart from these, an unexpected offset in one bending plane that was initially observed for the NRG-beamFoam methodology is analyzed. These outcomes are presented in Subsection 6.2.1. Once these are established, simulation results to determine the amplitude of the vibrations due to TIV are shown in Subsection 6.2.2 for both NRG-FSIFOAM and NRG-beamFoam.

6.2.1. Preliminary NRG-beamFoam study

First, for visualisation purposes, the mean and fluctuating pressure fields that are imposed as initial conditions for the FSI simulations are presented in Figure 6.1. There, the scales of the generated fluctuations can be observed in the axial and the radial direction. What's more, for the mean pressure distribution, as expected, the adverse pressure gradient caused by the skin friction only develops in the axial direction. In other words,

at each axial location of the beam, a uniform pressure distribution can be found.

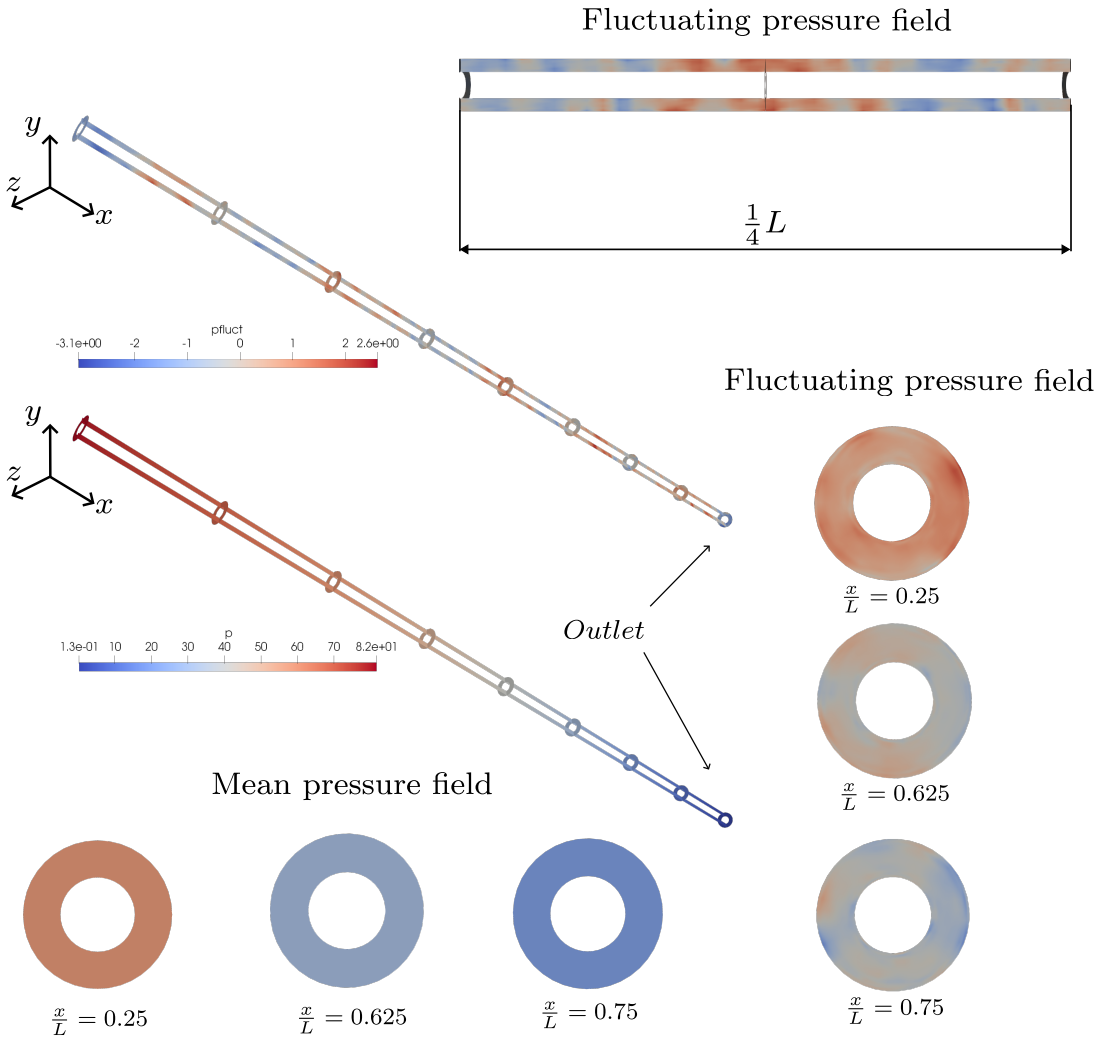


Figure 6.1: The fluctuating and mean pressure fields at $\frac{t}{T_{\text{pass}}} = 0$

Next, before any reference results were obtained, it was of interest to ensure that FSI convergence is satisfied for both workflows. For NRG-FSIFOAM, the same convergence criterion as for the URANS case was used ($\omega_{\text{UR},0} = 0.2, \Delta^{\text{k}} = 0.005$, no more than 10 subiterations for each timestep), given the small variations seen with the value of Δ^{k} for the laminar FSI case in Figure 5.5. For NRG-beamFoam, a shorter study of the number of required OC is undertaken than for the previous FSI cases, by conservatively assuming that the number of OC required for FSI simulations with the AniPFM fluid model is equal to or higher than the 10 subiterations used for the URANS case. Though this assumption is simplistic, and potentially not optimal from a computational perspective, it enabled a quick FSI convergence study. Hence, simulations with 10 and 20 OC are run using the NRG-beamFoam workflow, for 0.4 seconds, using an initial URF $\omega_{\text{UR},0} = 0.4$. As a convergence monitor, the midpoint lateral displacements are again used. The results are presented in Figure 6.2, where the time is adimensionalized using the average flow pass time through the domain, $T_{\text{pass}} = \frac{L}{U_{\text{axial}}} = 0.119$ s. There, it can be observed that the differences between the two different numbers of FSI subiterations are marginal, and hence in the following 10 OC will be used for the NRG-beamFoam workflow.

Apart from the number of FSI subiterations that should be executed at each timestep, another important setting for NRG-beamFoam is whether for the AniPFM fluid model the anisotropic velocity fluctuations should be computed once per timestep, or at every subiteration. Within the NRG-FSIFOAM workflow, based on how preCICE communicates with OpenFOAM, only the latter is possible. However, this is computationally expen-

sive, without bringing significant modelling benefits. Indeed, at a given timestep, the velocity fluctuations are expected to be a consequence of the newly generated non-dimensional random velocity fluctuations, and the convection of the perturbations that were already existing inside the domain. On the other hand, the FSI subiteration algorithm is necessary to ensure the equilibrium between the fluid loads and the structural displacements at the FSI interface. Hence, apart from the change in turbulence caused by the movement of the FSI boundary, the generation of synthetic turbulence and the FSI convergence can be seen as virtually independent of one another, thus implying that a single computation of the AniPFM routines per timestep would be required. To confirm this hypothesis, a comparison of the midpoint displacements for the case when the AniPFM is run once per timestep and at every subiteration, respectively, is conducted, for 0.2 seconds using NRG-beamFoam, and is presented in Figure 6.3. Indeed, the differences obtained between the two are not significant, suggesting that in the following the AniPFM model can be run only once per timestep. This represents an important practical advantage of the NRG-beamFoam workflow, since van den Bos [38] found that when using NRG-FSIFOAM with the AniPFM fluid model, 44.5% of the total computational time is spent on determining the velocity and the pressure fluctuations. Since up to 10 subiterations were used at each timestep for the NRG-FSIFOAM methodology, reducing the number of AniPFM computations to a single one for a timestep could drastically reduce the overhead of using the AniPFM model for FSI computations.

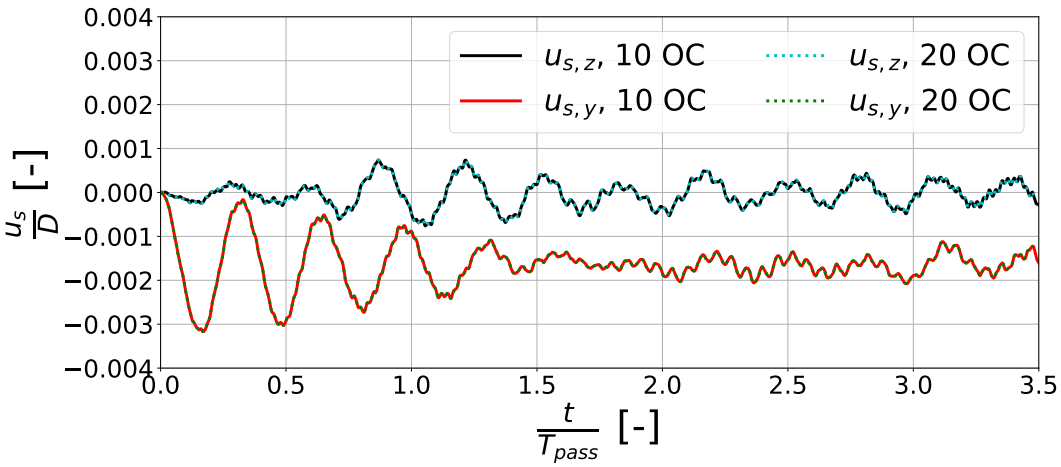


Figure 6.2: Convergence study for the number of FSI subiterations when using the Aitken subiteration algorithm, with an initial URF $\omega_{UR,0} = 0.4$, using the lateral displacement components of the midpoint of the rod, for the NRG-beamFoam workflow, in combination with the AniPFM fluid model

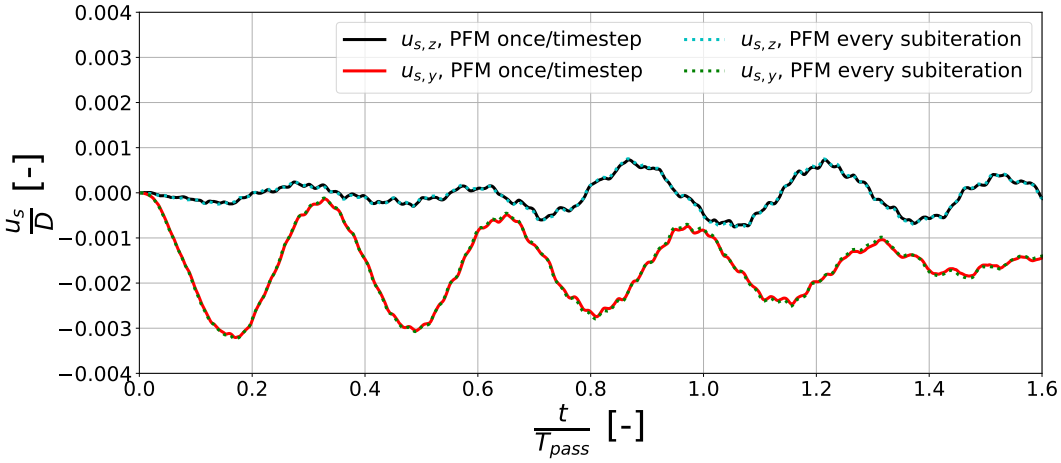


Figure 6.3: Comparison of the results obtained using the NRG-beamFoam workflow for the midpoint lateral displacements, when computing the velocity fluctuations of the AniPFM model once for each timestep, and at each subiteration

Having determined the number of subiterations and the correct call of the AniPFM routines within the NRG-beamFoam methodology, a peculiar behaviour of the outputs of the workflow when using the AniPFM is discussed. More specifically, in both Figures 6.2 and 6.3, one can observe that the midpoint vibrations in the $y-$ direction that are computed by the model are not centered around 0, that is, the oscillations are not computed around the undeformed configuration of the rod. On the other hand, as expected, the vibrations in the $z-$ direction do exhibit the expected behaviour. At first glance, this may appear to be caused by a difference between how the proposed methodology computes the loads or the displacements within the two bending planes. However, the equivalence between the two was shown for all modules of NRG-beamFoam, and is presented in Appendices C.1 and C.3 for the FEM solver and the displacement interpolator, and in D.1.4 for the subiterator.

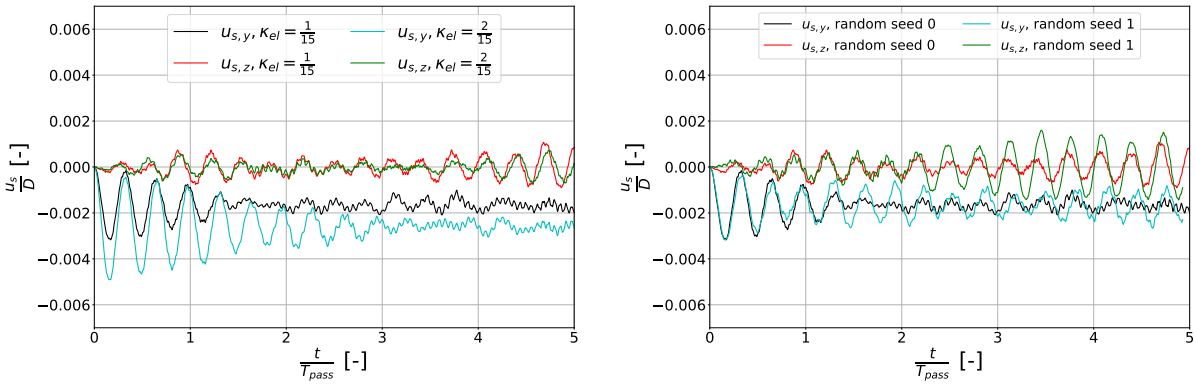


Figure 6.4: The changes in the midpoint displacements of the rod when using two times more structural elements (left), and a different random seed (right), compared to the initial configuration described in Section 6.1, for the NRG-beamFoam workflow, when using the AniPFM fluid model

To try and identify the source of this behaviour, firstly, it was supposed that it may be a consequence of an aliasing error, that is, too few beam elements were used. This was examined for a transient of 0.6 s, using 40 beam elements instead of 20, while still using 6 segments to divide each beam element within the load interpolator for both structural discretizations. The outcome can be analyzed on the left of Figure 6.4, where:

$$\kappa_{\text{el}} = \frac{\text{\# of structural elements}}{\text{\# of fluid elements}} \text{ (on the FSI boundary).} \tag{6.1}$$

Analyzing the left side of Figure 6.4, for the finer discretization, the vibrations in the $z-$ direction are still observed around the equilibrium position, while for the $y-$ direction, oscillations occur around a mean that is even farther from the zero position. Hence, it can be concluded that the offset is not a consequence of a coarse structural discretization and that it is potentially made worse by increasing the number of FEM elements.

Next, it was hypothesized that the offset could be a consequence of the sensitivity of the NRG-beamFoam methodology to the initial conditions of the problem. To this end, a simulation was run using the initial structural discretization of 20 beam elements and 6 segments, this time using, however, a random seed for the AniPFM equal to 1. For this case, results are presented on the right of Figure 6.4. For this configuration, similar values for the vibration means in the $y-$ and the $z-$ directions as for the initial case are seen. Thus, the error also appears to be independent of the seeding of the problem.

The problematic behaviour was finally improved when the number of segments was reduced for each element. In Figure 6.5, results are presented using the parameter:

$$\kappa_{\text{seg}} = \frac{\text{\# of segments}}{\text{\# of fluid elements}} \text{ (on the FSI boundary).} \tag{6.2}$$

The initial, problematic configuration used 6 segments, $\kappa_{\text{seg}} = \frac{2}{5}$, while the one for which improved results used 3 segments, $\kappa_{\text{seg}} = \frac{1}{5}$. The total number of beam elements remained the same. For $\kappa_{\text{seg}} = \frac{1}{5}$, the non-zero vibration mean in the $y-$ direction has now disappeared.

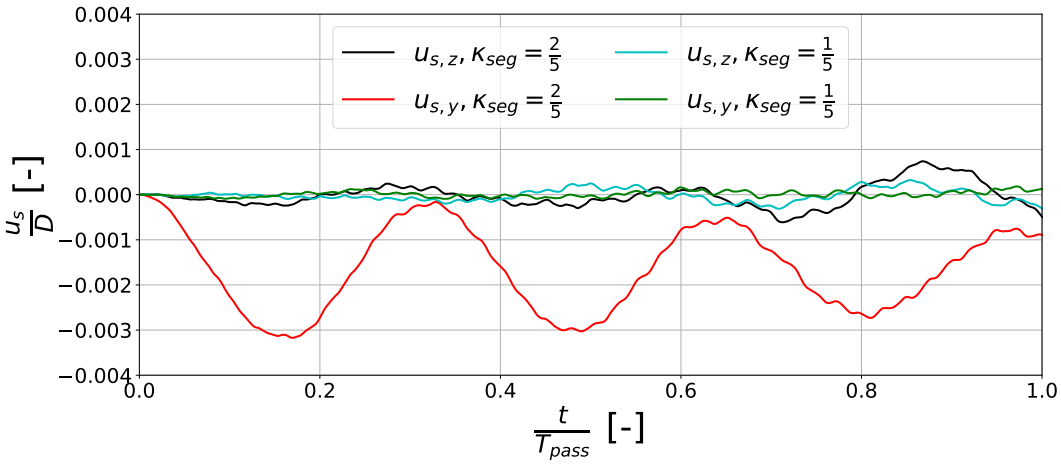


Figure 6.5: The changes in the midpoint displacements of the rod when using 3 segments instead of 6 for the load interpolator, compared to the initial configuration described in Section 6.1, for the NRG-beamFoam workflow, when using an AniPFM model

The two cases presented in Figure 6.5, one for which the offset can be observed, and one for which it cannot, differ based on a key qualitative aspect, as also hinted by the legend in Figure 6.5. For the initial case that produced the offset, 6 segments were used for each beam element, and as a consequence, there were 2 segments used for each 5 fluid elements on the FSI boundary. Since both the segments and the elements are uniform, this implies that the segments' nodes are non-matching with those of the fluid grid. On the other hand, when using 3 segments, a single segment exists for each 5 fluid cells. Thus, the segments are conformal with the fluid grid. The cause of the difference in the displacements obtained for the two cases is not clear. However, it is found that when the axial position of the segments' nodes theoretically corresponds to that of the fluid cell faces' centers, the distribution of the fluid cell faces between the adjacent segments is random. This issue is illustrated in Figure 6.6. This can be attributed to how the connectivity matrix between the fluid faces on the FSI boundary and the segments is obtained. As explained in Subsection 3.4.1.1, this is done by computing the axial position of the fluid cell face's center, and comparing it with that of the segments' nodes. However, based on the round-off error, for the same theoretical axial position, the fluid cell face center may be (slightly) upstream or downstream of the segment node. This hypothesis is confirmed by the results in Tables E.1 and E.2, Appendix E.4, where it can be observed that the number of fluid cell faces assigned to each segment for $\kappa_{\text{seg}} = \frac{1}{5}$ is exactly to equal to 5 times N_{circ}, the number of fluid faces in the circumferential direction, while for $\kappa_{\text{seg}} = \frac{2}{5}$ it is a varying result that is not necessarily an integer multiple of N_{circ}.

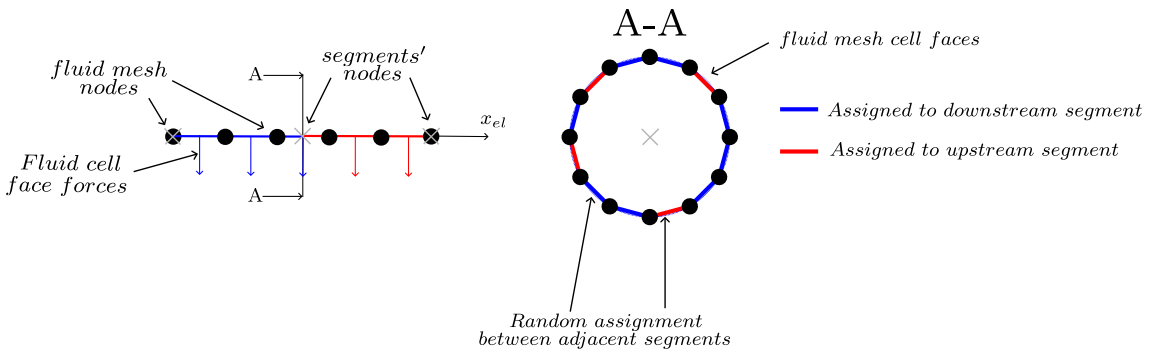


Figure 6.6: The round-off error dependent assignment of the fluid cell face between adjacent segments of the load interpolator for axially matching segments' nodes and fluid cell face centers

Based on the observations made in the last paragraph, it can be concluded that the current implementation of the load interpolator is dependent on the round-off errors of the computation. To mitigate this, in the remainder of this thesis, only discretizations where the segments' nodes are matching with those of the fluid grid will be used. In doing so, the connectivity matrix is uniquely computed, and the area over which a total force is applied is kept equal to that of the segments onto which it is mapped. On the other hand, in the future, as the methodology will be used for more complex configurations, likely, it will also be applied to unstructured fluid grids. There, the matching between the segments' and the fluid nodes would be impossible, making the temporary solution used for the rest of this Chapter inadequate. A better solution could be to map the fluid forces to the segments based on the relative area of their intersection. In the non-matching case presented in Figure 6.5, this would have led to an equal distribution of the loads to both adjacent segments, and hence to a more realistic value of the nodal forces. Due to time constraints, however, this was not explored in this thesis.

It is interesting to note that despite the limitations of the current version of the load interpolator, the workflow was nevertheless accurate for the URANS FSI case. This can be attributed to the URANS test design presented in Subsection 5.4.1, where an initial displacement is applied to the rod. Consequently, throughout the transient of the motion, a large part of the loads is caused by the inertia and the stiffness of the structure, rather than by the fluid load mapping, as is the case for the current Chapter. Despite the limitations of the load interpolator presented in this Subsection, it appears as if the current iteration of NRG-beamFoam is generally accurate in determining the added damping effect.

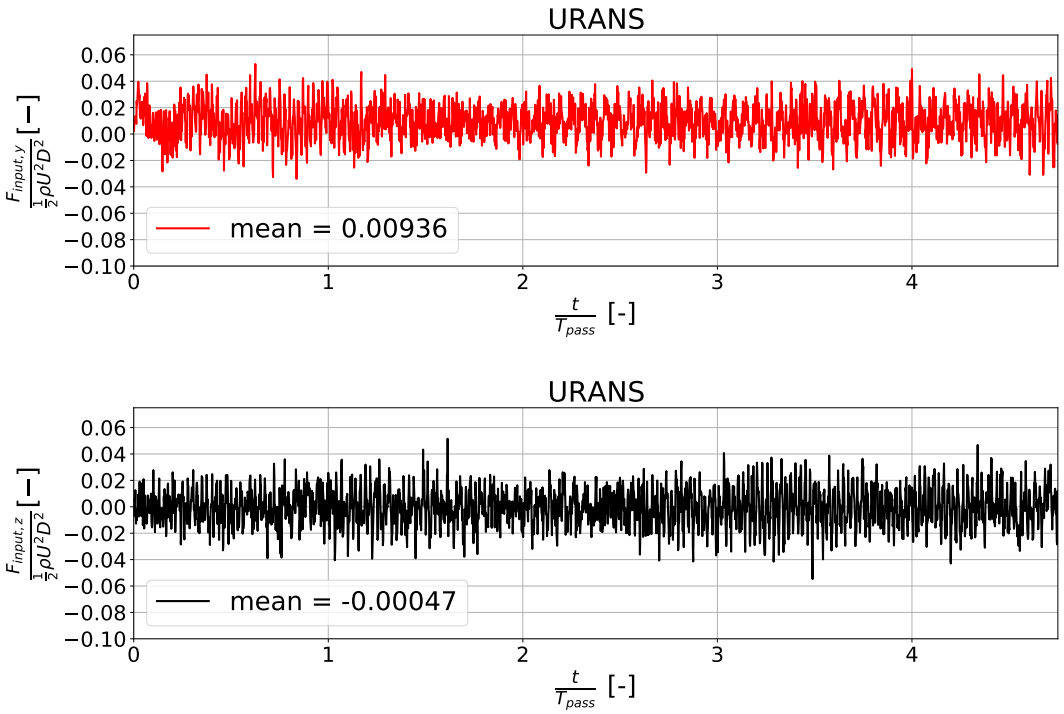


Figure 6.7: The total adimensionalized URANS forces applied on the FSI boundary by the pressure and the viscous stresses of the fluid domain in the $y-$ (top) and the $z-$ (bottom) directions of the beam reference system

To better understand the differences between the two cases in Figure 6.5, the total force received as input by the load interpolator, $\boldsymbol{F}_{\mathbf{input}}$, is calculated by summing up the cell face forces of the fluid domain on the FSI boundary:

$$\boldsymbol{F}_{\mathbf{input}} = \sum_{\text{faces, FSI}} \Big[A \Big(\underbrace{p_{\text{mean}}\mathbf{n} + \boldsymbol{\tau}_{\mathbf{f},\,\text{ik}} \cdot \mathbf{n}}_{URANS} + \underbrace{p'\mathbf{n}}_{PFM} \Big) \Big]_{\text{faces,FSI}}, \tag{6.3}$$

where "faces, FSI" refers to the summation over the fluid cell faces on the FSI boundary, A to their area, p_{mean} and p' to the mean and the fluctuating pressure components, and $\boldsymbol{\tau}_{\mathbf{f},\,\text{ik}}$ to the effective stresses. Based on the different terms in equation (6.3), according to the origin of the fluid model with which they are computed, URANS-based and PFM-based terms can be discerned. For clarity, the two terms of $\boldsymbol{F}_{\textbf{input}}$ will be presented separately. The $y-$ and the $z-$ components of $\boldsymbol{F}_{\textbf{input}}$ generated by the URANS model can be seen in Figure 6.7 for a transient FSI simulation of 0.5 seconds. There, it can be observed that the mean of the forcing imposed by the URANS model in the $y-$ direction is well above zero relative to the amplitude of the forcing, while for the $z-$ direction it is very close to it. The cause of this behaviour is not clear. On the other hand, the AniPFM model gives to the NRG-beamFoam workflow input forces that are close to a zero mean for both directions, as can be seen in Figure E.1, Appendix E.

Having established that the average total forcing received by the NRG-beamFoam model is higher for the $y-$ component for the considered case due to the URANS contributions, it is of interest to observe how these inputs are processed into concentrated nodal forces for the FEM solver by the load interpolator module. To determine this, the total forcing received by the FEM grid, $\boldsymbol{F}_{\textbf{FEM}}$, is obtained by summing all of the different components obtained at the structural grid's nodes. The difference between the total forces received by the FEM solver, and those received as input by the load interpolator is subsequently computed, $\boldsymbol{\Delta F} = \boldsymbol{F}_{\textbf{FEM}} - \boldsymbol{F}_{\textbf{input}}$, where for $\boldsymbol{F}_{\textbf{input}}$ both the URANS and the PFM components in (6.3) are considered. The adimensionalized values of the lateral components of ΔF are presented in Figure 6.8. Compared with Figure 6.7, it can be observed that by using a non-matching grid, the net forcing in the $y-$ direction is amplified five times. By contrast, for the same number of beam elements, the matching case produces only slight changes to the total forcing in the $y-$ direction, despite the lower number of segments used to map the loads. What's more, for both lateral components, a smaller variance of the transient values of $\boldsymbol{\Delta F}$ can be seen when matching segments are used. It can therefore be concluded that compared to a matching segments' discretization, a non-matching one leads to larger instantaneous changes of the total lateral forcing between the structural and the fluid grid, as well as to a higher time-averaged difference between the two.

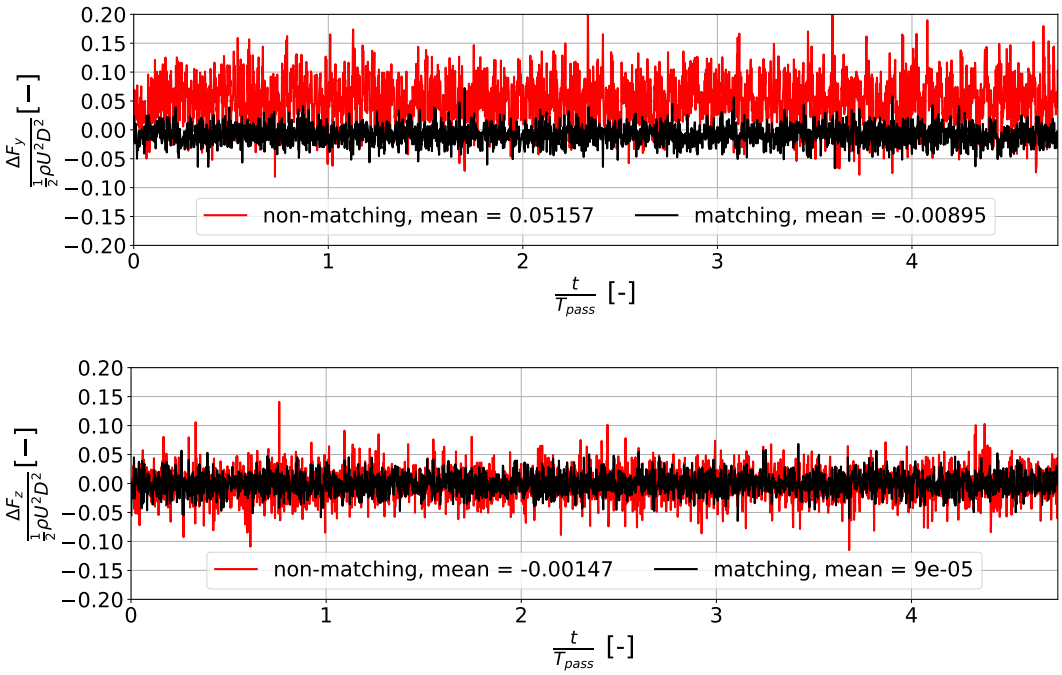


Figure 6.8: The adimensionalized difference in total forcing received by the structural grid, and that imposed by the fluid grid, for the $y-$ and the $z-$ components of the beam reference system

In this Subsection, limitations of the current version of the load interpolator were presented. To isolate the potentially undesirable effect of transforming the loads received by the structure and those applied by the fluid on the FSI boundary, two additional versions of the load interpolator were also implemented for the

remainder of this Chapter. The first could only be used for matching fluid and structural grids on the FSI boundary. The lateral force at each axial location where fluid cell face centers are located, $\boldsymbol{F_{cf}}$, is computed by adding the total forces around the circumference of the FSI boundary:

$$\boldsymbol{F_{cf}} = \sum_{\text{circ}} \left[A \cdot \left(p_{\text{mean}} \mathbf{n} + \boldsymbol{\tau}_{\mathbf{f},\text{ik}} \cdot \mathbf{n} + p' \mathbf{n} \right) \right]_{\text{circ}} , \tag{6.4}$$

where "circ" represents the summation over the different circumferential positions at the same axial location. Subsequently, the nodal structural forces are computed by splitting the total force equally for the two neighboring nodes, and the moments are computed by assuming that the force is applied in the middle of the fluid cell face. Based on this, the following equations can be written:

matching grid load interpolator:

$$F_{1y} = \frac{1}{2} \boldsymbol{F_{cf}} \cdot \boldsymbol{e_y}, \tag{6.5}$$

$$M_{1z} = -M_{2z} = \boldsymbol{F_{cf}} \cdot \boldsymbol{e_y} \cdot \frac{L_{\text{el}}}{2}, \tag{6.6}$$

$$F_{1z} = F_{2z} = \frac{1}{2} \boldsymbol{F_{cf}} \cdot \boldsymbol{e_z}, \tag{6.7}$$

$$M_{1y} = -M_{2y} = -\boldsymbol{F_{cf}} \cdot \boldsymbol{e_z} \cdot \frac{L_{\text{el}}}{2}, \tag{6.8}$$

where $\boldsymbol{e_y}$ and $\boldsymbol{e_z}$ represent the direction vectors of the $y-$ and the $z-$ axes in the beam reference system, $\boldsymbol{e_y}^T = (0\ 1\ 0)$, $\boldsymbol{e_z}^T = (0\ 0\ 1)$. Using the matching grid interpolator, the load mapping is expected to become consistent. This was confirmed by the results presented in Figure E.2, Appendix E, where one can observe the qualitative difference compared to Figure 6.8. The second alternative to the load interpolator was that where the loads are assumed to be uniform *over the entire beam element* for each bending plane. This corresponds to the particular situation where one segment is used for each beam element, and in the following it will be referred to as the uniform load interpolator. For this case, the statically equivalent nodal loads are given by:

uniform load interpolator:

$$F_{1y} = F_{2y} = \frac{\bar{q}_{\text{s},y,\text{i}} \cdot L_{\text{el}}}{2}, \tag{6.9}$$

$$M_{1z} = -M_{2z} = \frac{\bar{q}_{\text{s},y,\text{i}} \cdot L_{\text{el}}^2}{12}, \tag{6.10}$$

$$F_{1z} = F_{2z} = \frac{\bar{q}_{\text{s},z,\text{i}} \cdot L_{\text{el}}}{2}, \tag{6.11}$$

$$M_{1y} = -M_{2y} = -\frac{\bar{q}_{\text{s},z,\text{i}} \cdot L_{\text{el}}^2}{12}, \tag{6.12}$$

where $\bar{q}_{\text{s},y,\text{i}}$ and $\bar{q}_{\text{s},z,\text{i}}$ represent the average distributed loads obtained over the beam element for each lateral direction. The values for the two are obtained by computing a connectivity matrix between the fluid faces and the beam elements using similar rules as for that originally used for the segments, and subsequently using relation (3.13).

6.2.2. Prediction of the vibrations RMS

While the results presented in the last Subsection were intended to ensure that the FSI convergence and a reasonable output are obtained by using NRG-beamFoam, this Subsection intends to determine the accuracy of the newly proposed methodology in determining the amplitudes of the brass beam geometry due to TIV. For this, simulated times in the order of 3-10 seconds are required [38, 41], such that the effects of the initial conditions over the output become negligible, and the RMS value of the vibrations becomes statistically independent. Similar to the previous FSI simulations in this work, it was desired to first generate a reference case for the NRG-beamFoam workflow using the already validated NRG-FSIFOAM workflow. For this, the case of Freitas [41] corresponding to the NRG-FSIFOAM settings presented in Section 6.1 was rerun. However, the same results could not be reproduced. Instead, a significantly lower vibration amplitude was obtained. The comparison between the original results of Freitas [41] and those obtained for this thesis can be seen in Figure E.3, Appendix E. It is not clear what caused this discrepancy, but it is likely an error in how the NRG-FSIFOAM workflow was run for this thesis, since the results of Freitas [41] could be reproduced by other researchers within NRG in the past.

The disagreement between the results of Freitas [41] and those obtained for this thesis imply that the NRG-FSIFOAM workflow cannot be used as a reference for the current Chapter. To circumvent this, a reference case was attempted to be created using the matching grid load interpolator described at the end of Subsection 6.2.1. This should ensure that no potential errors are made when using the load interpolator module,

and that those associated with the displacement interpolator are minimized. Though the matching grid case could be run, a crash occurred after approximately 1.8 seconds of simulated time, as the solution of the non-dimensional velocity fluctuations' transport diverged. To further investigate this behaviour, an additional simulation was launched with the same numerical settings, including the same number of beam elements, but with the uniform version of the load interpolator that was presented at the end of Subsection 6.2.1. Based on the identical computations of the lateral forces for both simulation runs, it was expected that the stability of the FSI solvers would be the same. The midpoint displacements that were obtained for the comparison are presented in Figure 6.9, where it can be observed that although the midpoint displacements are essentially the same, the crash occurred at two slightly different time instances for the two.

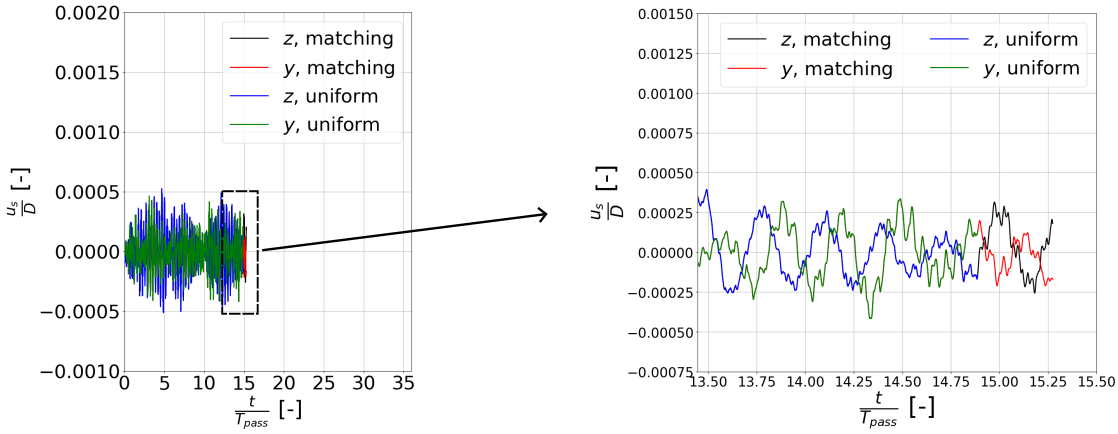


Figure 6.9: The changes in the midpoint displacements of the rod when using the matching and the uniform load interpolators with matching grids on the FSI boundary, for the NRG-beamFoam workflow, when using the AniPFM fluid model

Next, two possible explanations are offered for the crashes at different timesteps for the two similar load interpolator schemes: the crash of the AniPFM model may occur randomly for a given setup, potentially due to the generation of the non-dimensional velocity, or the different moments obtained for the two types of load interpolators play a role in the stability of the workflow. The dependency of the stability of the FSI workflow on the random component of the AniPFM model can be investigated, for example, by using matching grids with different random seed numbers for longer transients, and observing whether significant changes occur for the different considered values. Alternatively, after the solver's divergence, one could also restart the problem from the last written solution using the same random seed, and observe whether the crash occurs at the same time instance. On the other hand, the nodal moments obtained for the structural solver can influence, for example, the smoothness of the FSI boundary, which could in turn affect the stability of the transport equations inside the AniPFM solver due to badly imposed boundary conditions. To observe the curvature of the deformations, the axial distribution of the deformations in the lateral directions that are imposed by the displacement interpolator are presented in Figure 6.10, along with the corresponding distributions of $\varphi_{s,y}$ and $\varphi_{s,z}$, after 0.8 seconds of the simulation. Although the rotations cannot be computed directly since as mentioned in Section 3.2 only the fluid node displacements are passed to the mesh motion solver, the values of the rotations were postprocessed using the kinematic relations (2.62) and (2.82) and the discretized values of the lateral displacements plotted at the top of Figure 6.10. In analyzing this Figure, it can be observed that the distributions of the two methods coincide. Though it is not clear whether this is physically reasonable or not, a potentially non-physical stairstepping aspect of the rotation distributions is observed. Further research is therefore needed to understand whether the rotations of the structure are correctly computed and integrated into the displacement interpolator. Nevertheless, the matching of the fields in Figure 6.10, in combination with the transient information for the midpoint in Figure 6.9, could suggest that the lateral deformations may be the same throughout the transient for both load interpolator methods. However, this is not a conclusion that could already be drawn based on the available information. For this, the lateral displacements must be compared for a time instance that is as close as possible to the crash. Finally, separate from investigating the causes that led to the stopping of the FSI computations, a possible measure to improve the stability is to consider modifying the numerical schemes that are used for the terms associated with the AniPFM model.

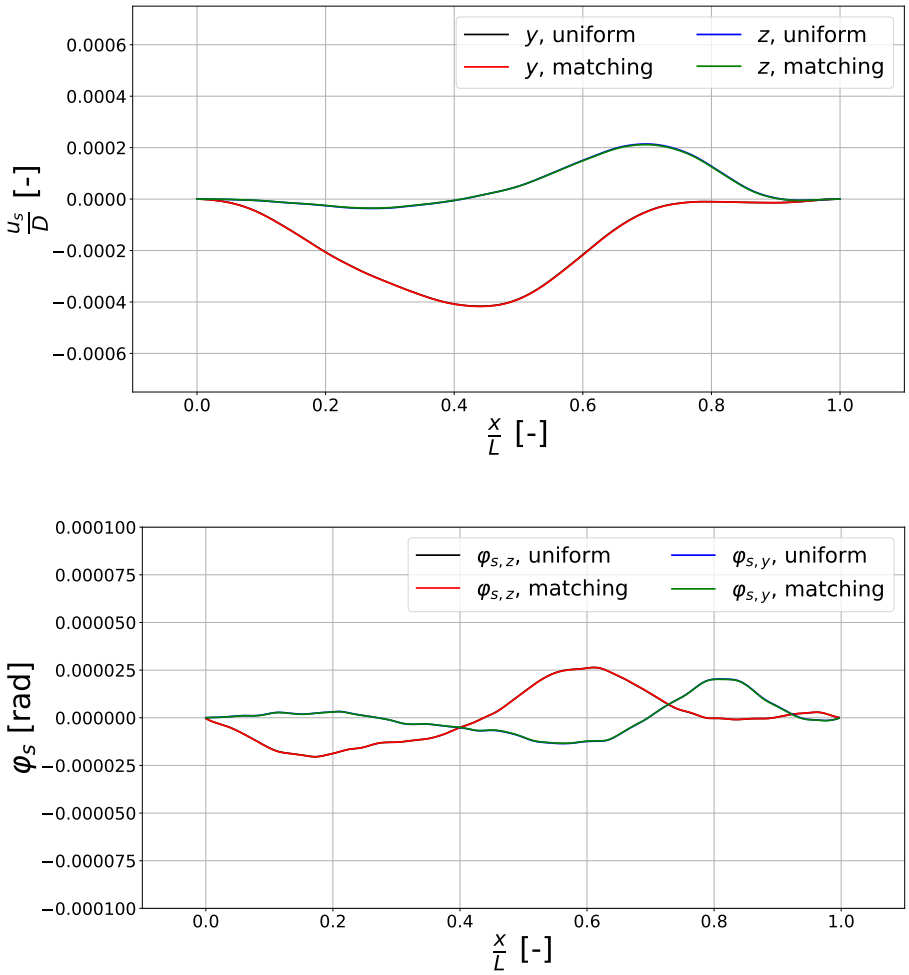


Figure 6.10: Axial distribution of the lateral displacements imposed by the displacement interpolator onto the fluid grid's nodes on the FSI boundary (top), along with the corresponding rotations (bottom), for $\frac{t}{T_{pass}} \approx 6.72$

Having offered some potential hypotheses and research recommendations for the crashes occurring when using the matching grid simulations, the behaviour of the NRG-beamFoam workflow when using an AniPFM fluid model was further investigated by running transient simulations using the uniform and the segments-based load interpolators using different numbers of beam elements. The results are presented in Figure 6.11 and 6.12, respectively. It is generally observed that the stability is better for the segments' based load interpolator. However, for the segments' based load interpolator, there is no correlation between the the number of elements and the stability: the longest simulated time is obtained for 50 elements ($\kappa_{el} = 0.1667$), while for 100 elements ($\kappa_{el} = 0.33$) it is reduced by approximately a third.

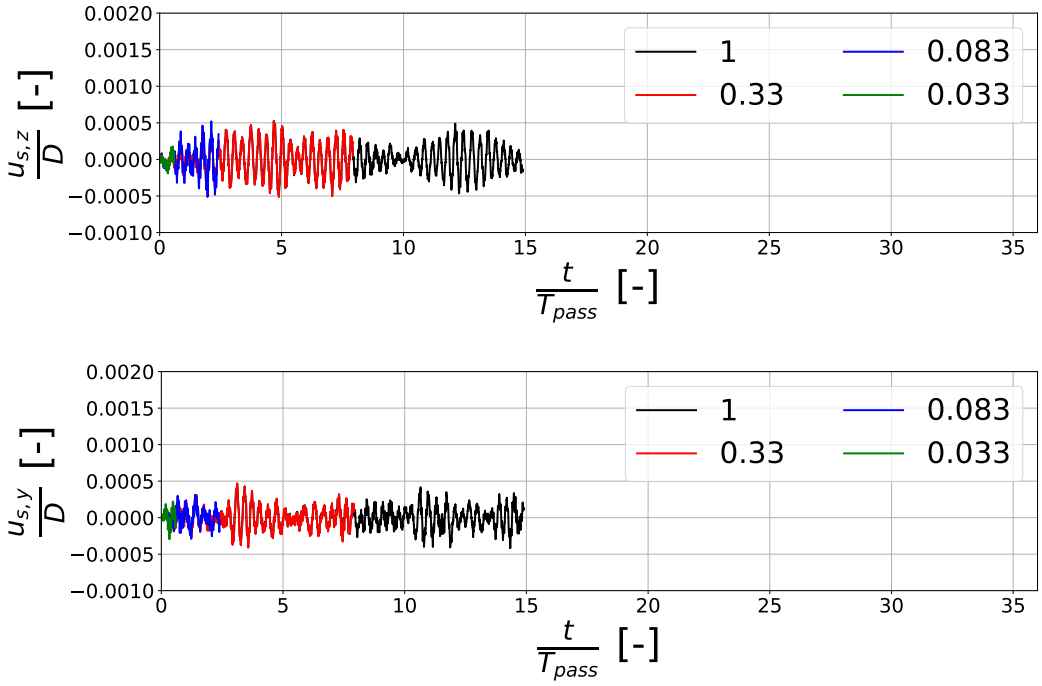


Figure 6.11: The lateral midpoint displacements for the $z-$ (top) and the $y-$ (bottom) directions, when using the uniform implementation of the load interpolator, equations (6.9)-(6.12), for different values of the κ_{el} ratio, equation (6.1)

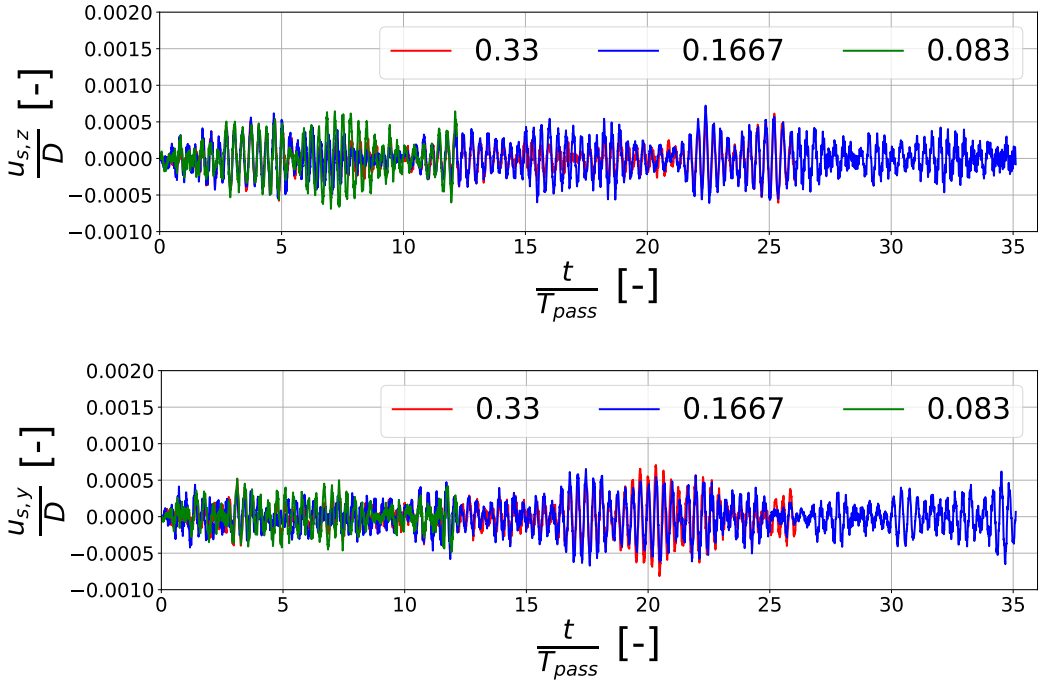


Figure 6.12: The lateral midpoint displacements for the $z-$ (top) and the $y-$ (bottom) directions, when using the the segments' based implementation of the load interpolator presented in Subsection 3.4.1.1, for different values of the κ_{el} ratio, equation (6.1)

7

Verification of the ROM solver used in NRG-beamFoam

The outcome presented in Chapters 4-6 serves as proof of the functioning of the NRG-beamFoam workflow in its most basic form, where a beam-element FEM structural solver is used. However, the ROM literature presented in Section 2.4 suggests that the structural response could potentially be described with an even smaller number of degrees of freedom than those used for the beam-element FEM solver, by using methods such as the Mode Displacement (MD), and the Mode Acceleration (MA), as presented in Subsections 2.4.3.2 and 2.4.3.3, respectively. The main purpose of this Chapter is thus to test this hypothesis for the case when the ROM solver is coupled with the AniPFM fluid solver. Based on the results presented in Subsection 5.4.4, where the FEM solver is observed to have a low computational cost, it is expected that the added benefit of using the ROM solver is not going to be significant for the single-rod test case used for this thesis. Nevertheless, the potential benefits of using a ROM solver can become visible once multiple rod configurations are tested. This small impact on the short-term development of the research is also why the ROM solver is treated in the last part of the project.

Although the main objective of the Chapter is to use the ROM solver in combination with the AniPFM, since the ROM solver needs to be fully implemented in OpenFOAM, it was decided it was necessary to first test it using a simpler, less expensive setup, that is, a laminar FSI case. The settings used and the results obtained when using the ROM solver in a laminar FSI framework are therefore presented in Section 7.1. In Section 7.2, the same is done for the AniPFM solver.

7.1. Initial verification of the ROM solver

This Section is intended to verify the MD and the MA ROM methods in a computationally cheap and simple context. Since the MA essentially represents a correction to the MD ROM, the development of the two methods is done by first ensuring that the MD is correctly implemented, and subsequently using its output as a reference for the MA method. First, the used settings are discussed in Subsection 7.1.1, with corresponding results presented in Subsection 7.1.2.

7.1.1. Case setup and tests description

All of the tests presented in this Section were conducted once all modules of the OpenFOAM architecture in Figure 3.3 were already implemented, except for the `ROMsolver`. As a consequence, for the developed OpenFOAM code, the addition of the ROM structural solver consisted of simply switching off the FEM solver, activating the ROM one, and reusing the same implementation for the rest of the modules. The change between the two structural solvers is done based on the user entries in the `BeamFoamDisplacement` subdictionary in `0/pointDisplacement`, as can also be seen in the example provided in Section B.1, Appendix B.

To be able to present a fair comparison between the two structural solvers, the same settings for the fluid solver were used for both. The same settings as those in Section 5.1 were imposed, along with similar discretization settings for the structure (20 beam elements, 6 segments, trapezoidal time integration scheme).

However, apart from these, to run the ROM solver, it is also necessary to provide the total number of modes that should be used to reduce the structural problem. The number of modes used for each of the tests differs, and will be mentioned in the following for each particular case. For all of these, external nodal forces f_B are applied for different durations, independent of the forcing received from the laminar flow, to generate FSI phenomena.

The first test that was used to verify the ROM solver module is to simply run a laminar FSI case for different sizes of the reduced system, and use the FEM solution as a reference. Thus, the same uniform external force was used as for the FEM case, equation (7.1).

$$\text{For the first ROM test: } f_{B,y}\left(\frac{x}{L},\frac{t}{T_1}\right)=f_{B,z}\left(\frac{x}{L},\frac{t}{T_1}\right)=\begin{cases}f_{ext},\ \frac{t}{T_1}<0.385\\ 0,\ \frac{t}{T_1}>0.385\end{cases}, \tag{7.1}$$

where the initial application time is the same as for the URANS FSI simulations of van den Bos [38], and f_{ext} = 0.654 N represents a constant value that was manually chosen to match the amplitudes presented in the same study.

For the first verification test, only the MD ROM solver is used. Hence, the purpose of this first study is to prove the basic functioning of the MD ROM solver method. In terms of the number of modes used, 1, 6, and 39 were considered for each bending plane. Since both bending planes were active during the laminar FSI simulations, the actual number of total modal coordinates $\boldsymbol{a_s}$ in (2.102) is double those, leading to ratios between the reduced number of degrees of freedom, N_{ROM}, and those of the initial FEM problem, N_{FEM}, equal to $\frac{N_{ROM}}{N_{FEM}} \approx 0.03, 0.16$ and 1, respectively. While the first two number of modes simply represent the ranges of the number of eigenmodes expected to be active during the laminar FSI phenomenon, the last choice represents the particular case where the reduction contains the same number of elements as the initial FEM problem. As a consequence, the MD ROM becomes the Mode Superposition approach, and it is expected that the same structural solution would be obtained as for the FEM solution [81].

The second verification test has the purpose of testing the implementation of the MA ROM method, and at the same time observing the *spatial* reconstruction capability that the ROM methods have. Though still executed in a laminar FSI setting, this test uses a different deformation mechanism of the rod, as can be deduced from equation (7.2).

$$\text{For the second ROM test: } f_{B,y}\left(\frac{x}{L},\frac{t}{T_1}\right)=f_{B,z}\left(\frac{x}{L},\frac{t}{T_1}\right)=\frac{f_{ext}}{2A(\boldsymbol{\phi}_{\mathbf{s},5})}\cdot\boldsymbol{\phi}_{\mathbf{s},5}\left(\frac{x}{L}\right),\ \frac{t}{T_1}<0.385. \tag{7.2}$$

Thus, the initial load is applied so that only the fifth eigenmode of the structure is activated. By contrast, for the uniform load, all eigenmodes were active during the initial application of the force, with some of them damped quickly after the force stops being applied. **To reconstruct the solution, only four eigenmodes are used, that is, one less than the actual excitation force**. It is therefore of interest to observe what the capacity of the ROM solvers is in obtaining the deformations produced by excitation forces outside of the considered eigenspectrum. For the application considered in this thesis, this capability is of interest because although the turbulent excitation has a relatively large bandwidth, a large portion of the energy is stored at the lower frequencies. Thus, it is of interest to observe whether only the eigenmodes that are mostly activated can be used for the reconstruction, and the last bit of energy accounted for by the MA approximation, without having to fully include within the eigenspectrum of the reduced basis.

The third test is intended to verify the ROM's performance in terms of *dynamic* reconstruction, equation (7.3), where the same force is applied in both bending planes, $f_B = f_{B,y} = f_{B,z}$. Both the MD and the MA ROM are active, and the exact same size of the reduction eigenbases is used as for the second test. The spatial shape of the initial load is also unchanged. However, the amplitude of the excitation is now varying in time, where a ratio of 0.3 between the dynamic and the static components is used. Given that a laminar fluid model is used, based on the results presented in Chapter 5, it was expected that little damping would be produced by the FSI physics. Thus, to avoid unstable resonant behaviour, the frequency of the excitation is not equal to the fifth eigenfrequency of the structure, $\omega_{s,5}$, but rather to the arithmetic mean between the fifth and the sixth, $\omega_{s,6}$.

$$\text{For the third ROM test: } f_B\left(\frac{x}{L},\frac{t}{T_1}\right)=\left[1+0.3\cos\left(\frac{\omega_{s,5}+\omega_{s,6}}{2}t\right)\right]\cdot\frac{f_{ext}}{2A(\boldsymbol{\phi}_{\mathbf{s},5})}\cdot\boldsymbol{\phi}_{\mathbf{s},5}\left(\frac{x}{L}\right),\ \frac{t}{T_1}<1.15. \tag{7.3}$$

7.1.2. Results

For the monitors in these three tests, it was decided to use the differences in displacements between the FEM and the ROM solvers that is observed at different time instances. Although in other studies (for example, that of Wilson et al. [112]), the error in the force representation inside the reduced space is used, this metric wouldn't be necessarily relevant for the FSI case considered here, given that the excitations of the higher eigenmodes are damped by the fluid. In other words, for the method considered herein the force representation error is only relevant for the active eigenmodes, instead of the entire band of the excitation forces.

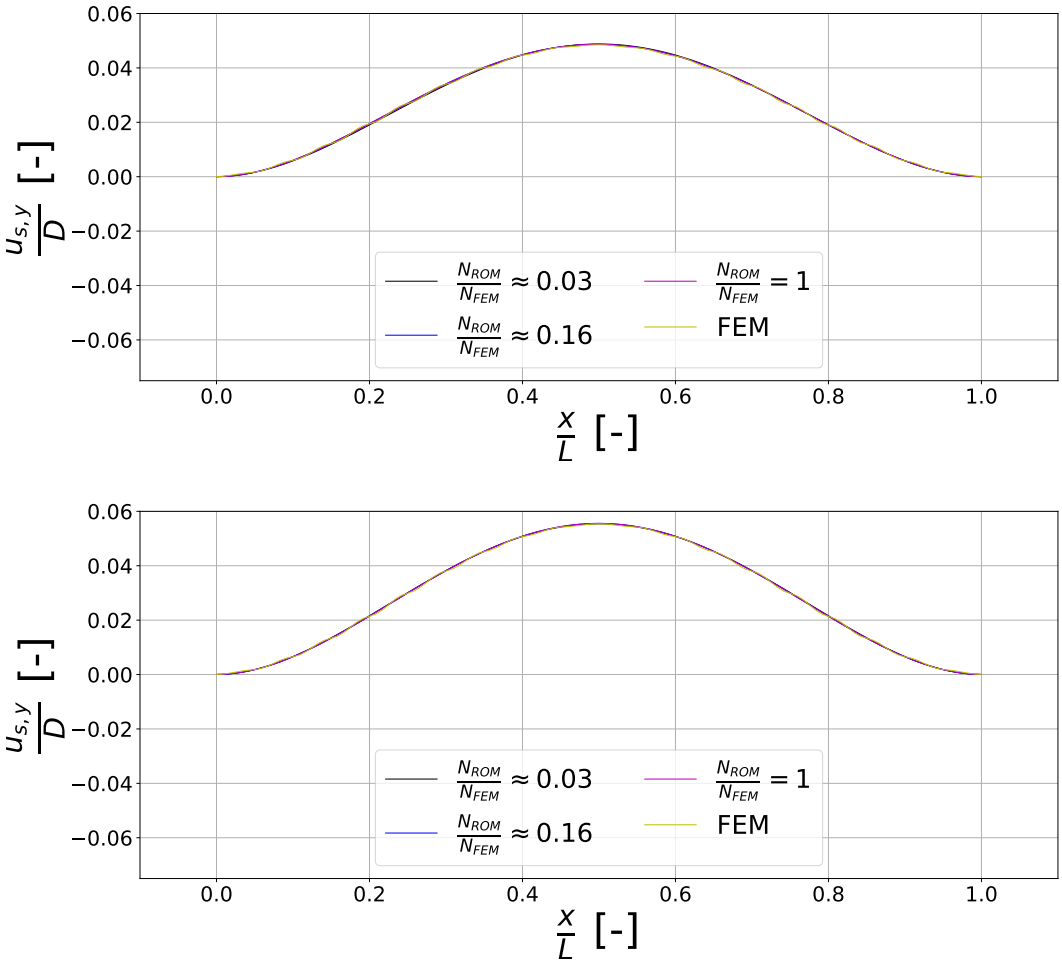


Figure 7.1: Comparison of the axial distributions of lateral displacements in the $y-$ direction obtained within the ROM and the FEM structural solvers for the first ROM verification test, equation (7.1), at $\frac{t}{T_1} \approx 1.28$ (top) and $\frac{t}{T_1} \approx 3.42$ (bottom), considering four modes in each bending plane for the ROM methods, and the full FEM solution

For the first verification test, the results are presented in Figure 7.1, where the instantaneous lateral displacements in the $y-$ direction are presented for the entire brass beam rod at time instances *after* the initial deformation was applied. For all numbers of considered eigenmodes, a good matching between the FEM and the ROM solution is obtained. This shows that the implementation of the MD ROM is fundamentally correct. It is also interesting to note that already for $\frac{N_{ROM}}{N_{FEM}} = 0.03$, that is, for a single eigenmode used for each bending eigenmode to reduce the initial structural problem, a very good matching is obtained with the FEM solution. This can be explained by the inactivity of all modes of the rod during the FSI free vibration, except for the first one. Although the uniform initial external load f_B in (7.1), activates all eigenmodes of the struc-

ture, those are quickly damped out by the fluid once the force stops being applied. Thus, a single degree of freedom is enough to describe the structural response of the rod within the laminar FSI case if using the ROM.

In Figure 7.1, only the results for the XY bending plane are presented. However, it is worth noting that the same behaviour is also observed for the XZ bending plane, as presented in Figure F.1, Appendix F. This shows that the ROM solver is correctly integrated within the two-plane bending formulation of the structural solvers. Consequently, in the remainder of this Chapter results will be presented only for the XY bending plane, keeping in mind, however, that the same also applies to the XZ bending plane.

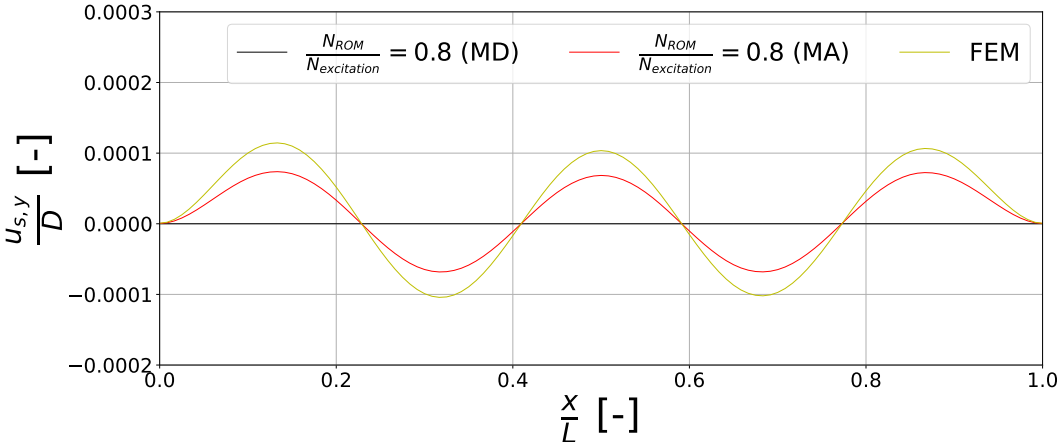


Figure 7.2: Axial distributions of lateral displacements in the $y-$ direction for the second ROM verification test, equation (7.2), at $\frac{t}{T_1} \approx 0.13$ (during the application of the constant-amplitude force)

The results obtained for the second verification test are presented in Figure 7.2, where the same monitor as for the first verification test is used. However, in contrast to the first test, the instantaneous deformations are observed at $\frac{t}{T_1} \approx 0.13$, a time instance where the initial force is still being applied, since based on the outcome in Figure 7.1 it was expected that the fifth eigenmode would be quickly damped after the force stopped being applied. It is worth noting that the FEM solution presents the expected deformation pattern, where four zeros can be observed in the axial direction. This indicates that the initial load is correctly applied to only excite the fifth eigenmode. What's more, it can be seen that the MD ROM is incapable of reconstructing the excitation of the fifth eigenmode when only four eigenmodes are used for each bending plane to reduce the original structural problem. On the other hand, the added quasi-static correction of the MA ROM method enables a correct qualitative estimation of the higher-eigenmode excitation. However, the MA ROM underestimates the amplitude of the deformation of the structural displacements. This can be explained by the fact that within the MA ROM, a quasi-static correction is applied, and the inertial effects of the truncated modes during the application of the initial external force are neglected [103]. On the other hand, the FEM solution does model the inertial effects, leading in this case to a higher amplitude of the deformations. Those findings about the MA ROM are consistent with what was expected based on the literature, and they show that the method was correctly implemented. More broadly, for the context of this work, this second verification test shows that the excitation forces that are outside of the eigenspectrum of the reduced basis, but close to it, can be reproduced spatially.

Apart from the results in Figure 7.2, a further proof of the correct implementation of both the MD and the MA ROM methods is that if 10 modes are used for each bending plane to obtain the decomposition, the same solution as for the FEM is obtained for the fifth eigenmode excitation. This can be seen in Figure F.2, Appendix F. It is also important to note that although this is not visible in Figure 7.1, the MA ROM method has the disadvantage of being less stable than the MD ROM. If fewer than 4 modes were used for the second verification test, it was found that the structural solver immediately crashed. On the other hand, for the MD ROM, stability issues were never encountered. This shows that although the MA is capable of partially reconstructing the higher eigenmode excitation, it still is a linear approximation [99], that is valid only if the eigenbasis that is used for the reduction is close enough to the actual excitations.

In Figure 7.3, the results for the third verification test of the ROM solver are presented. In contrast to the previous two, this time only the displacement of the midpoint is considered, for different time instances. Similar to the previously presented results, the MD ROM is not able to include any of the excitation energy outside of the considered eigenspectrum. The MA, although using the same number of modes as MD, is capable of determining the mean amplitude of the displacements throughout the transient. This can again be expected, since the MA method was introduced as a quasi-static correction for the reduced space of the structural problem [108]. It is also worth noting that the same behaviour that was observed at a particular time instance for the second verification test, that is, the inaccuracy in computing the amplitude of the displacement at a given time instance, can be observed throughout the transient. For the main application considered in this work, this test shows that in the vicinity of the reduction eigenspectrum, apart from the correct deformation shape of the truncated modes at a given time instance, as seen in Figure 7.2, the MA ROM method also reproduces the transient mean of the displacements.

Finally, for the verification conditions of the third test, similar to the second one, it was of interest to determine whether the two considered ROM methods are consistent, that is, provided the excitation is fully contained within the eigenspectrum of the reduced space, the deformations obtained are the same as for the FEM. To verify this, the third verification test was run using 5 eigenmodes for both the MA and the ROM. The results can be seen in Figure F.3, Appendix F, where it can indeed be observed that for both the structural solution converges to that of the FEM problem.

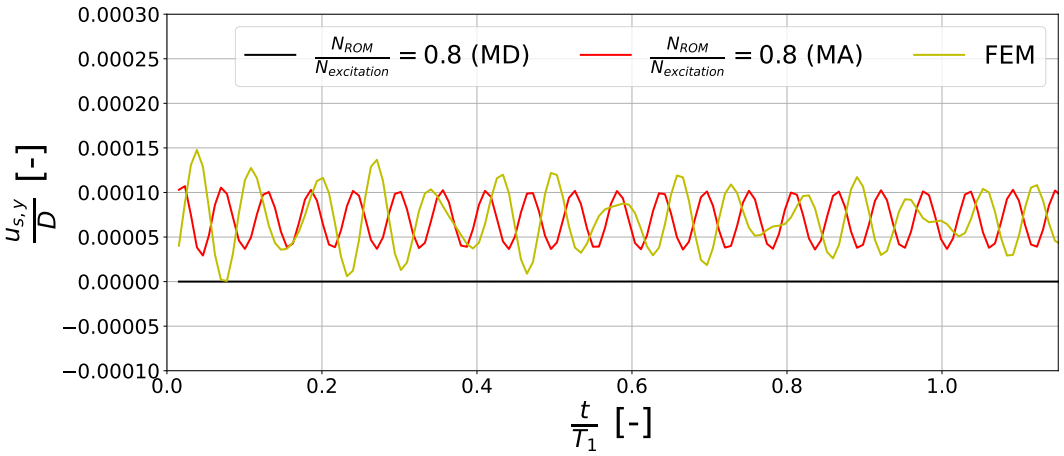


Figure 7.3: Midpoint lateral displacements in the $y-$ direction for the third ROM verification test, equation (7.3), when considering four modes in each bending plane for the ROM method, and the full FEM solution, during the application of the varying-amplitude force

7.2. Verifications using AniPFM-FSI simulations

The results presented in Section 7.1 showed that the MD and the MA ROM behave as expected, and that the ROM solver module is correctly integrated within the NRG-beamFoam workflow. It was therefore decided that it is not necessary to also conduct a URANS FSI case using the ROM, since the only difference compared to the laminar FSI case is the nature of the inputs to the entire workflow, which is independent of the behaviour of the ROM. Instead, for the remainder of this Chapter AniPFM cases will be run, to observe the performance when coupled with the most complex fluid model that is currently intended to be used for axial flow FSI by NRG. Thus, in Subsection 7.2.1, the ROM solver verification tests undertaken with the AniPFM model are presented, with the associated results outlined in Subsection 7.2.2.

7.2.1. Case setup

The results in Section 7.2 indicate that the MA ROM can accurately estimate the mean of the displacements generated by the excitation components whose frequencies are slightly higher than the eigenmodes included in the reduction basis. Thus, compared to the MD ROM, it is expected that an additional benefit of the MA ROM is that the total energy contained within the reduced system is closer to the full FEM system for the same number of modes. Hence, unless specified otherwise, only the MA ROM is used with the AniPFM fluid

model. For consistency, the same settings presented in Section 6.1 are used. For the structural discretization, 100 beam elements and 3 segments are chosen.

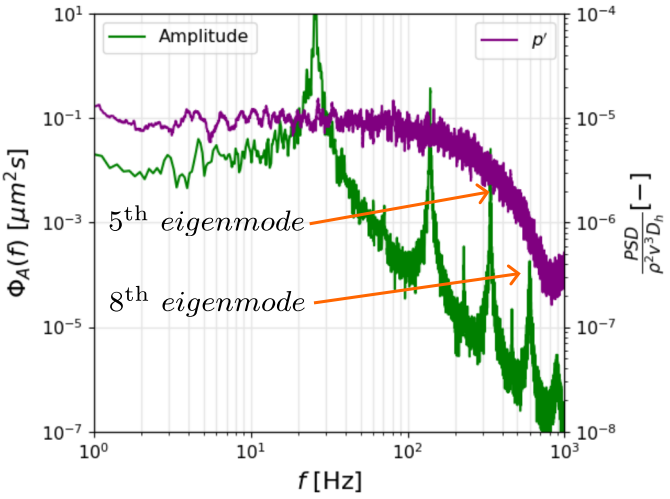


Figure 7.4: The spectral decomposition of the pressure fluctuations and the structural displacements obtained by Freitas [41] using the AniPFM fluid model with the settings considered in this Section; taken from [41]

Next, the number of modes to be used for the ROM is discussed. The modes that are expected to be active during the AniPFM FSI simulation can be seen based on the results of Freitas [41], who for the same fluid and AniPFM settings used in this Section, but using the NRG-FSIFOAM methodology, obtained the spectral decomposition of the structural displacements seen in green in Figure 7.4. Given the quick reduction in vibration amplitude beyond the 5$^{\text{th}}$ eigenmode of the structure, this represents the smallest size of the ROM basis considered, corresponding to a fraction $\frac{N_{\text{ROM}}}{N_{\text{FEM}}} \approx 0.025$ of the original degrees of freedom being contained within the reduced space. To ensure that a significant portion of the entire excitation energy is obtained in the ROM when using 5 modes, the solution was compared to that obtained when using 8 and 15 modes per bending plane $\left(\frac{N_{\text{ROM}}}{N_{\text{FEM}}} \approx 0.04 \text{ and } \frac{N_{\text{ROM}}}{N_{\text{FEM}}} \approx 0.08, \text{ respectively}\right)$. Finally, to check whether the ROM results are consistent with the FEM solver when the same number of degrees of freedom is used, simulations were also undertaken with 198 modes for each bending plane, that is, $\frac{N_{\text{ROM}}}{N_{\text{FEM}}} = 1$. For this third reduction basis choice, it is expected that the same solution as for the FEM solver would be obtained, and this hypothesis serves as a verification of the implementation.

7.2.2. Results

Similar to the outcome presented in Chapter 6, the lateral displacements of the rod's midpoint are first used to assess the performance of the ROM within the FSI workflow. Thus, the lateral displacements obtained in the lateral directions of the structural solver are seen in Figure 7.5 for the different numbers of modes considered. It is worth noting that the simulation using 5 modes for each bending plane is not plotted, because it led to a crash within a few timesteps. This can be simply attributed to using the MA ROM with an eigenspectrum that is too narrow compared to the band of the excitation forces. Nevertheless, already for 8 modes per bending plane, $\frac{N_{\text{ROM}}}{N_{\text{FEM}}} \approx 0.04$, a close matching with the FEM solution can be observed for the transient. What's more, the increase in the number of modes does not appear to significantly modify the midpoint displacements that are obtained. This shows that when using 8 modes, the reduced solution is already close to the FEM output. A difference with respect to the size of the reduced basis can be seen, however, in the stability of the computations, as summarized in Table 7.1. It can therefore be observed that when using the MA ROM, the stability of the FSI computations decreases compared to the FEM structural solver. However, no correlation between the size of the reduced basis and the computational stability could be made, since when using the highest number of modes for the ROM the crash occurs the quickest. The cause of the crash is the same that was also reported in Chapter 6: the divergence of the transport of the AniPFM non-dimensional velocity fluctuations. Thus, the suggestions in Subsection 6.2.2 for better integrating the AniPFM with NRG-beamFoam also apply to the case when the ROM solver is used. It is also important to note that for all of the

numbers of modes considered, the ROM solver slightly underpredicts the amplitude of the vibrations. For $\frac{N_{\mathbf{ROM}}}{N_{\mathrm{FEM}}} = 0.04$ and $\frac{N_{\mathbf{ROM}}}{N_{\mathrm{FEM}}} = 0.08$ this can be attributed to a small quantity of the excitation energy being stored at high frequencies. However, this does not explain the underprediction when the same number of modes are used as the degrees of freedom of the initial FEM problem, since for this it should theoretically be that the same exact solution is obtained [81].

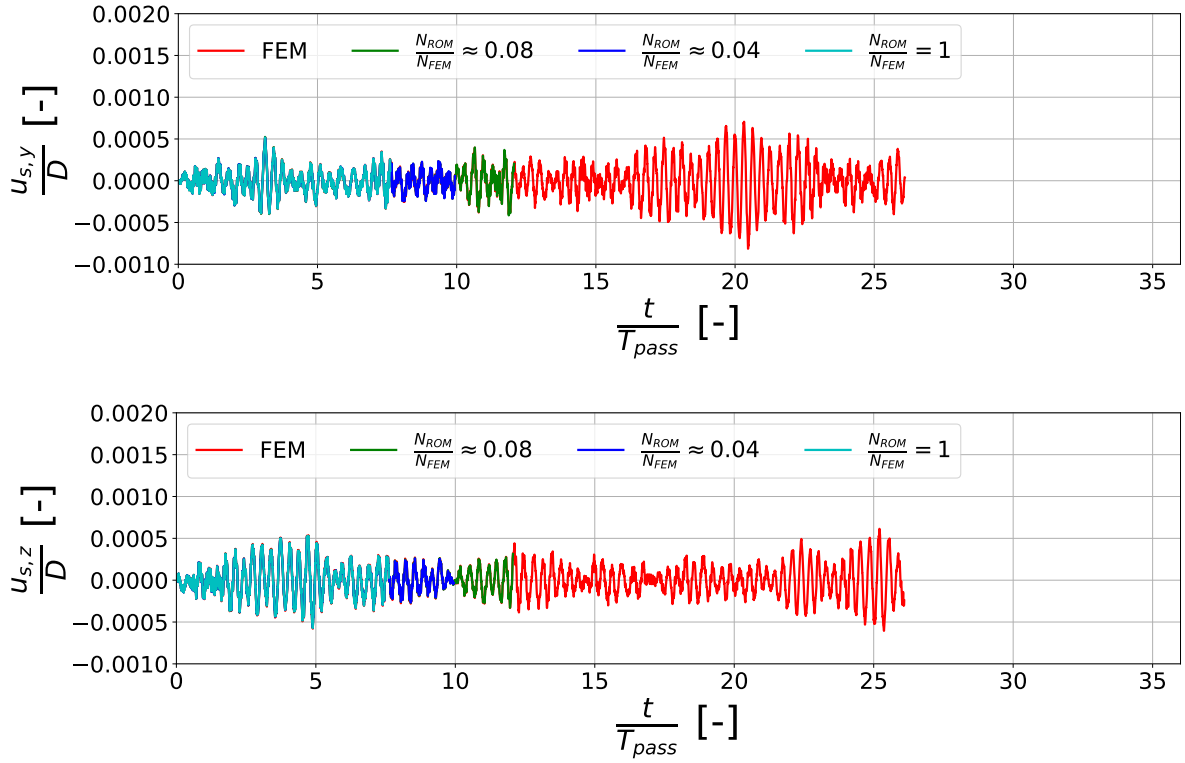


Figure 7.5: Midpoint lateral displacements in the $y-$ (top) and the $z-$ (bottom) directions of the beam reference system that are obtained inside the ROM solver, when using 8, 15, and 198 modes in each bending plane, $\frac{N_{\mathrm{ROM}}}{N_{\mathrm{FEM}}} \approx 0.04, 0.08$ and 1, respectively, relative to the initial FEM formulation, when using the AniPFM fluid model

$\frac{N_{\mathrm{ROM}}}{N_{\mathrm{FEM}}}$ [−]	$\frac{t_{crash}}{T_{pass}}$ [−]
0.04	9.938
0.08	12.069
1	7.620

Table 7.1: The crash times obtained for the AniPFM FSI simulations for the different numbers of modes considered for the ROM solver

Having shown the good matching between the ROM and the FEM solvers for AniPFM FSI simulations when considering the brass beam's midpoint, the discussion is extrapolated by considering the displacements of the entire rod. This is done for three different time instances in Figure 7.6. There, it can be observed that the ROM solver is capable of reproducing relatively complex deformation patterns, even when a small number of modes is used. What's more, a convergent behaviour towards the FEM solution is observed with respect to the number of modes considered, showing the correct working of the ROM implementation. Nevertheless, slight differences in the amplitude of the displacements can be observed, even when all of the modes are used, and when the FEM and the ROM solutions should theoretically overlap. Although the results in this Section only used the MA ROM, the same underestimation of the amplitude of the displacements was also observed when using the MD ROM, Figure F.4, Appendix F. Based on the similar deformation pattern but the different amplitudes obtained when using the ROM solver, a possible explanation for the underprediction of the MA method could be the incorrect scaling of the eigenmodes that are used to reconstruct the solution, especially for the higher eigenmodes. This would not be visible in Section 7.1 since only a small number of

the eigenmodes actively contributed to the solution obtained for each laminar verification case.

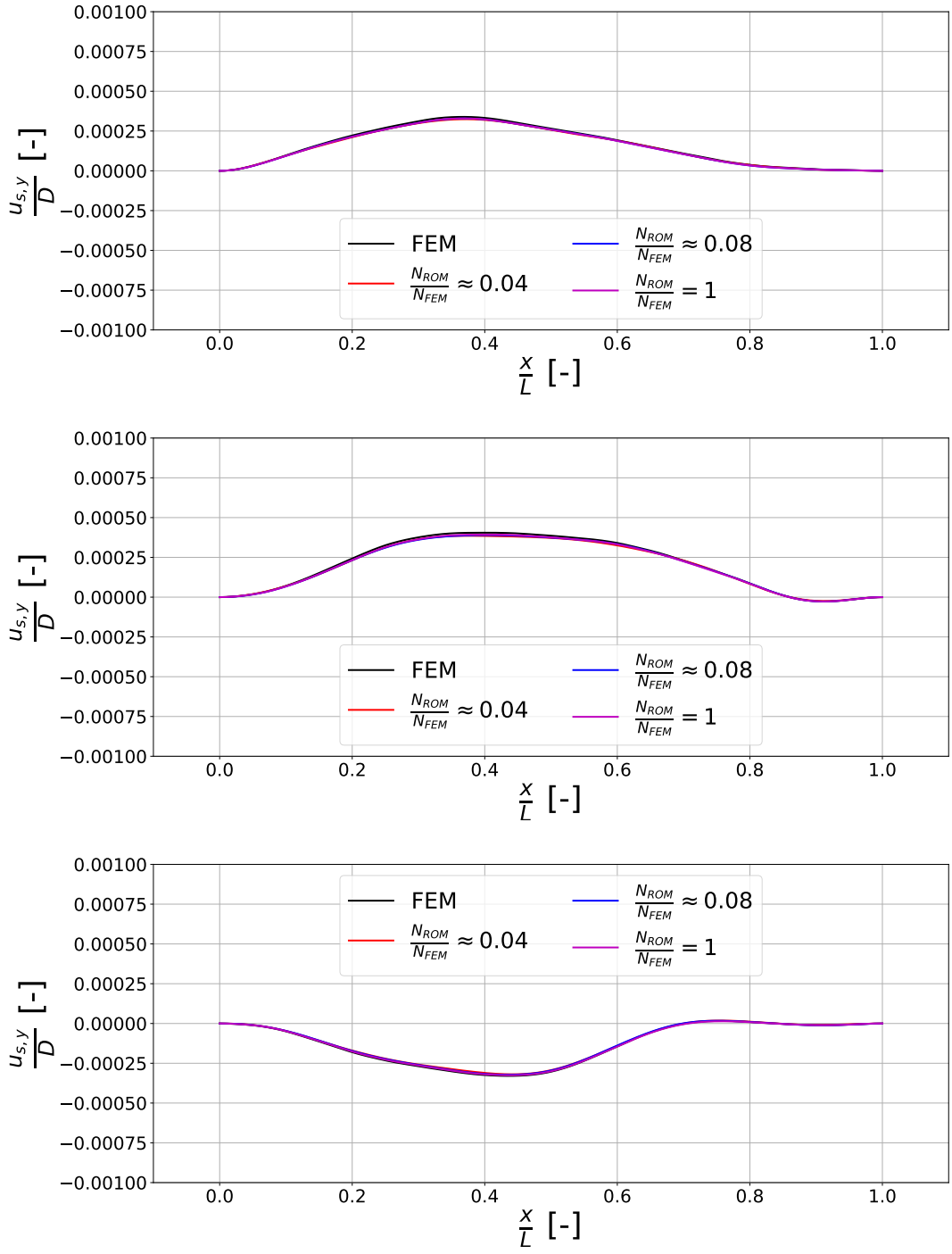


Figure 7.6: Axial distribution of the lateral displacements imposed by the displacement interpolator onto the fluid grid's nodes on the FSI boundary when using different number of modes relative to the total number of degrees of freedom used for the FEM model, for $\frac{t}{T_{pass}} \approx 3.36$ (top), $\frac{t}{T_{pass}} \approx 5.04$ (middle), and $\frac{t}{T_{pass}} \approx 6.72$ (bottom)

8

Conclusions

This last Chapter presents a summary of all of the results obtained in this study (Section 8.1), their correspondence with the research questions (Section 8.2), and the recommendations for continuing the research (Section 8.3).

8.1. Thesis summary

In this thesis, a Mode-Acceleration ROM that simplifies a 1D beam-element FEM solver is proposed as a simple and computationally efficient alternative to the 3D second-order Lagrange FEM solver that was used within the NRG-FSIFOAM workflow for axial-flow FSI inside nuclear reactors. This novel structural solver is used in combination with an AniPFM synthetic turbulence fluid model that was developed by NRG in OpenFOAM. The new structural solver is directly integrated into OpenFOAM, thereby simplifying the complex architecture of the NRG-FSIFOAM workflow, where the structural solution was computed using deal.II, and the coupling with OpenFOAM was ensured by using preCICE. To account for the three-dimensional flow of the fluid over the fuel elements, the structural solver computes the displacements in two separate bending planes of the structure.

To replace both the structural solver and the coupling software, apart from the newly developed structural solver, it was also necessary to implement new mapping routines between the structural grids, as well as a subiteration algorithm for the strong coupling between the two domains. To this end, a load interpolator (mapping the loads from the fluid cell face centers on the FSI boundary to the structural nodes), a displacement interpolator (transferring the nodal structural displacements to the nodes of the fluid mesh on the FSI boundary), and a subiterator module were developed. The combination of the newly developed structural solver, mapping routines, and subiterator, was called NRG-beamFoam.

The load interpolator implementation is based on two steps. First, to compute the 1D distributed loads in each of the two bending directions considered by the structural model, the load interpolator divides each structural element into multiple segments, where for each segment the applied loads are assumed to be uniform. The fluid cell face forces are allocated to each segment based on the axial position of the cell face centers. Secondly, once the distributed loads are known for each bending plane, they are transformed into statically equivalent nodal loads, a mapping approach that is based on preserving the total work done by the external forcing over the structure. Under this methodology, the first step of the load interpolator is consistent, while the second step is inherently conservative.

The displacement interpolator is based on an absolute formulation, where the deformations of the FSI boundary are computed starting from the undeformed configuration of the rod. Its implementation is more straightforward than that of the load interpolator, since the 1D beam-element formulation already assumes the local deformations on the outer boundary of the structure based on the nodal solution and the axial position of the point of interest. The subiterator that is used for the thesis is based on the underrelaxation of the structural displacements, and it uses the Aitken algorithm to solve the fixed-point iteration problem of the FSI convergence.

In terms of implementation, all of these different modules of the workflow are integrated in the PIMPLE algorithm in OpenFOAM. More specifically, all of the NRG-beamFoam routines are contained within a boundary condition for a Laplacian mesh motion solver, and which is prescribed to the patch of the fluid domain that corresponds to the FSI boundary. Thus, to control the number of FSI subiterations, the number of outer correctors for the PIMPLE algorithm needs to be modified. All of the routines are called before the fluid solution at the current subiteration is computed, thereby leading to a F∘S serial FSI scheme.

Under this general methodology, the thesis essentially consisted of verifying and partially validating the functioning of the different components of the NRG-beamFoam workflow, as well as the combined working within FSI numerical simulations with varying fluid modelling complexity (laminar, URANS, and AniPFM). The large majority of these tests were conducted using the experimental setup of Chen and Wambsganss [1], consisting of a flexible single brass beam subjected to turbulent axial water flow in an annular channel.

First, the beam-element FEM solver, the load interpolator, and the displacement interpolator were individually tested in Chapter 4. For the FEM solver, the assembling of the global matrices and the time marching of the structural algorithm were proven using three separate tests, which showed that the output of the module converges to the expected analytical behaviour if the spatial and time discretizations are refined. What's more, it was shown that the accuracy in obtaining the nodal loads quickly converges with respect to the number of segments considered. The accuracy in deforming the fluid mesh is also found to be convergent with the number of beam elements considered for the discretization. The errors introduced by the mapping routines are hence expected to be significantly lower than those associated with the AniPFM model, thereby not decisively affecting the accuracy of the FSI computations.

In Chapter 5, the last essential module of the newly implemented FSI loop, the subiterator, was tested. It is found that the Aitken implementation gives the same results as a fixed URF subiteration scheme, but it significantly reduces the number of solver calls that need to be executed. Thus, the Aitken algorithm is conducive to efficient computations of FSI problems by the NRG-beamFoam workflow. With all of the modules now verified, FSI simulations using laminar and URANS fluid models were run for the brass beam case, where an initial deformation is applied to the structure, after which the rod is allowed to vibrate freely inside the 10 m/s water flow. It was found that the NRG-FSIFOAM and the NRG-beamFoam present similar frequency and damping ratios for the beam's vibration, where the relative difference between the two methodologies is within 1.5% for both monitors. What's more, the newly implemented structural solver and mapping routines reduce the computational costs of URANS simulations per FSI subiteration by 48% compared to the latest structural settings recommended when using NRG-FSIFOAM [38]. If one assumed that the structural solver were faster than the fluid one for the NRG-FSIFOAM methodology, the NRG-beamFoam approach would nevertheless lead to a 6% in computational cost savings per FSI subiteration, given the small dimensionality and the simplified mapping routines. Nevertheless, the current implementation of NRG-beamFoam uses a less efficient subiteration algorithm and FSI convergence criterion than NRG-FSIFOAM, leading to a total number of FSI subiterations that is approximately a third higher than its counterpart.

Chapter 6 presents the NRG-beamFoam FSI simulations that were attempted to be run in combination with the AniPFM fluid model. It is found that the AniPFM routines can be called only once per timestep, without incurring losses in output accuracy. This represents an advantage over the current NRG-FSIFOAM workflow, where the AniPFM had to be called at every subiteration, and was one of the main contributors to the computational cost of the method [38]. Furthermore, it is shown that for TIV simulations, the current version of the load interpolator may present non-physical results if the nodes of the segments' discretization do not match with those of the fluid grid. To mitigate this, FSI simulations with matching fluid and structural solver grids were run. However, it is found that those inevitably led to premature crashes. What's more, no clear link between the structural discretization and the stability of the FSI computations could be formulated. It is therefore not yet clear whether this is caused by the AniPFM model, or by the NRG-beamFoam workflow.

Finally, in Chapter 7, the ROM solver was individually verified, and subsequently used to compute AniPFM FSI simulations. It is found that the behaviour of the ROM solver is congruent with that of the FEM solver for both laminar and AniPFM FSI cases. For the former, only one mode per bending plane is required to compute the free vibration of the rod, while for the latter, approximately 8 modes are needed for each bending plane. Despite these positive outcomes, and the qualitative similarity between the FEM and the ROM solver, it is

observed that the amplitude of the vibrations is currently slightly underestimated when using the ROM, even if the same number of degrees of freedom as for the full FEM system is used. What's more, the same stability issues are observed when using the ROM solver in combination with the AniPFM as those found in Chapter 6.

8.2. Answering the research questions

In Chapter 1, a series of questions was posed to guide this research. These are reproduced and answered here:

1. **Can the newly implemented structural solvers be used along with the newly developed mapping routines and subiteration algorithm for two-way FSI simulations? What is the accuracy of the ensuing workflow, compared to NRG-FSIFOAM for such a scenario?**

The outcome of Chapter 5 offers clear answers to these questions. There, it is shown that when running FSI simulations using a laminar fluid model, a very similar performance to that of the previously validated NRG-FSIFOAM workflow is obtained. More specifically, the relative differences in the frequency and damping ratio for the laminar case are 0.6% and 1.3%. This shows that the general architecture of the NRG-beamFoam workflow is sound, and it proved at the time that the methodology could be used for more complex fluid models, thereby leading to the second research subquestion:

2. **What is the accuracy of the new FSI solver in computing axial-flow FSI when using a URANS fluid model? Are there any computational benefits compared to the NRG-FSIFOAM workflow?**

Here, the results presented in Chapter 5 can again be used. There, by running similar URANS cases for NRG-FSIFOAM and NRG-beamFoam, good similarity is found between the frequency and the damping ratios that are obtained, with a relative difference equal to 0.5% for both monitors. However, since the NRG-beamFoam uses both a significantly smaller number of degrees of freedom to describe the structural response, and also mapping routines specialized for 1D structural solvers, it is found that only 0.55% of the total computational time is spent on obtaining the structural solution and coupling the FEM solver to the fluid one. By contrast, the fluid solver in NRG-FSIFOAM spends 48.9% of the total time waiting for the structural solution to be computed and mapped. If the structural solution were computed quicker than the fluid within NRG-FSIFOAM, given its use of a parallel subiteration scheme, then for the comparison with the total time spent on NRG-beamFoam, one would need to consider only the time that it takes to map the structural solution to the fluid domain. In doing so, it is found that 6.7% of the processing time is spent on mapping the structural solution to the fluid solver. This is a share that is still significantly larger than that associated with NRG-beamFoam. This finding proves the computational benefit that the NRG-beamFoam workflow can bring, apart from the advantage of simplifying the entire workflow by using an approach that is fully integrated in OpenFOAM, instead of the three separate pieces of software used for the NRG-FSIFOAM (OpenFOAM, deal.II, and preCICE). The good performance of the NRG-beamFoam workflow for URANS fluid models meant that the next step in the verification could be taken, namely, the determination of the FSI performance when used with the AniPFM. This led to the answer of the third question:

3. **What are the preferential settings that should be used for the novel FSI solver when combined with the AniPFM model?**

The FSI study of the NRG-beamFoam in combination with the AniPFM model was hampered by stability issues, as was already pointed out in Chapters 6 and 7. Nevertheless, valuable conclusions for future research regarding the setup could still be drawn. First, in Chapter 6, it was found that the AniPFM fluid model can be used only once per timestep, without any significant loss in accuracy, thereby reducing a significant portion of the computational time that would be spent on computing the fluid solution. What's more, when using NRG-beamFoam in the immediate future, it is recommended that the segments' discretization being used for the load interpolator is made matching with the fluid grid to avoid the local reordering of the distributed loads. Even better, it is suggested that matching grid simulations be run, to avoid any potential errors introduced by the load interpolator, until the other issues identified are solved. As for the ROM solver, in Chapter 7, using 8 modes per bending plane appears to give a very good accuracy of both the amplitude of the vibrations, as proven by the midpoint of the clamped rod, and of the general deformation pattern of the structure at a given timestep. This is observed for time instances where the perturbation of the initial conditions over the FSI problem has faded away.

8.3. Further research

Although the NRG-beamFoam methodology has shown promising results, it is also clear that its development is far from complete. The recommendations for the future development of the NRG-beamFoam can be split into four categories: verification, optimization, validation, and extrapolation. Those will be discussed separately in the following.

First of all, the verification steps that need to be taken in the future can be mostly tied to the issues observed in Chapters 6 and 7: the instability of the AniPFM when coupled with the NRG-beamFoam workflow, the reduction in accuracy of the load interpolator when the segments' nodes are non-matching with the structural ones, the inability to reproduce the previous AniPFM results obtained by Freitas [41], and the underestimation of the vibration's amplitudes when using the ROM solver with the AniPFM. It is believed by the author that ensuring stability when running the AniPFM with NRG-beamFoam is of paramount importance, and therefore specific solutions to tackle this problem are discussed, as also mentioned in Chapter 6:

- studying the influence of the random generator of non-dimensional velocity fluctuations over the stability of the AniPFM when coupled with NRG-beamFoam by using multiple random seeds and restarting the computations after the crash has occurred; this is justified because the crashes appeared to be somewhat arbitrary, not depending, for example, on the level of structural discretization used for the beam element FEM solver
- analyzing the accuracy of the structural rotations imposed onto the FSI boundary by the NRG-beamFoam workflow, and their impact on the stability of the problem; this is beacause a slightly different stability was observed when using the matching grid interpolator, equations (6.5)-(6.8), and the uniform load interpolator, (6.9)-(6.12); the two only differ based on how the nodal moments are computed
- using different discretization schemes for the transport equations of the AniPFM, given that the crashes always occurred when the convection of the non-dimensional velocity fluctuations was computed

If none of these options are found to work, then the NRG-beamFoam could be coupled with a coarse LES fluid model, that provides a similar highly-fluctuating fluid forcing on the FSI boundary, but is better validated than the AniPFM. A coarse LES FSI test could therefore reveal underlying problems of NRG-beamFoam that are leading to the instability of the AniPFM.

Apart from the topic of coupling the AniPFM with NRG-beamFoam, the results in Chapter 6 clearly showed the current intrinsic limitations of the proposed load interpolator, where the load mapping may give non-physical results if the fluid forcing is randomly distributed between the different segments. To mitigate this, it is recommended that a load interpolator where the fluid forces assigned to each segment are dependent on the intersection area between the fluid solver and the hypothetical outer surface of the segment. Alternatively, one could also use a uniform load interpolator, as presented in equations (6.9)-(6.12), to map the fluid loads to the structural nodes, although this may incur a slight increase in the number of elements required for the NRG-beamFoam methodology, and similar behaviour as presented in Figure 6.6 if the FEM nodes and the fluid cell face centers coincide. In its current form, the potentially negative effects of the load interpolator appear to be neutralized if axially matching grids are used for the fluid and the structure on their boundary, though this temporary fix wouldn't work for unstructured grids, which are expected to be used when the NRG-beamFoam methodology is used for more complex configurations. Finally, for the identified ROM solver issues, it is recommended that the scaling and the orthogonality of the eigenmodes be verified.

Next, the measures that could be taken to optimize the NRG-beamFoam workflow are discussed. First, it is expected that a significant computational benefit would be obtained if the FSI subiterations were controlled by using a convergence criterion that is based on the residuals of the nodal displacements that are obtained at consecutive subiterations. At the moment, only a fixed number of subiterations can be used, equal to the number of outer correctors imposed for the PIMPLE loop. To implement this criterion, it is likely required that the `pimpleControl` class in OpenFOAM is modified. Additionally, the FSI convergence may be improved by implementing an IQN-ILS subiteration algorithm, as used by NRG-FSIFOAM. Currently, a potentially less efficient Aitken algorithm is implemented. An interesting direction of the research could also be to attempt to use a subiteration algorithm that is based on the underrelaxation of the fluid loads at the FSI boundary, rather than that of the structural displacements, as it is currently being done. This measure is proposed because an

underrelaxation of the loads may bring faster convergence than using the displacements.

Additional validation work is also required before NRG-beamFoam becomes an established FSI solver for modelling axial flow in nuclear reactors. The tests in this thesis were only conducted using the brass beam geometry, at 10 m/s. The next step in the validation process could be to analyze the results obtained at other axial flow velocities for the same experiment, and subsequently to observe the performance when using more complex configurations, containing multiple rods.

Finally, the NRG-beamFoam workflow could also potentially be extrapolated for other FSI applications. An immediate example is the use of a more complex physical model of the rod, where the physical properties of the rod vary in the axial direction based on the local content of nuclear fuel. The structural damping could also be included, to observe the effects over the rod's vibrations. In principle, the NRG-beamFoam workflow can also be used for other, more distinct applications. In this regard, the only constraints for using NRG-beamFoam are the use of the PIMPLE algorithm, and the assumption that the structure is predominantly subjected to lateral bending for which the corresponding lateral displacements are small. Immediate potential modifications could refer to the support conditions of the rod: at the moment, the workflow only supports the modelling of clamped-clamped beams, but quick modifications could lead to the use of the methodology for other end conditions. Finally, the same workflow could in principle be used for thin-walled beams, if, for example, a Timoshenko beam formulation is used for the structural solver module, with the rest of the workflow remaining the same.

Bibliography

1. Shoei-sheng Chen & Wambsganss, M. W. Parallel-flow-induced vibration of fuel rods. *Nuclear Engineering and Design* **18,** 253–278. ISSN: 0029-5493 (1972).
2. International Energy Agency. *CO2 emissions by product and flow* IEA CO2 Emissions from Fuel Combustion Statistics (database). 2012.
3. Foster, V. & Bedrosyan, D. *Understanding CO2 emissions from the global energy sector* tech. rep. 2014/5 (World Bank Group, Washington, D.C., 2014).
4. International Energy Agency. *World Energy Outlook 2022* (IEA, Paris, 2022).
5. Intergovernmental Panel on Climate Change. *Climate Change 2022: Mitigation of Climate Change. Contribution of Working Group III to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change* (Cambridge University Press, Cambridge, UK and New York, NY, USA, 2022).
6. International Energy Agency. *Global Energy and Climate Model* Paris, 2022. `https://www.iea.org/reports/global-energy-and-climate-model`.
7. Pehl, M., Arvesen, A., Humpenöder, F., Popp, A., Hertwich, E. G. & Luderer, G. Understanding future emissions from low-carbon power systems by integration of life-cycle assessment and integrated energy modelling. *Nature Energy* **2,** 939–945 (2017).
8. Bolson, N., Prieto, P. & Patzek, T. Capacity factors for electrical power generation from renewable and nonrenewable sources. *Proceedings of the National Academy of Sciences* **119,** e2205429119 (2022).
9. Markandya, A. & Wilkinson, P. Electricity generation and health. *The lancet* **370,** 979–990 (2007).
10. Sovacool, B., Andersen, R., Sorensen, S., Sorensen, K., Tienda, V., Vainorius, A., Schirach, O. & Bjørn-Thygesen, F. Balancing Safety with Sustainability: Assessing the Risk of Accidents for Modern Low-Carbon Energy Systems. *Journal of Cleaner Production* **112** (Mar. 2015).
11. Kim, Y., Kim, M. & Kim, W. Effect of the Fukushima nuclear disaster on global public acceptance of nuclear energy. *Energy Policy* **61,** 822–828. ISSN: 0301-4215 (Oct. 2013).
12. European Commission. *Special Eurobarometer no.297: Attitudes towards radioactive waste* June 2008. `https://www.iaea.org/sites/default/files/ebs%7B%5C_%7D297%7B%5C_%7Den.pdf`.
13. Papukchiev, A., Zwijsen, K., Vivaldi, D., Hadžić, H., Benhamadouche, S., Benguigui, W. & Planquart, P. *The European GO-VIKING Project on Flow-Induced Vibrations* in *Proceedings of the 20th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-20)* (Washington, D.C., USA, 2023).
14. Cioncolini, A., Zhang, S., Nabawy, M. R. A., Li, H., Cooper, D. & Iacovides, H. Experiments on axial-flow-induced vibration of a free-clamped/clamped-free rod for light-water nuclear reactor applications. *Annals of Nuclear Energy* **190,** 109900 (2023).
15. Wang, H., Hu, Z., Lu, W. & Thouless, M. D. The effect of coupled wear and creep during grid-to-rod fretting. *Nuclear Engineering and Design* **318,** 163–173 (2017).
16. Kim, K. & Suh, J. M. Impact of Nuclear Fuel Assembly Design on Grid-to-Rod Fretting Wear. *Journal of Nuclear Science and Technology* **46,** 149–157. ISSN: 0022-3131 (Feb. 2009).
17. Kim, K. T. The study on grid-to-rod fretting wear models for PWR fuel. *Nuclear Engineering and Design* **239,** 2820–2824. ISSN: 00295493 (Dec. 2009).
18. Kim, K. T. A study on the grid-to-rod fretting wear-induced fuel failure observed in the 16×16KOFA fuel. *Nuclear Engineering and Design* **240,** 756–762. ISSN: 00295493 (Apr. 2010).
19. Lazarevic, S., Lu, R. Y., Favede, C., Plint, G., Blau, P. J. & Qu, J. Investigating grid-to-rod fretting wear of nuclear fuel claddings using a unique autoclave fretting rig. *Wear* **412-413,** 30–37. ISSN: 00431648 (Oct. 2018).
20. Liu, P., Li, F., Chen, B. & Zhang, S. Theoretical investigations on flow-induced vibration of fuel rods with spacer grids in axial flow. *Annals of Nuclear Energy* **133,** 916–923. ISSN: 18732100 (Nov. 2019).

21. Kim, K. T. The effect of fuel rod supporting conditions on fuel rod vibration characteristics and grid-to-rod fretting wear. *Nuclear Engineering and Design* **240,** 1386–1391. ISSN: 00295493 (June 2010).
22. Hu, Z., Thouless, M. D. & Lu, W. Effects of gap size and excitation frequency on the vibrational behavior and wear rate of fuel rods. *Nuclear Engineering and Design* **308,** 261–268. ISSN: 00295493 (Nov. 2016).
23. Païdoussis, M. P. & Besancon, P. Dynamics of arrays of cylinders with internal and external axial flow. *Journal of Sound and Vibration* **76,** 361–379. ISSN: 0022-460X (June 1981).
24. Païdoussis, M. P., Suss, S. & Pustejovsky, M. Free vibration of clusters of cylinders in liquid-filled channels. *Journal of Sound and Vibration* **55,** 443–459. ISSN: 0022-460X (Dec. 1977).
25. Shoei-sheng Chen. Vibration of nuclear fuel bundles. *Nuclear Engineering and Design* **35,** 399–422. ISSN: 0029-5493 (Dec. 1975).
26. Chen, S. S. & Jendrzejczyk, J. A. Experiments on fluidelastic vibration of cantilevered tube bundles. *J. Mech. Des.; (United States)* **100** (July 1978).
27. Gagnon, J. O. Flow-induced vibration of cylinder cluster in incompressible confined axial flow (1982).
28. Ricciardi, G. & Boccaccio, E. *Stiffening of a fuel assembly under axial flow* in *Proceedings of the 10th International Conference on Flow-Induced Vibration (& Flow-Induced Noise)—FIV2012, Dublin, Ireland* (2012), 2–6.
29. Dressel, B., Staude, U., Hofbeck, S., Münch, C.-J. & Kovacs, S. *Investigations on the FSI characteristic of a scaled and integral fuel structure under axial flow* in *ASME International Mechanical Engineering Congress and Exposition* **58370** (2017), V04AT05A038.
30. Encyclopædia, B. *Pressurized-Water Reactor* `https://www.britannica.com/technology/nuclear-reactor/Types-of-reactors#/media/1/421763/160852`.
31. Isolation, D. *Fuel Asesembly* `https://www.deepisolation.com/about-nuclear-waste/what-is-spent-nuclear-fuel/`.
32. NRG. *Fuel Creep* 2023. `https://www.ensuringnuclearperformance.com/en/products/irradiation-services/fuel-irradiation/fuel-creep`.
33. Hu, Z. Developments of analyses on grid-to-rod fretting problems in pressurized water reactors. *Progress in Nuclear Energy* **106,** 293–299 (2018).
34. Anderson, J. Computational Fluid Dynamics: The Basics with Applications. 1995. *McGrawhill Inc.* ISSN: 1520-6033 (1995).
35. Bengtsson, K. *FSI-analysis on vibrations of a slender rod exposed to axial flow, Calculations for nuclear power aplications (MSc Thesis)* Göteborg, Sweden, 2016.
36. Papukchiev, A. *Numerical prediction of flow-induced vibrations in reactor relevant geometries* in *Proceedings of the 8th International Conference on CFD in Nuclear Reactor Safety Engineering and Environmental Application* (Nov. 2020).
37. Matozinhos, C. F. & Hassan, Y. Scale-resolving simulations of the flow in a nuclear fuel bundle with a channel spacer grid using partially averaged Navier–Stokes and large-eddy simulation. *Physics of Fluids* **35,** 015115. ISSN: 1070-6631 (Jan. 2023).
38. Van den Bos, N. *Turbulence-Induced Vibrations Prediction: Through Use of an Anisotropic Pressure Fluctuation Model (MSc Thesis)* Delft, Apr. 2022.
39. Kottapalli, S., Shams, A., Zuijlen, A. H. & Pourquie, M. J. Numerical investigation of an advanced U-RANS based pressure fluctuation model to simulate non-linear vibrations of nuclear fuel rods due to turbulent parallel-flow. *Annals of Nuclear Energy* **128,** 115–126. ISSN: 0306-4549 (June 2019).
40. Bos, N. v. d., Zwijsen, K., Zuijlen, A. H. v., Frederix, E. M. & Roelofs, F. Turbulence-induced vibrations prediction through use of an anisotropic pressure fluctuation model. *EPJ Nuclear Sciences & Technologies* **9,** 7. ISSN: 2491-9292 (2023).
41. Freitas, A. *Numerical study on turbulence induced vibrations of fuel rods (MSc Thesis)* Delft, Oct. 2023.
42. Weller, H. G., Tabor, G., Jasak, H. & Fureby, C. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in physics* **12,** 620–631 (1998).

43. Arndt, D., Bangerth, W., Feder, M., Fehling, M., Gassmöller, R., Heister, T., Heltai, L., Kronbichler, M., Maier, M., Munch, P., Pelteret, J.-P., Sticko, S., Turcksin, B. & Wells, D. The deal.II library, Version 9.4. *Journal of Numerical Mathematics* **30,** 231–246 (2022).

44. Bungartz, H.-J., Lindner, F., Gatzhammer, B., Mehl, M., Scheufele, K., Shukaev, A. & Uekermann, B. preCICE–a fully parallel library for multi-physics surface coupling. *Computers & Fluids* **141,** 250–258 (2016).

45. Kaneko, S., Nakamura, T., Inada, F., Kato, M., Ishihara, K., Nishihara, T., Mureithi, N. W. & Langthje, M. A. *Flow-Induced Vibrations: Classifications and Lessons from Practical Experiences: Second Edition* 28. ISBN: 9780080983479 (Elsevier Ltd, 2013).

46. Pettigrew, M. J., Taylor, C. E., Fisher, N. J., Yetisir, M. & Smith, B. A. W. Flow-induced vibration: recent findings and open questions. *Nuclear Engineering and Design* **185,** 249–276 (1998).

47. *Flow-induced structural vibrations : symposium, Karlsruhe (Germany) August 14-16, 1972* eng. in (Springer-Verlag, Berlin ; 1974). ISBN: 038706317X.

48. Paidoussis, M. Real-life experiences with flow-induced vibration. *Journal of Fluids and Structures* **22,** 741–755. ISSN: 0889-9746 (2006).

49. Naudascher, E. & Rockwell, D. *Flow-induced vibrations : an engineering guide* 414. ISBN: 0486136132 (Courier Corporation, 2012, Mar. 2012).

50. Kottapalli, S. *Numerical Prediction of Flow Induced Vibrations in Nuclear Reactors through Pressure Fluctuation Modeling (MSc Thesis)* Delft, 2016.

51. Paidoussis, M. P. Vibration of Cylindrical Structures Induced by Axial Flow. *Journal of Engineering for Industry* **96,** 547–552. ISSN: 0022-0817 (May 1974).

52. Paidoussis, M. P. The dynamical behaviour of cylindrical structures in axial flow. *Annals of Nuclear Science and Engineering* **1,** 83–106 (1974).

53. Lin, W. H., Wambsganss, M. W. & Jendrzejczyk, J. A. *Wall-pressure fluctuations within a seven-rod array* tech. rep. (General Electric Co., 1981).

54. Bertocchi, F., Rohde, M., De Santis, D., Shams, A., Dolfen, H., Degroote, J. & Vierendeels, J. Fluid-structure interaction of a 7-rods bundle: Benchmarking numerical simulations with experimental data. *Nuclear Engineering and Design* **356.** ISSN: 00295493 (Jan. 2020).

55. Paidoussis, M. P. Flow-induced vibrations in nuclear reactors and heat exchangers. *Practical Experiences and State of Knowledge in Practical Experiences with Flow-Induced Vibrations,* 1–81 (1980).

56. Païdoussis, M. P. Dynamics of cylindrical structures in axial flow: A review. *Journal of Fluids and Structures* **107,** 103374 (2021).

57. Han, S. Y., Seo, J. S., Park, M. S. & Choi, Y. D. Measurements of the flow characteristics of the lateral flow in the 6 × 6 rod bundles with Tandem Arrangement Vanes. *Nuclear Engineering and Design* **239,** 2728–2736. ISSN: 0029-5493 (Dec. 2009).

58. Mehl, M., Uekermann, B., Bijl, H., Blom, D., Gatzhammer, B. & Van Zuijlen, A. Parallel coupling numerics for partitioned fluid-structure interaction simulations. *Computers and Mathematics with Applications* **71.** ISSN: 08981221 (2016).

59. Lin, W. H. Hydrodynamic forces on multiple circular cylinders oscillating in a viscous incompressible fluid. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik* **67,** 487–501 (1987).

60. De Santis, D. & Shams, A. Numerical modeling of flow induced vibration of nuclear fuel rods. *Nuclear Engineering and Design* **320,** 44–56. ISSN: 0029-5493 (Aug. 2017).

61. Hadžić, H. *FIV in Reactor Fuel Assembly* Course material for lecture series "FSI for industrial applications" at von Karman Institute for Fluid Dynamics. Sept. 2020.

62. Papukchiev, A. Numerical analysis of reactor relevant vibrations using advanced multiphysics CFD-CSM methods. *Nuclear Engineering and Design* **350,** 21–32. ISSN: 00295493 (Aug. 2019).

63. Vivaldi, D. & Ricciardi, G. *Assessment of an Euler-Bernoulli Beam Model Coupled to CFD in Order to Perform Fluid-Structure Simulations* in *ASME 2022 Pressure Vessels & Piping Conference* (July 2022), V003T04A002.

64. Liu, Z. G., Liu, Y. & Lu, J. Fluid-structure interaction of single flexible cylinder in axial flow. *Computers and Fluids* **56.** ISSN: 00457930 (2012).

65. Liu, Z. G., Liu, Y. & Lu, J. Numerical simulation of the fluid–structure interaction for two simple fuel assemblies. *Nuclear Engineering and Design* **258,** 1–12 (2013).

66. Farhat, C. & Lesoinne, M. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering* **182,** 499–515. ISSN: 0045-7825 (Feb. 2000).

67. Gatzhammer, B. *Efficient and Flexible Partitioned Simulation of Fluid-Structure Interactions* in (2015).

68. Van Zuijlen, A. H. & Bijl, H. Implicit and explicit higher order time integration schemes for structural dynamics and fluid-structure interaction computations. *Computers & Structures* **83,** 93–105. ISSN: 0045-7949 (Jan. 2005).

69. De Boer, A., van Zuijlen, A. H. & Bijl, H. Comparison of conservative and consistent approaches for the coupling of non-matching meshes. *Computer Methods in Applied Mechanics and Engineering* **197,** 4284–4297. ISSN: 0045-7825 (Sept. 2008).

70. Irons, B. M. & Tuck, R. C. A version of the Aitken accelerator for computer iteration. *International Journal for Numerical Methods in Engineering* **1,** 275–277 (1969).

71. Haelterman, R., Bogaers, A. E., Scheufele, K., Uekermann, B. & Mehl, M. Improving the performance of the partitioned QN-ILS procedure for fluid–structure interaction problems: Filtering. *Computers & Structures* **171,** 9–17. ISSN: 0045-7949 (July 2016).

72. Bijl, H., van Zuijlen, A., de Boer, A. & Rixen, D. *An introduction to numerical coupled simulation - Fluid-structure interaction (wb1417) Lecture Notes* tech. rep. (TU Delft, Delft, 2008).

73. De Boer, A., van der Schoot, M. S. & Bijl, H. Mesh deformation based on radial basis function interpolation. *Computers and Structures* **85.** ISSN: 00457949 (2007).

74. Greenshields, C. J. & Weller, H. G. Notes on computational fluid dynamics: General principles. *(No Title)* (2022).

75. Menter, F. R. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal* **32.** ISSN: 00011452 (1994).

76. Wilcox, D. C. Reassessment of the scale-determining equation for advanced turbulence models. *AIAA Journal* **26,** 1299–1310. ISSN: 0001-1452 (Nov. 1988).

77. Jones, W. P. & Launder, B. E. The prediction of laminarization with a two-equation model of turbulence. *International Journal of Heat and Mass Transfer* **15,** 301–314. ISSN: 0017-9310 (1972).

78. Rubiolo, P. R. & Young, M. Y. On the factors affecting the fretting-wear risk of PWR fuel assemblies. *Nuclear Engineering and Design* **239,** 68–79 (2009).

79. Luk, K. H. *Pressurized-water reactor internals aging degradation study. Phase 1* in (1993).

80. Öchsner, A. & Merkel, M. *One-Dimensional Finite Elements* (Springer, 2013).

81. Bathe, K. J. *Finite element procedures* (Klaus Jurgen Bathe, 2006).

82. Kim, K. O. A review of mass matrices for eigenproblems. *Computers and Structures* **46.** ISSN: 00457949 (1993).

83. Païdoussis, M. P. & Gagnon, J. O. Experiments on vibration of clusters of cylinders in axial flow: Modal and spectral characteristics. *Journal of Sound and Vibration* **96,** 341–352. ISSN: 0022-460X (Oct. 1984).

84. Logan, D. L. *First Course in the Finite Element Method, Enhanced Edition, SI Version* (Cengage Learning, 2022).

85. Ko, P. L. Wear of power plant components due to impact and sliding (1997).

86. Yoo, Y., Kim, K., Eom, K. & Lee, S. Finite element analysis of the mechanical behavior of a nuclear fuel assembly spacer grid. *Nuclear Engineering and Design* **352,** 110179 (2019).

87. Khennane, A. *Introduction to finite element analysis using MATLAB® and Abaqus* (2013).

88. Megson, T. H. *Aircraft Structures for Engineering Students* (2012).

89. Hughes, T. *The Finite Element Method. Linear Static and Dynamic Finite Element Analysis* **2** (1987).

90. Papukchiev, A. GRS contributions to flow-induced vibrations related activities in Europe. *Kerntechnik* **88.** ISSN: 09323902 (2023).

91. Tengs, E., Charrassier, F., Holst, M. & Storli, P.-T. Model order reduction technique applied on harmonic analysis of a submerged vibrating blade. *International Journal of Applied Mechanics and Engineering* **24,** 131–142 (2019).

92. Antoulas, A. C. *Approximation of large-scale dynamical systems* (SIAM, 2005).

93. Besselink, B., Tabak, U., Lutowska, A., Van De Wouw, N., Nijmeijer, H., Rixen, D. J., Hochstenbach, M. E. & Schilders, W. H. *A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control* Sept. 2013.

94. Liu, Z. S., Chen, S. H., Liu, W. T. & Zhao, Y. Q. An accurate modal method for computing responses to harmonic excitation. *Modal Analysis: The International Journal of Analytical and Experimental Modal Analysis* **9,** 1–14 (1994).

95. Liu, Z. S., Wang, K., Huang, C. & Chen, S.-H. An extended hybrid method for contribution due to truncated lower-and higher-frequency modes in modal summation. *Engineering structures* **18,** 558–563 (1996).

96. Huang, C., Liu, Z. S. & Chen, S. H. An accurate modal method for computing response to periodic excitation. *Computers & structures* **63,** 625–631 (1997).

97. Zhao, Y. Q., Chen, S. H., Chai, S. & Qu, Q. W. An improved modal truncation method for responses to harmonic excitation. *Computers & structures* **80,** 99–103 (2002).

98. Tiso, P., Mahdiabadi, M. K. & Marconi, J. in *System- and Data-Driven Methods and Algorithms* 97–138 (De Gruyter, Nov. 2021). ISBN: 9783110498967.

99. Rixen, D. J. *Generalized mode acceleration methods and modal truncation augmentation* in *19th AIAA Applied Aerodynamics Conference* (American Institute of Aeronautics and Astronautics Inc., 2001).

100. Fehr, J. & Eberhard, P. Simulation process of flexible multibody systems with non-modal model order reduction techniques. *Multibody System Dynamics* **25,** 313–334 (2011).

101. Hansteen, O. E. & Bell, K. On the accuracy of mode superposition analysis in structural dynamics. *Earthquake Engineering & Structural Dynamics* **7,** 405–411 (1979).

102. Léger, P., Wilson, E. L. & Clough, R. W. *The use of load dependent vectors for dynamic and earthquake analyses* (Earthquake Engineering Research Center, College of Engineering, University of California, 1986).

103. Dickens, J. M., Nakagawa, J. M. & Wittbrodt, M. J. A critique of mode acceleration and modal truncation augmentation methods for modal response analysis. *Computers & structures* **62,** 985–998 (1997).

104. Yiu, Y. *Selective modal extraction for dynamic analysis of space structures* in *30th Structures, Structural Dynamics and Materials Conference* (1989), 1163.

105. Akgün, M. A. A New Family Of Mode-Superposition Methods For Response Calculations. *Journal of Sound and Vibration* **167,** 289–302. ISSN: 0022-460X (Oct. 1993).

106. Qu, Z. Hybrid expansion method for frequency responses and their sensitivities, Part I: undamped systems. *Journal of sound and vibration* **231,** 175–193 (2000).

107. Dickens, J. M. & Pool, K. V. Modal truncation vectors and periodic time domain analysis applied to a cyclic symmetry structure. *Computers & structures* **45,** 685–696 (1992).

108. Maddox, N. R. On the number of modes necessary for accurate response and resulting forces in dynamic analyses (1975).

109. Palmeri, A. & Lombardo, M. A new modal correction method for linear structures subjected to deterministic and random loadings. *Computers & structures* **89,** 844–854 (2011).

110. Rixen, D. J. *A Lanczos procedure for efficient mode superposition in dynamic analysis* in *Collection of Technical Papers - AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* **3** (American Inst. Aeronautics and Astronautics Inc., 2002), 1441–1450.

111. Bilello, C., Di Paola, M. & Salamone, S. A correction method for dynamic analysis of linear continuous systems. *Computers and Structures* **83,** 662–670. ISSN: 00457949 (Mar. 2005).

112. Wilson, E. L., Yuan, M.-W. & Dickens, J. M. Dynamic analysis by direct superposition of Ritz vectors. *Earthquake Engineering & Structural Dynamics* **10,** 813–821 (1982).

113. Nour-Omid, B. & Clough, R. W. Dynamic analysis of structures using Lanczos co-ordinates. *Earthquake engineering & structural dynamics* **12,** 565–577 (1984).

114. Léger, P. Load dependent subspace reduction methods for structural dynamic computations. *Computers & structures* **29,** 993–999 (1988).

115. Gu, J., Ma, Z. D. & Hulbert, G. M. A new load-dependent Ritz vector method for structural dynamics analyses: quasi-static Ritz vectors. *Finite Elements in Analysis and Design* **36,** 261–278 (2000).

116. Joo, K., Wilson, E. L. & Leger, P. Ritz vectors and generation criteria for mode superposition analysis. *Earthquake Engineering & Structural Dynamics* **18,** 149–167. ISSN: 10969845 (1989).

117. Arnold, R. R., Citerley, R. L., Chargin, M. & Galant, D. Application of Ritz vectors for dynamic analysis of large structures. *Computers & structures* **21,** 901–907 (1985).

118. Benner, P. & Saak, J. Efficient balancing-based MOR for large-scale second-order systems. *Mathematical and Computer Modelling of Dynamical Systems* **17,** 123–143 (2011).

119. Moore, B. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE transactions on automatic control* **26,** 17–32 (1981).

120. Gugercin, S. & Li, J.-R. *Smith-type methods for balanced truncation of large sparse systems* in *Dimension Reduction of Large-Scale Systems: Proceedings of a Workshop held in Oberwolfach, Germany, October 19–25, 2003* (2005), 49–82.

121. Lumley, J. L. The structure of inhomogeneous turbulent flows. *Atmospheric turbulence and radio wave propagation,* 166–178 (1967).

122. Kerschen, G. & Golinval, J.-C. Physical interpretation of the proper orthogonal modes using the singular value decomposition. *Journal of Sound and vibration* **249,** 849–865 (2002).

123. Han, S. & Feeny, B. Application of proper orthogonal decomposition to structural vibration analysis. *Mechanical Systems and Signal Processing* **17,** 989–1001 (2003).

124. Rafiq, D. & Bazaz, M. A. Model order reduction via moment-matching: a state of the art review. *Archives of Computational Methods in Engineering* **29,** 1463–1483 (2022).

125. Gallivan, K., Grimme, E. & Van Dooren, P. Asymptotic waveform evaluation via a Lanczos method. *Applied Mathematics Letters* **7,** 75–80 (1994).

126. Arnoldi, W. E. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quarterly of applied mathematics* **9,** 17–29 (1951).

127. Witteveen, W. On the modal and non-modal model reduction of metallic structures with variable boundary conditions. *World Journal of Mechanics* **2,** 311 (2012).

128. Witteveen, W. *Comparison of CMS, Krylov and balanced truncation based model reduction from a mechanical application engineer's perspective* in *Topics in Experimental Dynamics Substructuring and Wind Turbine Dynamics, Volume 2: Proceedings of the 30th IMAC, A Conference on Structural Dynamics, 2012* (2012), 319–331.

129. Liew, H. L. & Pinsky, P. M. Matrix-Padé via Lanczos solutions for vibrations of fluid-structure interaction. *International Journal for Numerical Methods in Engineering* **84,** 1183–1204. ISSN: 00295981 (Dec. 2010).

130. Uekermann, B., Chourdakis, G., Davis, K., Rodenberg, B., Schulte, M., Simonis, F., Abrams, G., Bungartz, H. J., Cheung Yau, L., Desai, I., Eder, K., Hertrich, R., Lindner, F., Rusch, A., Sashko, D., Schneider, D., Totounferoush, A., Volland, D., Vollmer, P. & Koseomur, O. Z. preCICE v2: A sustainable and user-friendly coupling library. *Open Research Europe* **2.** ISSN: 27325121 (2022).

131. Chourdakis, G., Schneider, D. & Uekermann, B. OpenFOAM-preCICE: Coupling OpenFOAM with External Solvers for Multi-Physics Simulations. *OpenFOAM® Journal* **3** (2023).

132. Jasak, H., Constantinescu, G. S., Krajewski, W. F., Ozdemir, C. E., Tokyay, T., Nešpor, V., Sevruk, B., Folland, C. K., Alletto, M., Breuer, M., Fitch, K. E., Hang, C., Talaei, A., Garrett, T. J., Thériault, J. M., Rasmussen, R. O., Petro, E., Trépanier, J. Y., Colli, M., Lanza, L. G., B. SEVRUK, J.-A. H., SPIESS, R., Cauteruccio, A., Brambilla, E., Stagnaro, M., Lanza, L. G., Rocchi, D., Hoover, J., Earle, M. E., Joe, P. I., Cauteruccio, A., Colli, M., Freda, A., Stagnaro, M., Lanza, L. G., Chinchella, E., Cauteruccio, A., Stagnaro, M., Lanza, L. G., Thériault, J. M., Rasmussen, R. O., Ikeda, K., Landolt, S., Davis, P. L., Rinehimer, A. T., Uddin, M., Cauteruccio, A., Colli, M., Lanza, L. G., La Barbera, P., Measurements, R. G., Sevruk, B., Hertig, J.-a. A., Tettamanti, R., Cauteruccio, A., Chinchella, E., Stagnaro, M., Lanza, L. G., Desalmand, M. D., Francoise, Baghapour, B., Sullivan, P. E., Nespor Sevruk R Spiess, V. B., Hertig, J.-a. A., Hall, D. J., Upton, S. L., Hoover, J., Earle, M. E., Joe, P. I., Sullivan, P. E., Colli, M., Lanza, L. G., Chan, P. W., Cauteruccio, A., Brambilla, E., Stagnaro, M., Lanza, L. G., Rocchi, D., Nespor, V., Baghapour, B., Wei, C., Sullivan, P. E., Colli, M., Pollock, M., Stagnaro, M., Lanza, L. G., Dutton, M., O'Connell, E., Leroux, N. R., Thériault, J. M., Rasmussen, R. O., Colli, M., Stagnaro, M., Lanza, L. G., Rasmussen, R. O., Thériault, J. M., Lanza, L. G., Bruce Baker, C. & Kochendorfer, J. Error Analysis and Estimation for the Finite Volume Method with Applications to Fluid Flows. *Journal of Hydrometeorology* **54.** ISSN: 01698095 (2021).

133. Guennebaud, G. & Jacob, B. *Eigen v3* 2010. `http://eigen.tuxfamily.org`.

134. Taylor, C. J. & Kriegman, D. J. Minimization on the Lie Group and Related Manifolds Minimization on the Lie Group SO (3) and Related Manifolds. *Technical Report No. 9405* (1994).

135. Chapra, S. C. & Canale, R. P. *Numerical Methods for Engineers* ISBN: 978-0-07-126759-5 (McGraw-Hill Higher Education, 2010).

136. Küttler, U. & Wall, W. A. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics* **43.** ISSN: 01787675 (2008).

137. Beckert, A. & Wendland, H. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology* **5.** ISSN: 12709638 (2001).

138. Helenbrook, B. T. Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering* **56.** ISSN: 00295981 (2003).

139. Lillberg, E., Angele, K. & Lundqvist, G. *Tailored experiments for validation of CFD with FSI for nuclear applications* in *NURETH-16* (2015).

140. De Ridder, J., Degroote, J., Van Tichelen, K., Schuurmans, P. & Vierendeels, J. Modal characteristics of a flexible cylinder in turbulent axial flow from numerical simulations. *Journal of Fluids and Structures* **43,** 110–123. ISSN: 0889-9746 (Nov. 2013).

141. Nazari, T., Rabiee, A. & Kazeminejad, H. Two-way fluid-structure interaction simulation for steady-state vibration of a slender rod using URANS and LES turbulence models. *Nuclear Engineering and Technology* **51,** 573–578. ISSN: 1738-5733 (Apr. 2019).

142. Smith, M. *ABAQUS/Standard User's Manual, Version 6.9* English (Dassault Systèmes Simulia Corp, United States, 2009).

143. Gere, J. & Timoshenko, S. *Mechanics of Materials* 4th, 549 (PWS, 1997).

144. Cioncolini, A., Silva-Leon, J., Cooper, D., Quinn, M. K. & Iacovides, H. Axial-flow-induced vibration experiments on cantilevered rods for nuclear reactor applications. *Nuclear Engineering and Design* **338,** 102–118. ISSN: 0029-5493 (Nov. 2018).

145. Arnold, D. N. & Logg, A. Periodic table of the finite elements. *Siam News* **47,** 212 (2014).

146. Arndt, A. *deal.II Doxygen Documentation* 2024.

Appendices

A

FEM supporting information

A.1. Combined element mass matrix for two-plane bending beam FE

$$\begin{array}{c} \\ u_{1y} \\ u_{1z} \\ \varphi_{1y} \\ \varphi_{1z} \\ u_{2y} \\ u_{2z} \\ \varphi_{2y} \\ \varphi_{2z} \end{array} \begin{array}{c} \begin{array}{cccccccc} u_{1y} & u_{1z} & \varphi_{1y} & \varphi_{1z} & u_{2y} & u_{2z} & \varphi_{2y} & \varphi_{2z} \end{array} \\ \begin{bmatrix} 156 & 0 & 0 & 22\,L_{\text{el}} & 54 & 0 & 0 & -13\,L_{\text{el}} \\ 0 & 156 & -22\,L_{\text{el}} & 0 & 0 & 54 & 13\,L_{\text{el}} & 0 \\ 0 & -22\,L_{\text{el}} & 4\,L_{\text{el}}^2 & 0 & 0 & -13\,L_{\text{el}} & -3\,L_{\text{el}}^2 & 0 \\ 22\,L_{\text{el}} & 0 & 0 & 4\,L_{\text{el}}^2 & 13\,L_{\text{el}} & 0 & 0 & -3\,L_{\text{el}}^2 \\ 54 & 0 & 0 & 13\,L_{\text{el}} & 156 & 0 & 0 & -22\,L_{\text{el}} \\ 0 & 54 & -13\,L_{\text{el}} & 0 & 0 & 156 & 22\,L_{\text{el}} & 0 \\ 0 & 13\,L_{\text{el}} & -3\,L_{\text{el}}^2 & 0 & 0 & 22\,L_{\text{el}} & 4\,L_{\text{el}}^2 & 0 \\ -13\,L_{\text{el}} & 0 & 0 & -3\,L_{\text{el}}^2 & -22\,L_{\text{el}} & 0 & 0 & 4\,L_{\text{el}}^2 \end{bmatrix} \end{array} \frac{\rho_s \frac{\pi D^2}{4} \cdot L_{\text{el}}}{420} = \boldsymbol{M_{el}} \tag{A.1}$$

A.2. The Mode Superposition technique

The Mode Superposition technique represents an alternative to the conventional numerical time integration techniques that are routinely used in structural analysis. As opposed to trying to satisfy the dynamic equilibrium at discrete time instances and modelling the variation of the structural displacements, velocities, and accelerations in between, the Mode Superposition technique is based on redefining the number of degrees of freedom of the problem through a projection of the original set of unknowns [81]. This is the same underlying concept that is also used for the ROMs presented in Section 2.4. In fact, it is most common that the reduction basis used to transform the initial FEM coordinates is the eigenbasis of the structure, the same approach as that used for the ROMs implemented in this thesis. Once the reduced system is obtained, an analytical solution can be used to determine the structural displacements.

Based on the observations made in the paragraph above, the Mode Superposition technique starts from the same projection as the modal methods presented in Subsections 2.4.3.2 to 2.4.3.4:

$$\boldsymbol{u}_{\mathbf{s,MS}}(t) = \sum_{\text{i}=1}^{N_{\text{FEM}}} a_{\text{s,i}}(t)\boldsymbol{\phi}_{\mathbf{s,i}} = \boldsymbol{\Phi}_{\mathbf{s}}\boldsymbol{a}_{\mathbf{s,MS}}(t), \tag{A.2}$$

with the important difference that here *all* of the N_{FEM} eigenmodes of the structure are used. In doing so, the solution $\boldsymbol{u}_{\mathbf{s,MS}}$ obtained by computing $\boldsymbol{a}_{\mathbf{s,MS}}$ and replacing it in (A.2) is *exact*, and not merely an approximation of the problem, as was the case in (2.100). Thus, a now-familiar reduced system is obtained:

$$\ddot{\boldsymbol{a}}_{\mathbf{s,MS}} + \boldsymbol{\Omega}_{\mathbf{s}}^2\boldsymbol{a}_{\mathbf{s,MS}} = \boldsymbol{\Phi}_{\mathbf{s}}^T\boldsymbol{f}_{\mathbf{s}}(t), \tag{A.3}$$

where the N_{FEM} equations are uncoupled from one another. Each of these can be individually written as:

$$\ddot{a}_{\text{s,i}}(t) + \omega_{\text{s,i}}^2 a_{\text{s,i}}(t) = \boldsymbol{\phi}_{\mathbf{s,i}}^T\boldsymbol{f}_{\mathbf{s}}(t),\ i = 1,2,\ldots N_{\text{FEM}}, \tag{A.4}$$

for which the projected initial conditions can be written as:

$$\begin{cases} a_{\mathrm{s,i}}\big|_{t=0} = \boldsymbol{\phi}_{\mathbf{s},\mathrm{i}}^T \boldsymbol{M}_{\boldsymbol{s}} \boldsymbol{u}_{\mathbf{s}}(0), \\ \dot{a}_{\mathrm{s,i}}\big|_{t=0} = \boldsymbol{\phi}_{\mathbf{s},\mathrm{i}}^T \boldsymbol{M}_{\boldsymbol{s}} \dot{\boldsymbol{u}}_{\mathbf{s}}(0). \end{cases} \tag{A.5}$$

Based on (A.4) and (A.5), each of the modal coordinates $a_{\mathrm{s,i}}$ can be solved for using a numerical integration technique, but one could also make use of the analytical solution of the Duhamel integral [81]:

$$a_{\mathrm{s,i}}(t) = \frac{1}{\omega_{\mathrm{s,i}}} \int_0^t \boldsymbol{\phi}_{\mathbf{s},\mathrm{i}}^T \boldsymbol{f}_{\mathbf{s}}(t') \sin\omega_{\mathrm{s,i}}(t-t')\,dt' + \zeta_{\mathrm{s,i}} \sin\omega_{\mathrm{s,i}} t + \eta_{\mathrm{s,i}} \cos\omega_{\mathrm{s,i}} t, \tag{A.6}$$

where $\zeta_{\mathrm{s,i}}$ and $\eta_{\mathrm{s,i}}$ are obtained based on the initial conditions (A.5). If one also considers structural damping, equation (A.6) becomes [81]:

$$a_{\mathrm{s,i}}(t) = \frac{1}{\omega_{\mathrm{s,d,i}}} \int_0^t \boldsymbol{\phi}_{\mathbf{s},\mathrm{i}}^T \boldsymbol{f}(t') e^{-\xi_{\mathrm{s,i}}\omega_{\mathrm{s,i}}(t-t')} \sin\omega_{\mathrm{s,d,i}}(t-t')\,dt' + e^{-\xi_{\mathrm{s,i}}\omega_{\mathrm{s,i}} t} \left(\zeta_{\mathrm{s,i}} \sin\omega_{\mathrm{s,d,i}} t + \eta_{\mathrm{s,i}} \cos\omega_{\mathrm{s,d,i}} t\right), \tag{A.7}$$

where $\xi_{\mathrm{s,i}}$ represents the damping of the i^{th} mode, and $\omega_{\mathrm{s,d,i}} = \omega_{\mathrm{s,i}}\sqrt{1-\xi_{\mathrm{s,i}}^2}$.

B

Supporting information, Chapter 3

B.1. Example of the `BeamFoamDisplacement` subdictionary

```
inner
{
    type    BeamFoamDisplacement;
    value   $internalField;

    // fluid model entries

        // density of the fluid (NEEDS TO BE SET INDEPENDENTLY FROM THE
        // transportProperties ENTRY, BUT THEY SHOULD BE EQUAL)
        rhoInf                                1000;

    // structural model entries

        // I. CHARACTERISTICS OF THE SPATIAL STRUCTURAL DISCRETIZATION

        // number of beam FEM elements used to model the structure
        NBeamElements                        20;

        // the number of segments into which each FEM element is subdivided
        NSegmentsPerElement               6;

        // II. CHARACTERISTICS OF THE TIME INTEGRATION SCHEME OF THE
        // STRUCTURAL PROBLEM

        // structural velocity integration parameter for the Newmark
        // method
        alphaNewmark           0.25;

        // structural displacement integration parameter for the Newmark
        // method
        deltaNewmark           0.5;

        // III. ROD GEOMETRICAL AND MECHANICAL PROPERTIES location of the
        // origin of the beam in the OpenFOAM frame of reference
        BeamOrigin                           (0 0 0);

        // length of the beam [m]
        BeamLength                           1.19;

        // radius of the beam [m]
        BeamRadius                           0.00635;

        // density of the beam [kg / m^3]
        rhoBeam                                 8400;

        // Young's Modulus of the beam [N/m^2]
        EBeam                                107000000000;

        // axial direction of the beam, as a normalized vector
        AxialDirection                       (0 0 1);
```

```
        // IV. SETTINGS FOR THE STRUCTURAL SOLVER

        // Should a non-fluid load be initially applied to the structure?
        // (relevant if dynamic response is of interest for a URANS
        // simulation); 1 for yes (URANS), 0 for no (PFM)
        ApplyInitialDistributedLoadBool 0;

        // note: the rest of the variables in IV are only relevant if
        // ApplyInitialDistributedLoadBool is set to 1

        // total duration for which the initial uniform load should be
        // applied [s]
        tApplicationInitialDistributedLoad 0;

        // amplitude of the initial load applied at each structural node [N]
        InitialLoadAmplitudeValue  0.654;

        // Eigenmode number that is desired to be imposed as forcing onto
        // structure (must be an integer greater or equal to 0; for 0, a
        // uniform load is applied in the axial direction of the rod)
        InitialLoadEigenmodeShapeNumber  0;

        // boolean which sets whether the initially applied load should be
        // static or dynamic
        DynamicInitialLoadBool 0;

        // relative amplitude of the dynamic component with respect to the
        // static one for the initial component
        DynamicInitialLoadRelAmplitude 0;

    // subiteration algorithm entries
        // the initial guess value of the Under Relaxation Factor (URF),
        // also called omega; this value is set at the first subiteration
        // of each timestep
        InitialUnderRelaxation 0.4;

    // ROM entries
        // should the structural solver be the ROM (either Mode
        // Displacement, MD, or Mode Acceleration, MA)? 1 for Yes, 0 for No
        UseROMbool 0;

        // note: the following 2 entries are only relevant if UseROMbool is
        // set to 1

        // should the ROM structural solver use the MA or the MD method? 1
        // for MA, 0 for MD
        UseModeAccelerationROMbool 0;

        // the number of modes to be used within the ROM (either for MA or
        // for MD); should be an integer with value greater than 0, and
        // smaller than the number of dof's within the FEM system
        NModesROM 10;

    // (Optional) Field names
        // those string inputs are used for the solver to be able to
        // identify what the name of the pressure, velocity, and density
        // fields is
        pName               p;
        rhoName             rhoInf;

//  Optional entries
    // field names
        // those string inputs are used for the solver to be able to
        // identify what the name of the pressure, and density
        // fields is
        pName               p;
        rhoName             rhoInf;

    // I. CHARACTERISTICS OF THE SPATIAL STRUCTURAL DISCRETIZATION
        // number of degrees of freedom allocated to each node (4 is
        // recommended; other valid option is 2, but it hasn't been fully
        // tested)
        nDofPerNode                         4;

        // the number of structural nodes for each element
        nNodesPerElement                2;

     // IV. SETTINGS FOR THE STRUCTURAL SOLVER
        // Are the structural elements uniform (not constant in length)?
        // Possible inputs are 1 for Yes (fully implemented), 0 for No
        // (partially implemented, not verified)
        UniformLengthElementsBool 1;
```

```
        // Are the structural elements homogeneous (same material and
        // geometrical properties for all axial positions)? Possible inputs
        // are 1 for Yes (fully implemented), 0 for No (partially
        // implemented)
        HomogeneousElementsBool 1;

}
```

C

Supporting information, Chapter 4

C.1. FEM solver

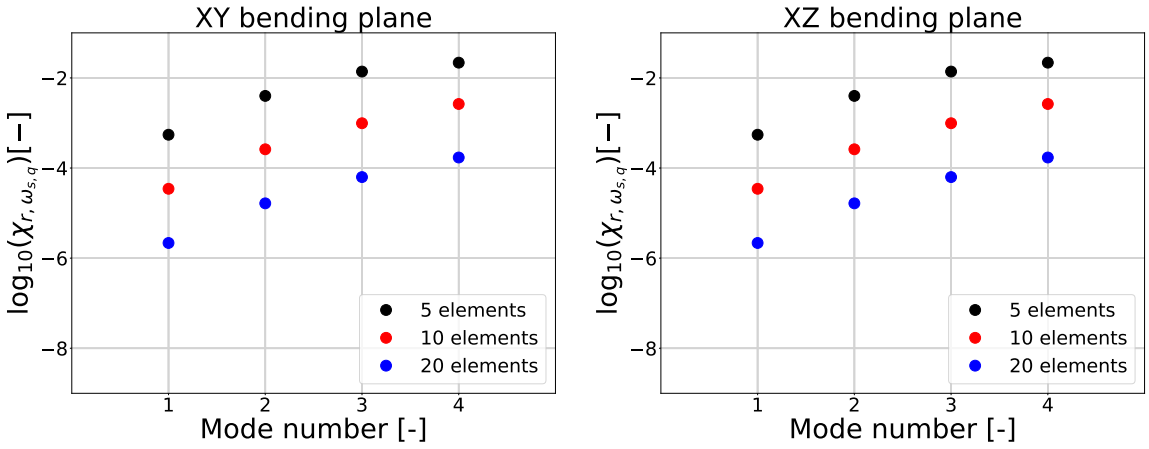


Figure C.1: The variation in relative error of the first four eigenfrequencies between the `beamFoam::FEMsolver` module and the analytical solution, for different numbers of elements used to discretize the fuel element geometry, when considering two-plane bending, for the XY (left) and the XZ (right) bending planes

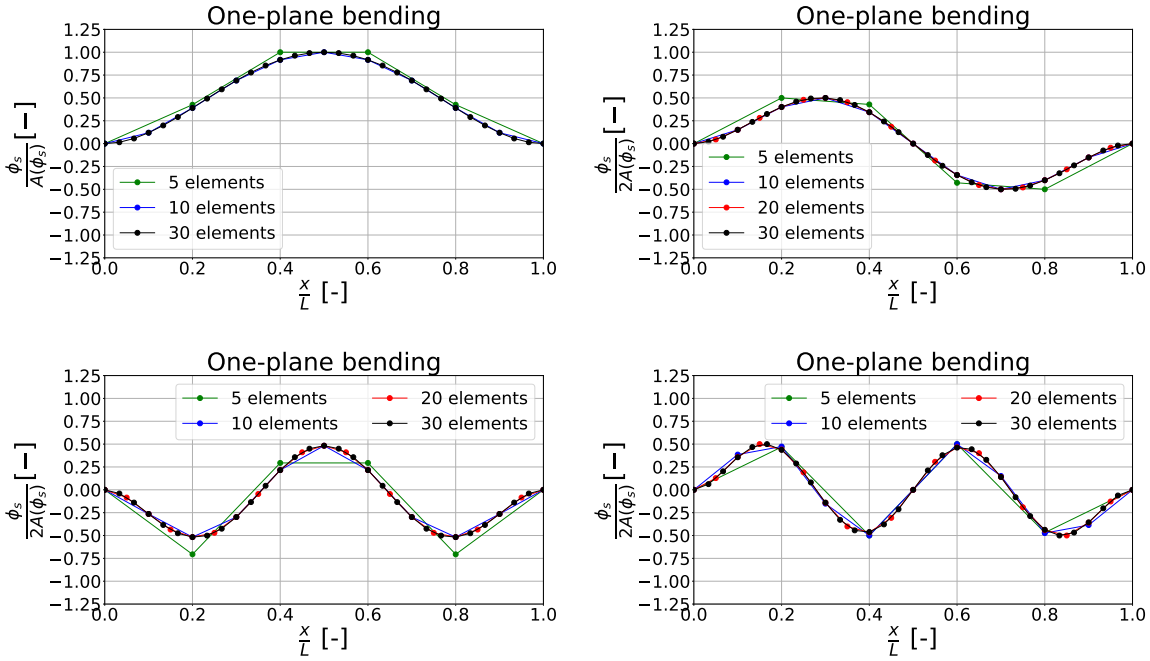


Figure C.2: The variation in the prediction of the first four eigenmodes for the `beamFoam::FEMsolver` module, when considering different numbers of beam elements to discretize the fuel element geometry, for one-plane bending FEM

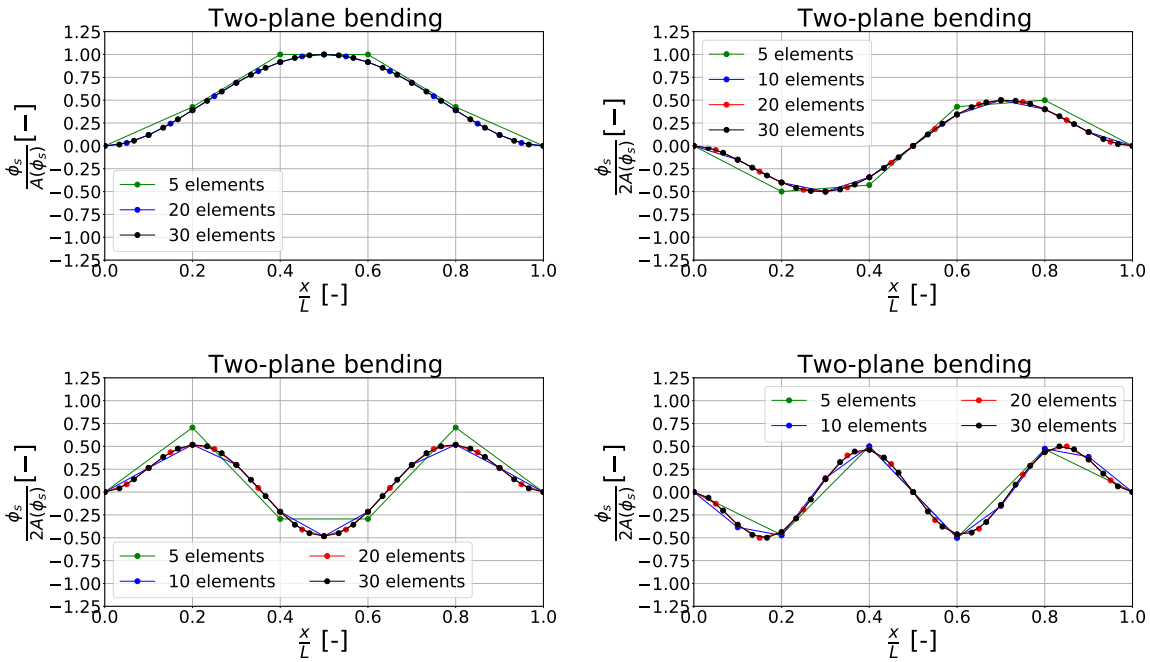
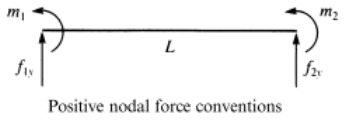


Figure C.3: The variation in the prediction of the first four eigenmodes for the `beamFoam::FEMsolver` module, when considering different numbers of beam elements to discretize the fuel element geometry, for two-plane bending FEM

C.2. Load Interpolator



Positive nodal force conventions

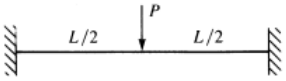
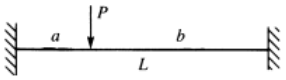
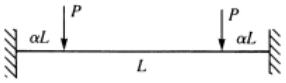
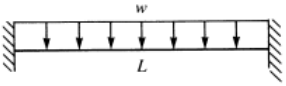
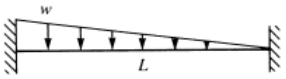
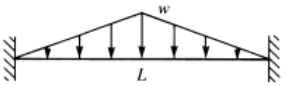
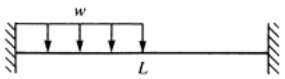
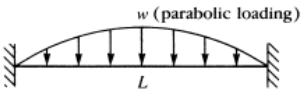
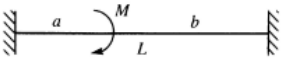
	f_{1y}	m_1	Loading case	f_{2y}	m_2
1.	$\frac{-P}{2}$	$\frac{-PL}{8}$		$\frac{-P}{2}$	$\frac{PL}{8}$
2.	$\frac{-Pb^2(L+2a)}{L^3}$	$\frac{-Pab^2}{L^2}$		$\frac{-Pa^2(L+2b)}{L^3}$	$\frac{Pa^2b}{L^2}$
3.	$-P$	$-\alpha(1-\alpha)PL$		$-P$	$\alpha(1-\alpha)PL$
4.	$\frac{-wL}{2}$	$\frac{-wL^2}{12}$		$\frac{-wL}{2}$	$\frac{wL^2}{12}$
5.	$\frac{-7wL}{20}$	$\frac{-wL^2}{20}$		$\frac{-3wL}{20}$	$\frac{wL^2}{30}$
6.	$\frac{-wL}{4}$	$\frac{-5wL^2}{96}$		$\frac{-wL}{4}$	$\frac{5wL^2}{96}$
7.	$\frac{-13wL}{32}$	$\frac{-11wL^2}{192}$		$\frac{-3wL}{32}$	$\frac{5wL^2}{192}$
8.	$\frac{-wL}{3}$	$\frac{-wL^2}{15}$		$\frac{-wL}{3}$	$\frac{wL^2}{15}$
9.	$\frac{-M(a^2+b^2-4ab-L^2)}{L^3}$	$\frac{Mb(2a-b)}{L^2}$		$\frac{M(a^2+b^2-4ab-L^2)}{L^3}$	$\frac{Ma(2b-a)}{L^2}$

Figure C.4: Static equivalent nodal loads for loads applied inside the structural domain; taken from [84]

C.3. Displacement Interpolator

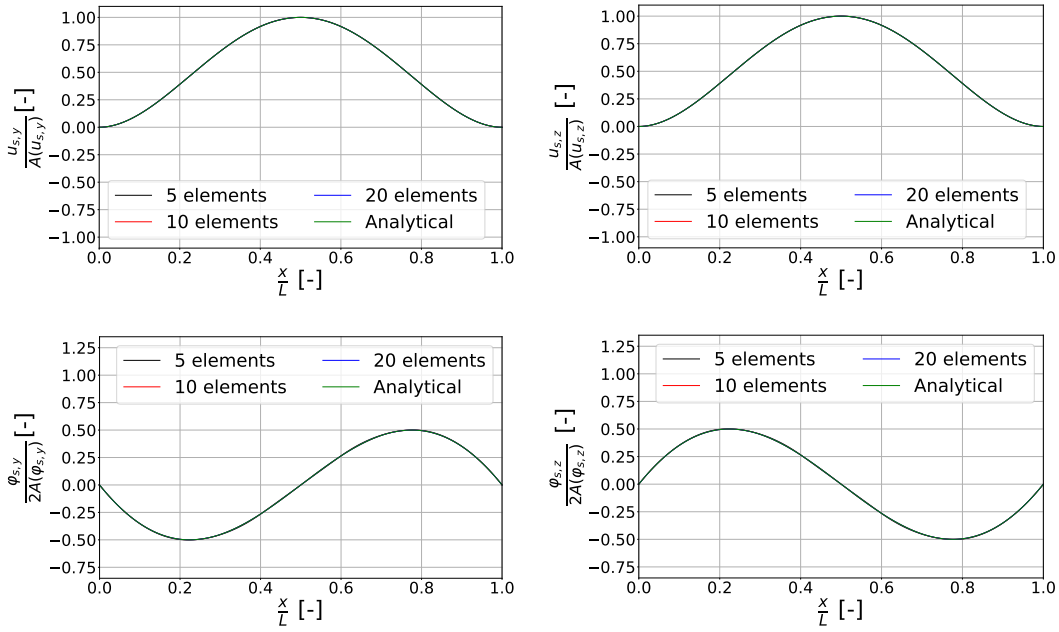


Figure C.5: The axial distributions of the unitary amplitude translational (top) and rotational (bottom) degrees of freedom obtained for the first eigenmode, at different levels of structural discretizations, compared with the analytical solution

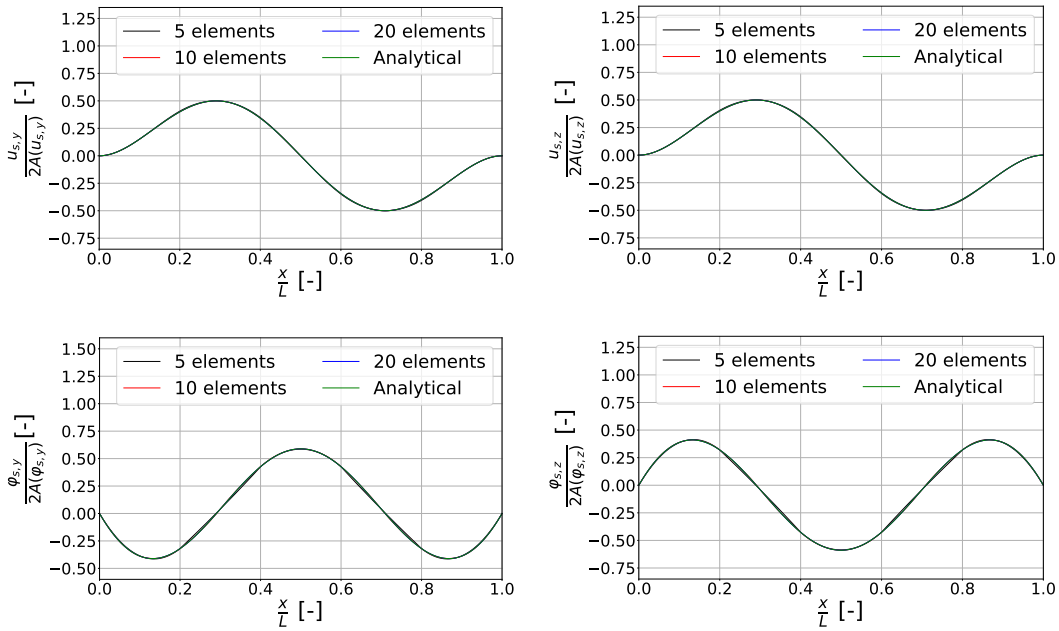


Figure C.6: The axial distributions of the unitary amplitude translational (top) and rotational (bottom) degrees of freedom obtained for the second eigenmode, at different levels of structural discretizations, compared with the analytical solution

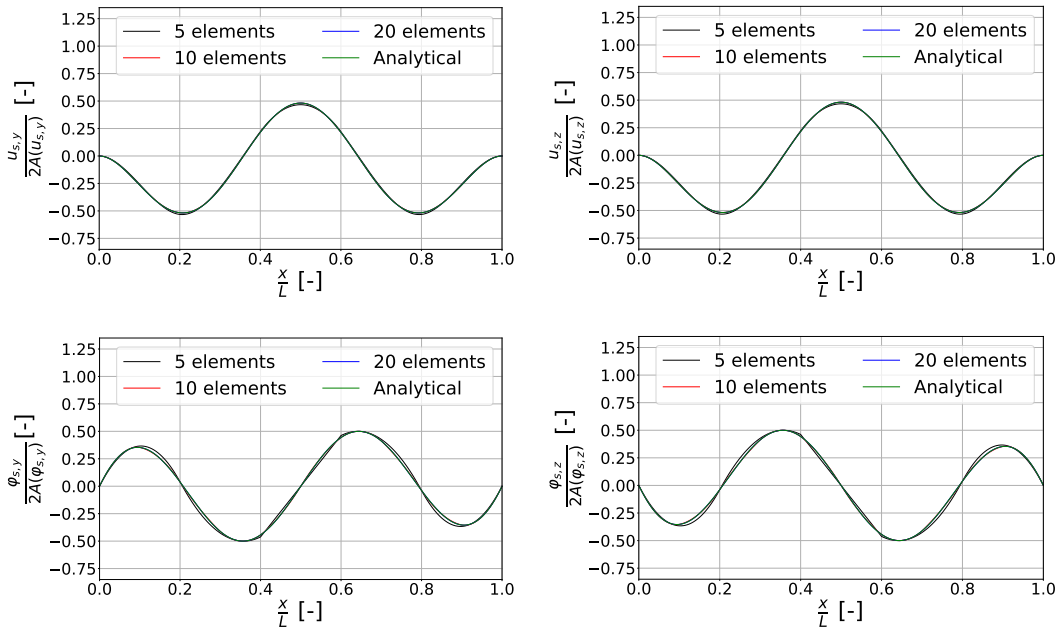


Figure C.7: The axial distributions of the unitary amplitude translational (top) and rotational (bottom) degrees of freedom obtained for the third eigenmode, at different levels of structural discretizations, compared with the analytical solution

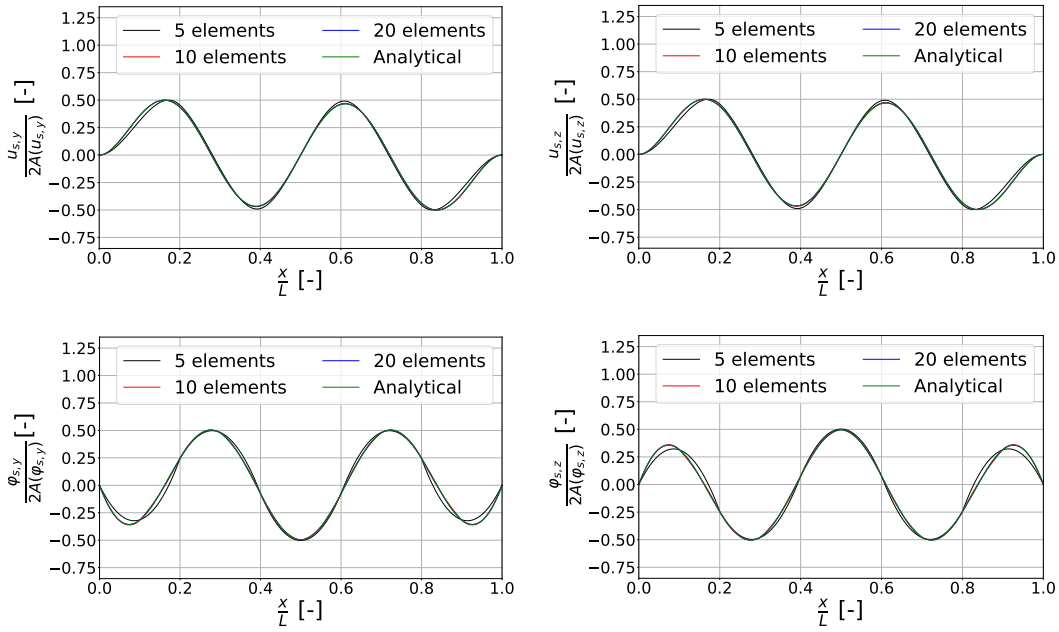


Figure C.8: The axial distributions of the unitary amplitude translational (top) and rotational (bottom) degrees of freedom obtained for the fourth eigenmode, at different levels of structural discretizations, compared with the analytical solution

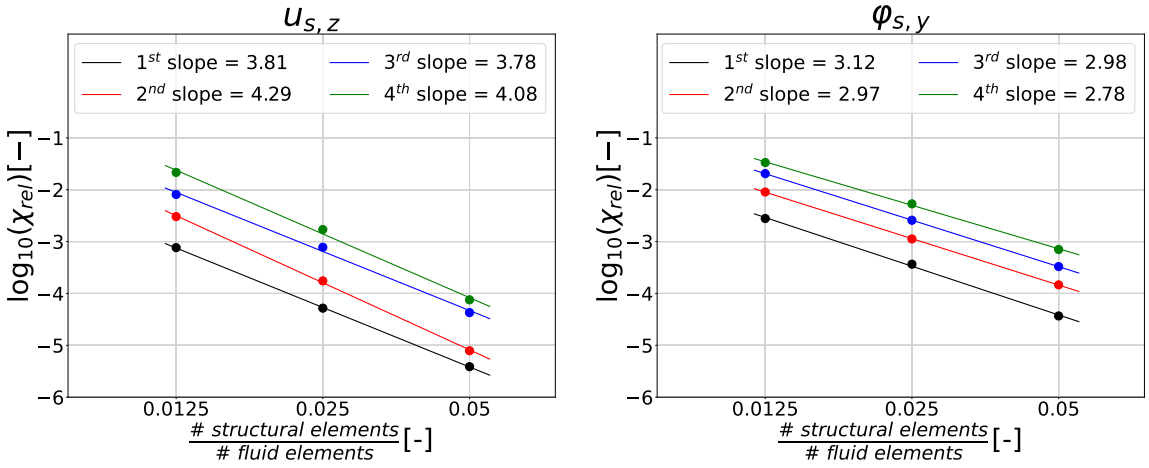


Figure C.9: The logarithmic convergence of the displacement interpolator relative error defined in (4.18) for the u_z (left) and the φ_y (right) degrees of freedom, at varying levels of structural discretization

D

Supporting information, Chapter 5

D.1. Laminar FSI simulations

D.1.1. Numerical schemes used (`fvSchemes` dictionary)

```
ddtSchemes
{
    default         backward;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default             none;
    div(phi,U)          Gauss linearUpwind linear;
    div((nuEff*dev2(T(grad(U))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear uncorrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         uncorrected;
}

wallDist
{
    method meshWave;
    nRequired true;
}
```

D.1.2. Solver settings used (`fvSolutions` dictionary)

```
solvers
{
    p
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-06;
        relTol          0;
        maxIter                 2000;
    }

    pcorr
    {
        $p;
    }

    pFinal
    {
        $p;
    }

    pcorrFinal
    {
        $pFinal;
    }

    "(U|cellDisplacement)"
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-07;
        relTol          0;
    }

    "(U|cellDisplacement)Final"
    {
        $U;
    }
}

PIMPLE
{
    // for NRG-beamFoam entry of nOuterCorrectors controls
    // the number of FSI subiterations for each timestep
    nOuterCorrectors 1;
    nCorrectors     2;
    nNonOrthogonalCorrectors 0;
    pRefCell        0;
    pRefValue       0;

    // SETTING ONLY USED WITH NRG-beamFoam
    moveMeshOuterCorrectors true;
}
```

D.1.3. Initial displacements, NRG-beamFoam vs NRG-FSIFOAM

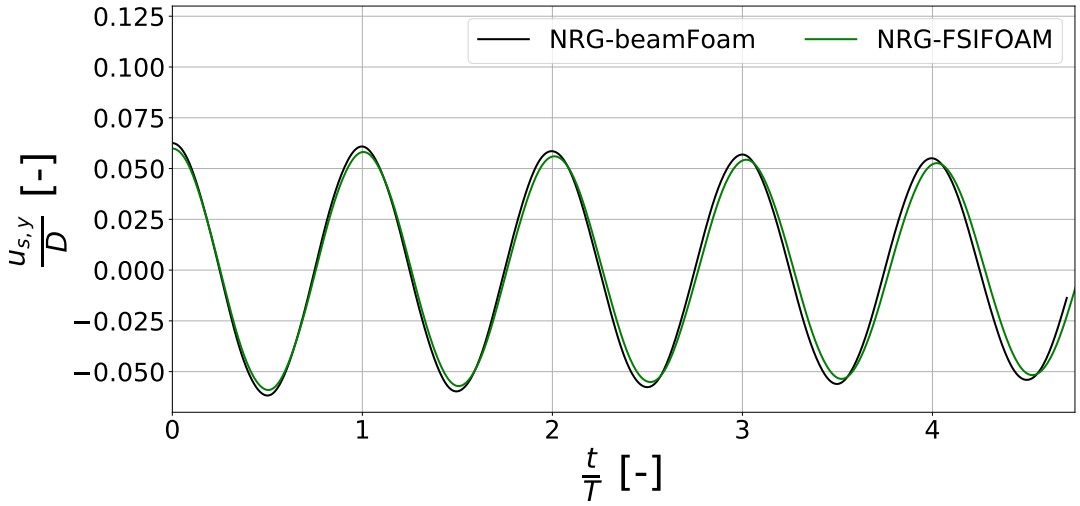


Figure D.1: The difference between the NRG-beamFoam and the NRG-FSIFOAM workflows for the transient lateral displacement obtained at the midpoint of the brass beam geometry; this difference in amplitude is caused by the the different methods of applying the initial forcing

D.1.4. Additional results, subiterator verification

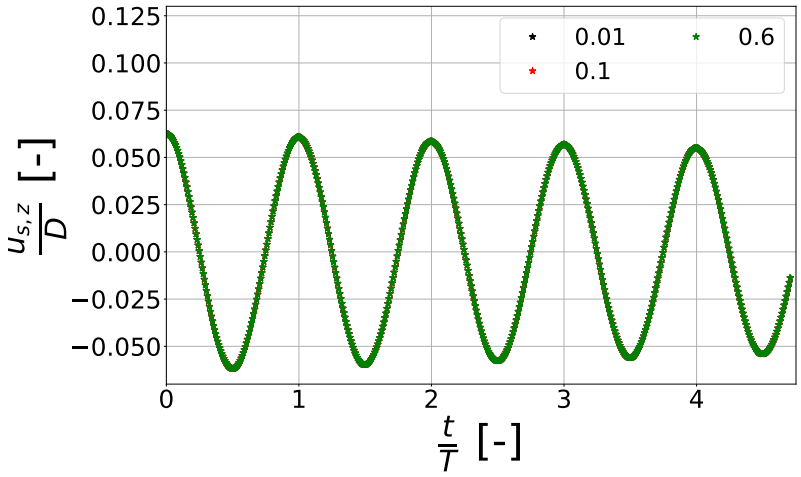


Figure D.2: The z component of the lateral midpoint displacement of the rod for the laminar FSI case for the fixed URF verification test, where different values of the URF is used for 2500 OC at each timestep

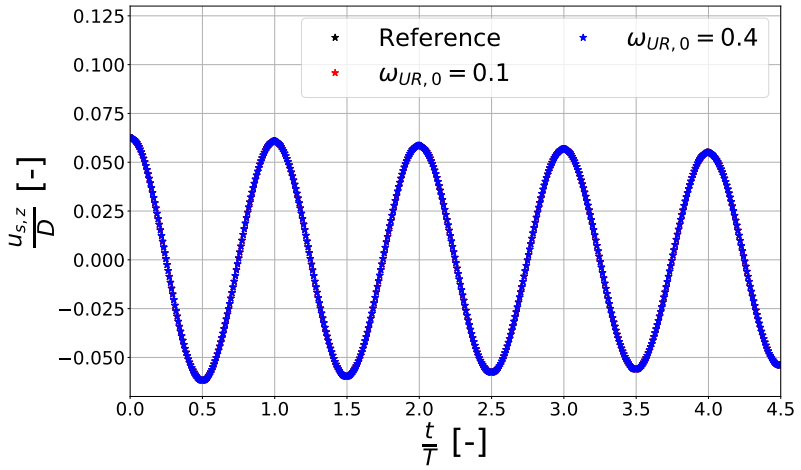


Figure D.3: The z component of the lateral midpoint displacement of the rod for the laminar FSI case for the Aitken URF verification test, where two different values of the intial URF is used for 10 OC at each timestep

D.2. URANS FSI simulations

D.2.1. Additional numerical schemes used in `fvSchemes` dictionary

```
divSchemes
{
    div(phi,omega)      Gauss linearUpwind linear;
    div(phi,k)          Gauss linearUpwind linear;
}
```

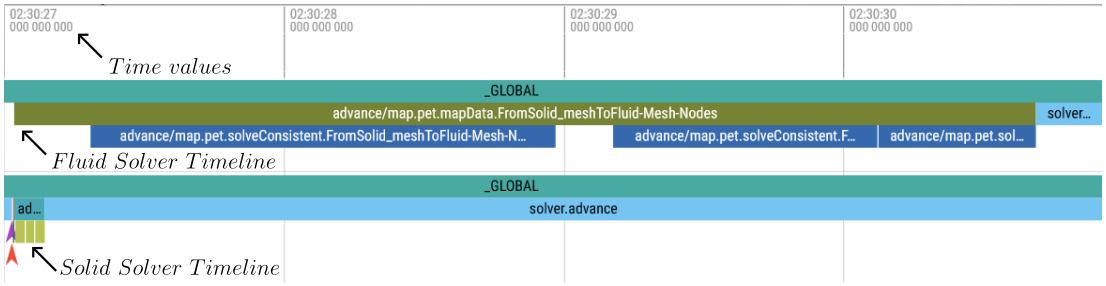
D.2.2. Additional solver settings used in `fvSolutions` dictionary

```
solvers
{
    "(omega|k)"
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-07;
        relTol          0;
    }

    "(omega|k)Final"
    {
        $omega;
    }
}

PIMPLE
{
    // SETTING ONLY USED WITH NRG-beamFoam
    turbOnFinalIterOnly false;
}
```

D.3. preCICE outputs example



Event	Count	Total[ms]	Max[ms]	Min[ms]	Avg[ms]	Time Ratio
_GLOBAL	1	59425017	59425017	59425017	59425017	1
advance	6427	29051442	18878	24	4520	0.489
advance/m2n.receiveData	6427	4228	23	0	0	7.11e-05
advance/m2n.sendData	6427	722	1	0	0	1.21e-05
advance/map.pet.AO.FromSolid_meshToFluid-Mesh-Nodes	1	0	0	0	0	0
advance/map.pet.computeMapping.FromSolid_meshToFluid-Mesh-Nodes	1	3579	3579	3579	3579	6.02e-05
advance/map.pet.computeRescaling.FromSolid_meshToFluid-Mesh-Nodes	1	2617	2617	2617	2617	4.4e-05
advance/map.pet.createMatrices.FromSolid_meshToFluid-Mesh-Nodes	1	33	33	33	33	5.55e-07
advance/map.pet.fillA.FromSolid_meshToFluid-Mesh-Nodes	1	321	321	321	321	5.4e-06
advance/map.pet.fillC.FromSolid_meshToFluid-Mesh-Nodes	1	136	136	136	136	2.29e-06
advance/map.pet.mapData.FromSolid_meshToFluid-Mesh-Nodes	6427	2175768	15614	10	338	0.0366
advance/map.pet.postFill.FromSolid_meshToFluid-Mesh-Nodes	1	168	168	168	168	2.83e-06
advance/map.pet.preComputeMapping.FromSolid_meshToFluid-Mesh-Nodes	1	0	0	0	0	0
advance/map.pet.preallocA.FromSolid_meshToFluid-Mesh-Nodes	1	127	127	127	127	2.14e-06
advance/map.pet.preallocC.FromSolid_meshToFluid-Mesh-Nodes	1	64	64	64	64	1.08e-06
advance/map.pet.solveConsistent.FromSolid_meshToFluid-Mesh-Nodes	19281	1433716	9532	0	74	0.0241
advance/map.pet.solverInit.FromSolid_meshToFluid-Mesh-Nodes	1	0	0	0	0	0
advance/query.index.getVertexIndexTree.Solid_mesh	1230	0	0	0	0	0
configure	1	0	0	0	0	0
configure/com.initializeMasterSlaveCom	1	0	0	0	0	0
finalize	1	110	110	110	110	1.85e-06
initialize	1	1310	1310	1310	1310	2.2e-05
initialize/m2n.broadcastVertexDistributions	3	5	2	1	1	8.41e-08
initialize/m2n.buildCommunicationMap	3	0	0	0	0	0
initialize/m2n.createCommunications	3	322	302	8	107	5.42e-06
initialize/m2n.exchangeVertexDistribution	3	115	71	14	38	1.94e-06
initialize/m2n.requestMasterConnection	1	846	846	846	846	1.42e-05
initialize/m2n.requestSlavesConnection	1	444	444	444	444	7.47e-06
initialize/partition.broadcastMesh.Solid_mesh	1	1	1	1	1	1.68e-08

Figure D.4: Examples of the outputs of preCICE for the performance analysis of the workflow; top: timeline of the different routines executed for the two solvers; bottom: summary of the different types of events in the timeline, along with their statistics for the entire simulation

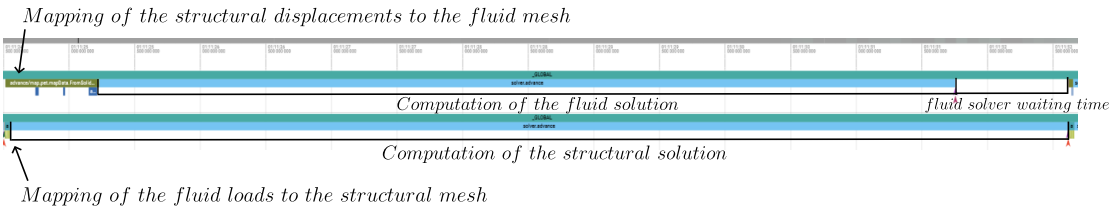


Figure D.5: Visualisation of the longer duration of computing the structural solution within the NRG-FSIFOAM workflow, for a random subiteration of the FSI transient

E

Supporting information, Chapter 6

E.1. Additional numerical schemes used in `fvSchemes` dictionary

```
divSchemes
{
    div(interpolate(Ufluct),Ufluct) Gauss linear;
    div(convection(interpolate(Ufluct),Ufluct)) Gauss linear;

    div(phi,Ufluct)         Gauss linear;
    div(phif,U)         Gauss linear;
    div(phif,Ufluct)        Gauss linear;
    div(phic,Vf)       Gauss linear;

    div(R)              Gauss linear;
    div(div(R))         Gauss linear;

    div(div(phi,Ufluct))    Gauss linear;
    div(div(phif,U))    Gauss linear;
    div(div(phif,Ufluct))   Gauss linear;
}
```

E.2. Additional solver settings used in fvSolutions dictionary

```
solvers
{
    pfluct
    {
        solver          PCG;
        preconditioner  DIC;
        tolerance       1e-09;
        relTol          0;
        maxIter         3000;
    }

    pfluctFinal
    {
        $pfluct;
    }

    Vf
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-09;
        relTol          0;
    }

    VfFinal
    {
        $Vf;
    }
}
```

E.3. The prescription of the Ufluct BC for the inlet and the outlet of the domain

In 0/Ufluct:

```
boundaryField
{
    inner
    {
        type            noSlip;
    }
    outer
    {
        type            noSlip;
    }
    outlet
    {
        type            mapped;
        field           Ufluct;
        setAverage      0;
        average         (0 0 0);
        interpolationScheme cellPointFace;
    }
    inlet
    {
        type            mapped;
        field           Ufluct;
        setAverage      0;
        average         (0 0 0);
        interpolationScheme cellPointFace;
    }
}
```

In constant/polyMesh/boundary:

```
outlet
{
    type            mappedPatch;
    inGroups        List<word> 1(mappedPatch);
    nFaces          2000;
    startFace       1810000;
    sampleMode      nearestCell;
    sampleRegion    region0;
    offsetMode      uniform;
    offset          (0 0 -0.1);
}
inlet
{
    type            mappedPatch;
    inGroups        List<word> 1(mappedPatch);
    nFaces          2000;
    startFace       1812000;
    sampleMode      nearestCell;
    sampleRegion    region0;
    offsetMode      uniform;
    offset          (0 0 0.1);
}
```

E.4. Distribution of fluid cell faces for matching and non-matching segments' nodes

$\kappa_{seg} = \frac{1}{5}$

Segment #	$\frac{N_{cf}}{N_{circ}}$	Segment #	$\frac{N_{cf}}{N_{circ}}$	Segment #	$\frac{N_{cf}}{N_{circ}}$
1	5	21	5	41	5
2	5	22	5	42	5
3	5	23	5	43	5
4	5	24	5	44	5
5	5	25	5	45	5
6	5	26	5	46	5
7	5	27	5	47	5
8	5	28	5	48	5
9	5	29	5	49	5
10	5	30	5	50	5
11	5	31	5	51	5
12	5	32	5	52	5
13	5	33	5	53	5
14	5	34	5	54	5
15	5	35	5	55	5
16	5	36	5	56	5
17	5	37	5	57	5
18	5	38	5	58	5
19	5	39	5	59	5
20	5	40	5	60	5

Table E.1: The number of cell faces assigned to each segment, N_{cf}, relative to the number of cell faces along the circumferential direction, N_{circ}, when segments' nodes are axially matching with the fluid ones on the FSI boundary, $\kappa_{seg} = \frac{1}{5}$

$\kappa_{seg} = \frac{2}{5}$

Segment #	$\frac{N_{cf}}{N_{circ}}$	Segment #	$\frac{N_{cf}}{N_{circ}}$	Segment #	$\frac{N_{cf}}{N_{circ}}$	Segment #	$\frac{N_{cf}}{N_{circ}}$	Segment #	$\frac{N_{cf}}{N_{circ}}$	Segment #	$\frac{N_{cf}}{N_{circ}}$
1	3	21	**2.28**	41	2	61	3	81	**2.08**	101	2
2	2	22	**2.72**	42	3	62	2	82	**2.92**	102	3
3	**2.15**	23	2	43	3	63	2	83	2	103	3
4	**2.85**	24	3	44	2	64	3	84	3	104	2
5	2	25	3	45	**2.02**	65	2	85	3	105	**2.3**
6	3	26	2	46	**2.98**	66	3	86	2	106	**2.7**
7	3	27	**2.12**	47	2	67	3	87	**2.12**	107	2
8	2	28	**2.88**	48	3	68	2	88	**2.88**	108	3
9	**2.05**	29	2	49	3	69	**2.08**	89	2	109	3
10	**2.95**	30	3	50	2	70	**2.92**	90	3	110	2
11	2	31	3	51	**2.05**	71	2	91	3	111	**2.12**
12	3	32	2	52	**2.95**	72	3	92	2	112	**2.88**
13	3	33	**2.48**	53	2	73	3	93	**2.15**	113	2
14	2	34	**2.52**	54	3	74	2	94	**2.85**	114	3
15	**2.65**	35	2	55	3	75	2	95	2	115	3
16	**2.35**	36	3	56	2	76	3	96	3	116	2
17	2	37	3	57	2	77	2	97	3	117	**2.08**
18	3	38	2	58	3	78	3	98	2	118	**2.92**
19	3	39	**2.02**	59	2	79	3	99	**2.33**	119	2
20	2	40	**2.98**	60	3	80	2	100	**2.67**	120	3

Table E.2: The number of cell faces assigned to each segment, N_{cf}, relative to the number of cell faces along the circumferential direction, N_{circ}, when segments' nodes are axially matching with the fluid ones on the FSI boundary, $\kappa_{seg} = \frac{2}{5}$; the bolded entries show that the number of fluid cell faces allocated to each segment is not necessarily a multiple of the number of fluid faces in the circumferential direction, confirming the hypothesis in Figure 6.6

E.5. Preliminary NRG-beamFoam study

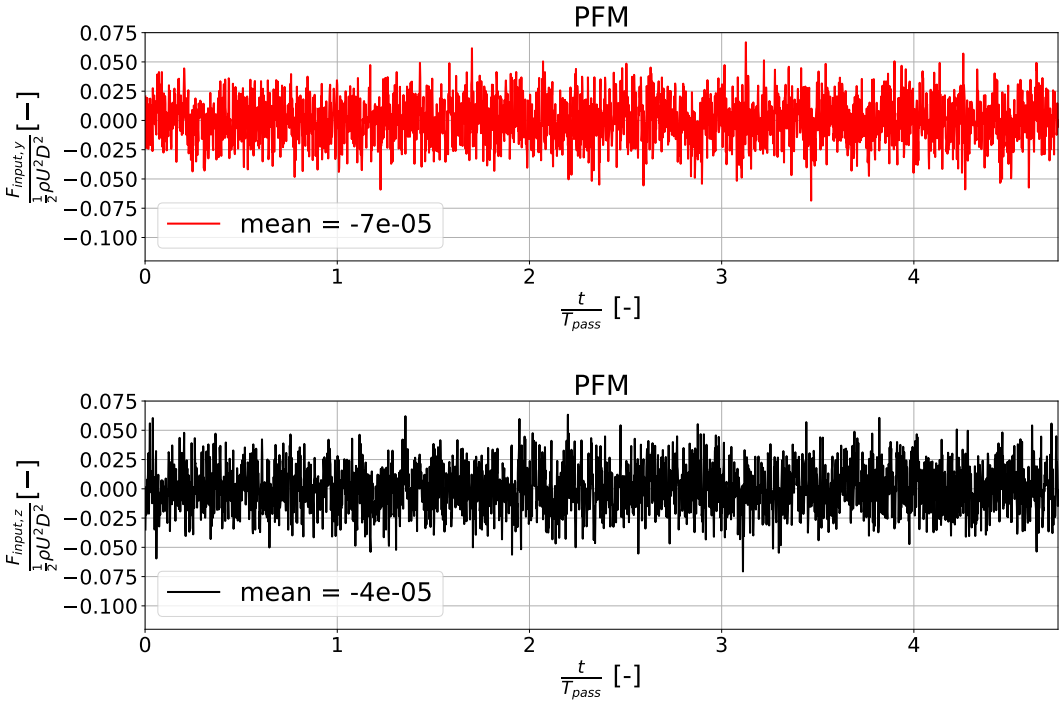


Figure E.1: The adimensionalized total forces applied on the FSI boundary by the fluctuating pressure components of the fluid domain in the $y-$ (top) and the $z-$ (bottom) directions of the beam reference system

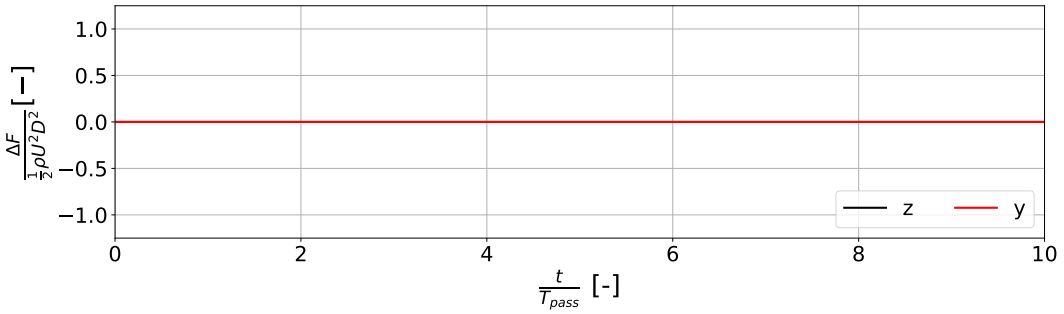


Figure E.2: The adimensionalized difference in total forcing received by the structural grid, and that imposed by the fluid grid, for the $y-$ and the $z-$ components of the beam reference system, when using a matching grid

E.6. Prediction of the vibrations RMS

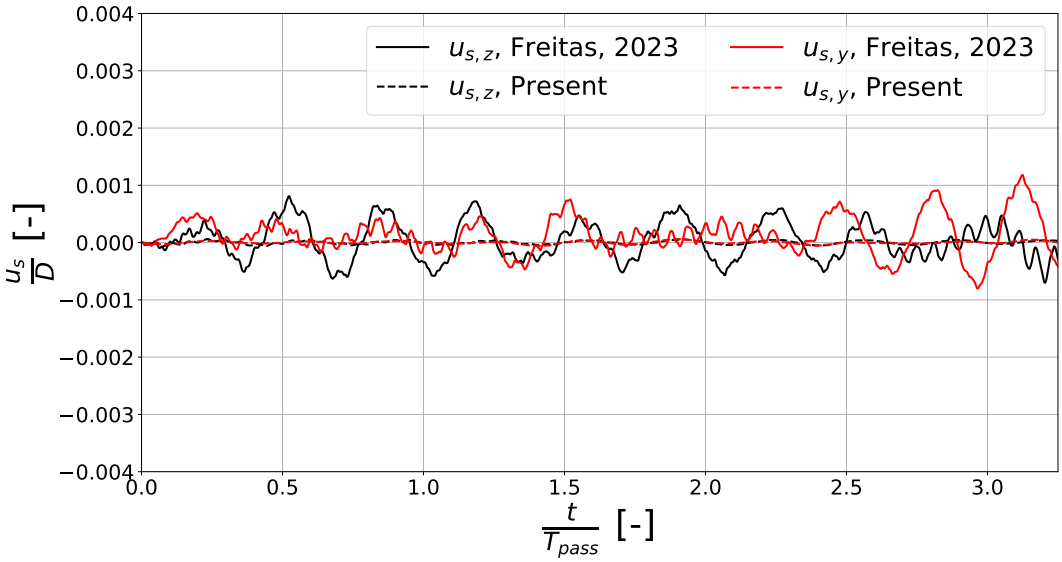


Figure E.3: The lateral midpoint displacements of the rod when using the NRG-FSIFOAM workflow with the settings presented in Section 6.1 obtained by Freitas [41], and by the current research

F

Supporting information, Chapter 7

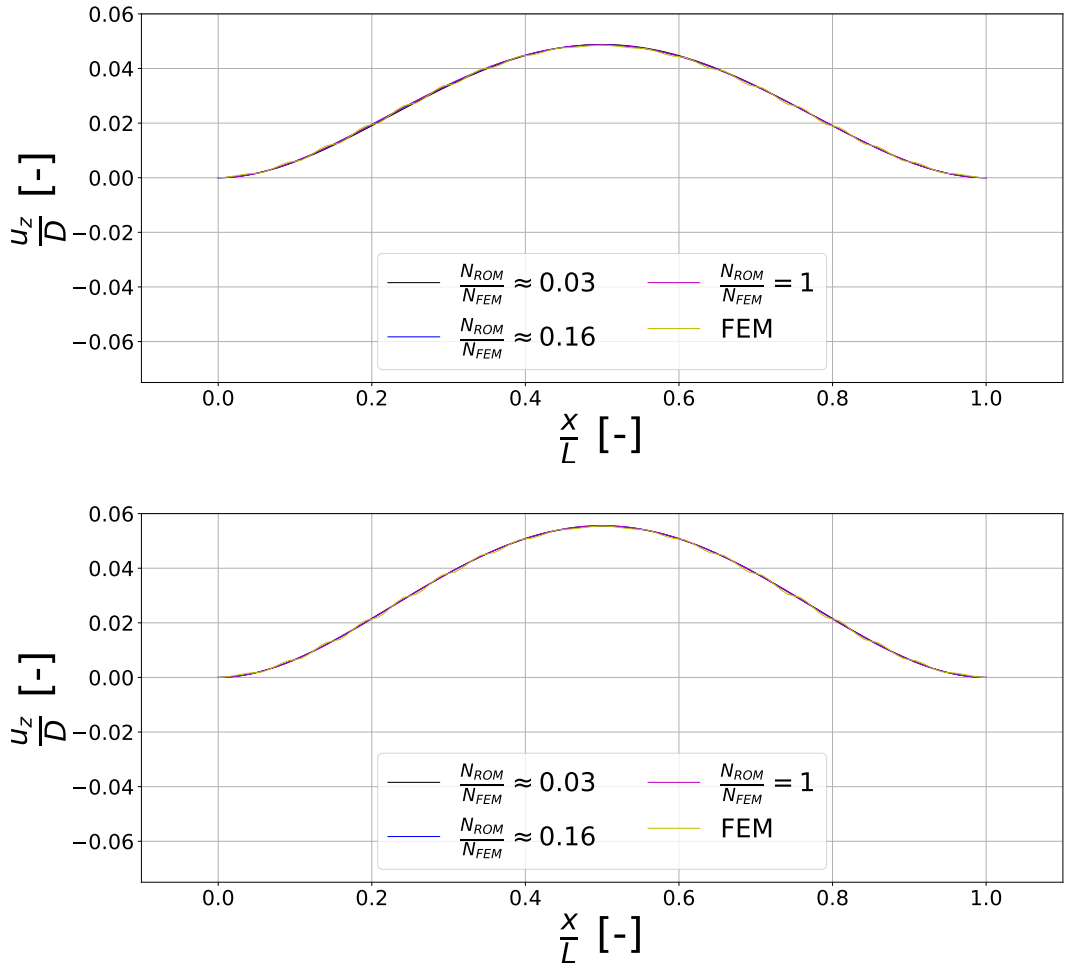


Figure F.1: Comparison of the axial distributions of lateral displacements in the $z-$ direction obtained within the ROM and the FEM structural solvers when a static uniform force is initially applied, and different numbers of modes are considered for each bending plane, after $\frac{t}{T_1} \approx 1.28$ (top) and $\frac{t}{T_1} \approx 3.42$ (bottom)

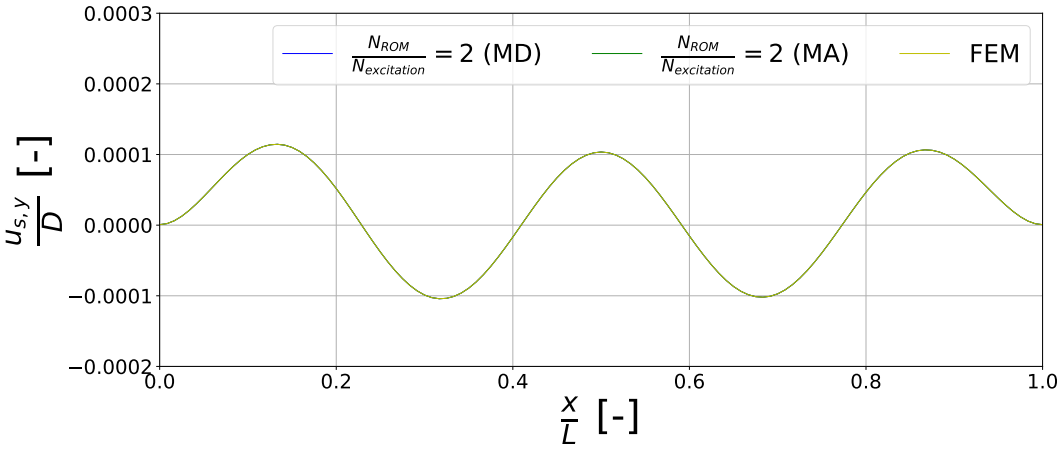


Figure F.2: Comparison of the axial distributions of lateral displacements in the $y-$ direction obtained within the ROM and the FEM structural solvers for the first ROM verification test, equation (7.1), at $\frac{t}{T_1} \approx 1.28$ (top) and $\frac{t}{T_1} \approx 3.42$ (bottom), when considering ten modes in each bending plane for the ROM, and the full FEM solution

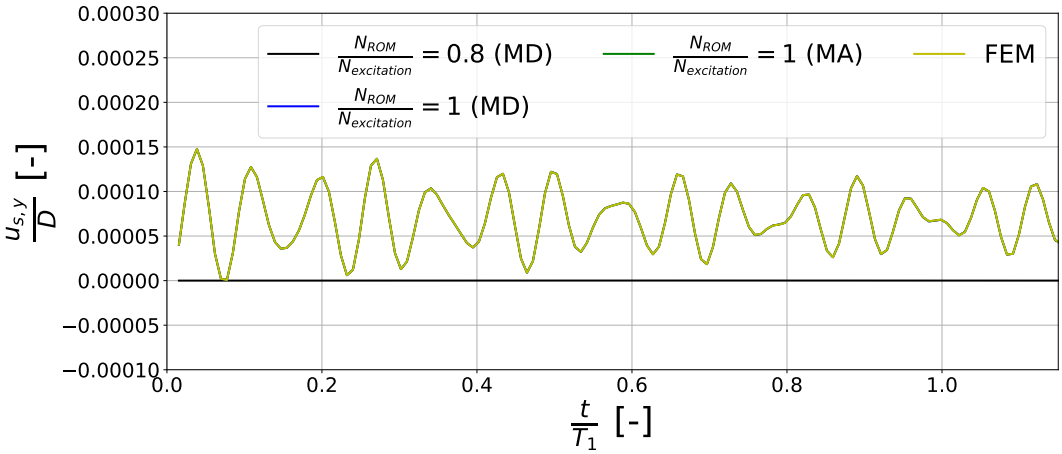


Figure F.3: Midpoint lateral displacements in the $y-$ direction for the third ROM verification test, equation (7.3), when considering five modes in each bending plane for the ROM method, and the full FEM solution

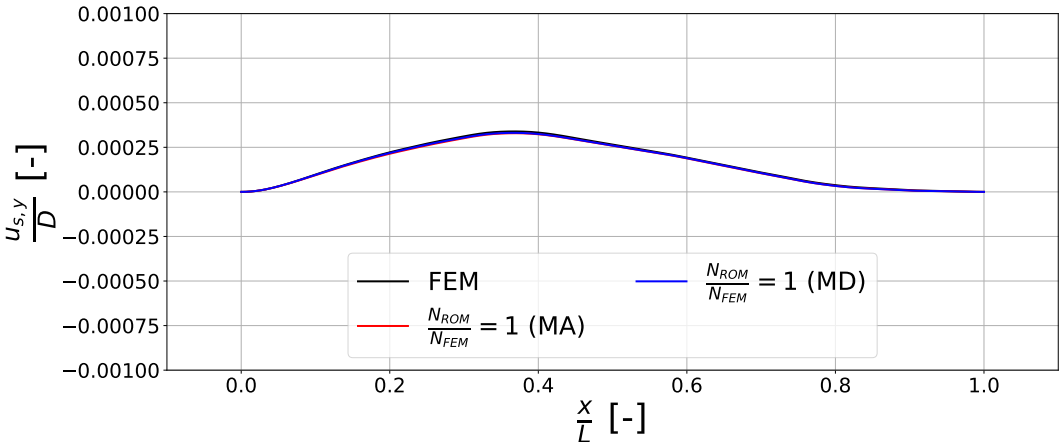


Figure F.4: Axial distribution of the lateral displacements imposed by the displacement interpolator onto the fluid grid's nodes on the FSI boundary when using the same number of modes as the total number of degrees of freedom used for the FEM model, $N_{\text{ROM}} = N_{\text{FEM}}$, for $\frac{t}{T_{pass}} \approx 3.36$, for both the MD and the MA ROM