SAPIENZA
UNIVERSITÀ DI ROMA

FACULTY OF CIVIL AND INDUSTRIAL ENGINEERING
DEPARTMENT OF ASTRONAUTICAL, ELECTRICAL AND ENERGY ENGINEERING

A Fluid Structure Interaction Framework for Digital Twins in Turbomachinery

Ph.D. Candidate:
Valerio Francesco Barnabei

Supervisor:
Prof. Alessandro Corsini, Ph.D.

Co-supervisors:
Alessio Castorrini, Ph.D.

2020

A DISSERTATION SUBMITTED IN PARTIAL SATISFACTION OF THE REQUIREMENTS FOR THE DOCTOR OF PHILOSOPHY DEGREE IN ENERGY AND ENVIRONMENT, XXXIII CYCLE.

To Cristina, Pasqualino, Lorenzo, and Claudia,
you are constantly shaping my happiness.

Contents

Acknowledgements

Throughout the entire period of my PhD I have received a great deal of support and assistance. I have also been constantly inspired by many professors, colleagues, and students that gave me the strength and motivation to overcome stagnation, doubts and fears.

I would like to thank my supervisor Prof. Alessandro Corsini, for the countless suggestions he gave me on every aspect of my work: your constant presence and your invaluable knowledge of numerical analysis, turbomachinery and artificial intelligence shaped my studies "on a multi-scale level", from understanding the single equation to the definition of the dissertation chapters layout.

I would also like to thank Dr. Alessio Castorrini, who guided me for all these years on a daily basis: you taught me so much more than programming, algorithms and fluid structure interaction. Your support truly made all of this possible, and you will always have my most sincere gratitude.

I would also like to thank Prof. Franco Rispoli, for all the questions he made me I was not able to answer: you constantly remembered me that the knowledge and the confidence in a subject are built on many bits of information, and that if something is missing, I have to search it from the fundamentals.

My special thanks are for Giovanni Delibra, Lorenzo Tieghi, Francesco Aldo Tucci: you were always there for me every time I needed you, no matter the day of the year, or the hour of the day. I hope you made a new friend in me as I did with all of you.

I would like to thank Prof. Johan van der Spuy, Prof. Domenico Borello, Prof. Paolo Venturini, for their help and guidance, whether it was in our department or at a conference abroad.

I would like to thank Dario Formisano for the extremely good work he did for his Master's thesis, since it made me easily drift above the vast literature about Digital Twins and Cyber Physical Systems.

At last, I would like to thank the ISMN group at CNR, for opening me the doors of their laboratory, and the eyes of their microscopes, to let me have a closer look on polymeric materials.

Abstract

Fluid Structure Interaction (FSI) is a class of multiphysics problems that couple the analysis of fluid dynamics of flows around solid objects and the structural dynamics of the same solid that interacts with the flow around it. By solving an FSI problem, is possible to obtain information about both the fluid and solid phases, as a function of their complex interactions, exchanges of forces, changes in shape, induced dynamics. This class of problem can be recognized in a large number of phenomena in nature, and also in the human technologies. Recently, following the constant effort to push forward the performances of the existing energy conversion technologies, FSI has been applied to several classes of rotating fluid machines: wind turbines, tidal turbines, air turbines, fans can all be observed from the FSI perspective when the designer is interested into the optimization and improvement of a device that, for different reasons (extremely large rotor radius, flexible blades, passive adaptive appendices or blades, thin structures) is expressing a non negligible interaction between the structure and the fluid flow.

So far, the FSI problem for complex geometries and flows such the ones characterizing internal and external flows in turbomachinery, was solved numerically using computational FSI models and algorithms, that are designed to couple the approximate solution of the Navier-Stokes (NS) equations with the elastodynamics equations.

At the same time, the growing interest in Cyber-Physical Systems (CPS) and Digital Twin (DT) technologies from the turbomachinery producers is explained by the numerous benefits that would occur adopting those technologies: prediction capabilities, design, testing and monitoring, in a real time digital environment would cut by a significative amount costs and time consumption at multiple stages of the products lifetime.

The attempt to build a DT (or in general, a CPS) able to describe some of the FSI phenomena in turbomachinery represents a challenging task, yet extremely interesting and promising. Among the several technical obstacles that a DT (or a CPS in general) would face when demanded to describe the FSI of a rotating fluid machinery there's the evident mismatch of the time scales of the time consuming FSI numerical simulations, and the real time functioning required by the DT to process the input signals and to give an adequate output with the least possible delay. This obstacle can be possibly overcome by decoupling the slow but high fidelity simulations and the data usage

from the DT: to achieve this, a database that collects FSI data can be built for a specific set of problems, and a Machine Learning (ML) algorithm might be used to build a Reduced Order Model (ROM) that would constitute the virtual core of the DT. A DT built on top of a ROM should be able to obtain results comparable to the ones from a canonical numerical simulation in a fraction of the time required by the simulation (orders of magnitudes). The shortcoming of this approach, aside of the complexity of the entire system, and the technical difficulties to build a suitable database and a reliable ROM, would obviously be the lack of generalization: such ROM (and DT) would be appplicable only to the specific subset of conditions that built the original database, possibly accepting only minor deviations. In spite of this shortcoming, from a manifacturer perspective this issue would be minor: the lack of generalization can still be compensated for by the production in series of the device, and the technology could still fit the scale economy.

In an attempt to follow the research path that would eventually lead to a DT with FSI capabilities in turbomachinery, in this dissertation is introduced the first fundamental block of this roadmap, on top of which the subsequent components will be built: FEMpar, a in-house developed software for FSI analysis using Finite Elements Method (FEM), is presented together with a description of the numerical models implemented to solve the NS equations, the nonlinear structural dynamics equations and the moving mesh problem.

Along with the presentation of the software tool and its main components, several applications of FEMpar to CFD-FSI analysis in turbomachinery are showcased as well, to highlight the potential and relevance of FSI analysis in those devices. The proposed cases were selected to highlight different features and benefits to adopt FSI at the design and testing stage of a rotating fluid machinery: the flow around fans with large diameters, extreme aspect ratio, or made with flexible materials is simulated to observe the interaction of thin structures immersed in an unsteady flow; the possible adoption of low stiffness materials to design passive adaptive blade, and the design of specific constraints that would allow the passive morphing of the blades up to the desired configurations, are explored for a reversible axial fan and a Wells turbine (both devices characterized by quick and impulsive changes in the flow direction).

1. Introduction

1.1 Background and motivations

If it would be possible to reduce all the research effort in the energy production to a single sentence, that would obviously be "increase efficiency and reduce costs". Regardless of how superficial and extreme this reduction may appear, it's the *leitmotiv* of the entire energy industry, and it's by far the strongest pulling factor for the research in every field strictly correlated to the energy conversion systems, particularly in engineering.

The increase in efficiency is commonly obtained via technological innovation (directly upgrading the available technology with a new, more efficient design), fine tuning of the design, or optimization processes specific for each technology. The reduction in costs can be achieved, in most cases, without compromising the quality of the final product, by drastically reducing the production time, reducing the possibility of failures, adopting cheaper materials (it is important to note that a "cheaper" material does not imply a "low quality" material: in this context "cheaper" stands for "less expensive", while not compromising any core functionality of the product, its quality, or the safety for those around it).

It is no secret that, at least so far in the history of technology, the most efficient, scalable, and most adaptive way to convert large amount of energy into electrical energy (or viceversa, to convert electrical energy into useful mechanical energy) is using a fluid machinery, where an energetic vector (i.e. the fluid) is used to convert mechanical or thermal energy from a wide range of different sources into electrical energy to directly supply the grid.

An interesting branch of the research in turbomachinery that is taking place is the application of the Fluid Structure Interaction (FSI) analysis to some specific devices, or some specific configurations of a device. An exhaustive definition of FSI will be given in later sections, however it can be briefly introduced as the attempt to solve the complex multi-physics problem regarding the interaction between a fluid in motion and the solid structure around which the fluid is flowing: in such interaction, the fluid will apply a force on the solid at the fluid structure interface, and consequently the solid will respond, eventually, with its motion, or by changing its shape; the motion

of the solid, consequently, will modify the flow around it, directly interacting with the fluid dynamics.

One of the most immediate, and relevant, example of FSI problems in turbomachinery is represented by the coupled dynamics that involves devices characterized by extremely thin blades, or by large diameter rotors, such as for example wind turbine blades. In fact, in the last 30 years has been observed a significative increment of the converted power in wind turbines, and consequently in rotor size. The adoption of large diameters, justified by a reduction of costs per KWh, directly leads to the insurgence of a series of undesired aeroelastic effects, caused by the interaction of aerodynamic load, elastic bending and inertia force. Those could easily lead to aeroelastic instabilities, undesired vibrations, up to performance losses and disruptive failures. The importance of FSI in the context of fluid machinery is not only justified by a need to control destructive interactions and to observe changes in performance due the growing size of wind turbines. Adopting a more general and wide perspective it is possible to recognize numerous FSI problems in turbomachinery: in general, if the structure of a rotating fluid machine is sufficiently thin or flexible to be positively affected by the aerodynamic load, it could experience vortex induced vibrations: the mutual coupling between body motion and vortex shedding produces a stable or unstable periodical state of the entire system with forced oscillation of the body coupled with a vortex shedding downstream. Hence, a multi-physics fully coupled analysis can provide higher fidelity information with respect to different analyses conducted on the separate domains: with respect to turbomachinery, it makes possible to have at the same time, better predictions of fatigue cycles and stress state and better prediction of performance.

The same tools and considerations can be applied at the design stage, when designing devices and components that, in general, interact with fluids. This is true, of course, for turbomachinery, not only regarding the design of specific turbines that by their core nature manifest a strong interaction between fluid and structure (i.e. the aforementioned large diameter wind turbines, tidal turbines submerged in a high density fluid such as water), but also the design of specific devices that use FSI to increase the base performance: an example of this kind of application could be represented by smart blades (for example in wind turbines, fans, Wells turbines), smart wings and ailerons (for example in aircrafts, automotive vehicles, ships), and morphing control surfaces (such as flexible appendices to passively control the flow). When moving the attention away from turbomachinery, the applicability of FSI analysis and methods grows exponentially: the design of valves, parachutes, airbags, the study of blood vessels, and the analysis of flapping motions of flying animals are only a few of the countless examples where a solid FSI analysis can provide useful insights.

Regardless of the engineering or research domain of interest, it appears crucial

to investigate the FSI problem in its integrity, by coupling as much as possible the structural dynamics and the fluid dynamics. Commonly, due to the complexity of the FSI problem (complex geometries, complex fluid dynamics) this is often done via numerical simulations. Unfortunately complex numerical simulations take a large amount of time, often more than what is affordable for a manufacturer: the required time and computational resource increases with the refinement, the reliability and the resolution of the simulation.

According to the Moore's law (which is an observation, more than a law), in the last 40 years the computational power of computers have grown linearly, almost doubling every two years, giving to engineers, physicists, programmers and numerical analysts free space to innovate their software, improve their models, and obtain astounding results every year. Unfortunately, even though the computational power limits are pushing forward their limits continuously, this is not always of good use for a manufacturer. In spite of the fact that the digital prototyping and testing of any new device is the standard procedure to even start considering a new product, the high level of details that could be obtained with extremely detailed and high fidelity simulation is still locked behind the gates of time, and probably will be for many years to come. In general, a production schedule is so tight in order to be economically viable, that no manufacturer would be remotely interested in such rich information, as it would rather prefer a quicker, less refined analysis, than to present late a product in the market, or to miss a deadline (with huge costs).

A possible solution to indirectly include the richness of high definition simulations is given by the growing large amount of data that is collected constantly, as a direct consequence of the increasing capabilities of computers, and data storage. The provided high quality data could be used to train and test Artificial Intelligence (AI) solutions that would constitute a surrogate alternative to the complex physics simulated by the computational fluid dynamics or the FSI tools. Such alternative is proving itself as a technology that, if properly tuned, can give comparable results to the canonical CFD or structural numerical simulations in a substantially lesser time (the reduction of the required time to have a comparable result is usually quantified in order of magnitudes).

In this context, the innovation brought by the paradigm of the Digital Twin, and by Cyber-Physical Systems in general, is remarkable: a Digital Twin, as will be described, is able to help to fulfill the "golden needs" of the energy industry, the increase in performance via constant optimization, and the reduction in costs. The first outcome, once a definition of Digital Twin is given, will be easy to recognize; the latter is not-so-well hidden in the Digital Twin name: imagine how less expensive would it be to insert in a design pipeline, or a testing workflow, a digital counterpart of a large, complex, and expensive machine, while still obtaining the same outputs given the same inputs

for a specific task or analysis! The constitution of this paradigm, from a technical perspective, is far from easy and several critical issues has to be considered, especially with fast machines involving complex physics like fluid machinery. Nonetheless, this paradigm shift is feasible and the research is thriving. The attempt to match even more complex physics like fluid structure interaction, is even harder. However, the possibility to compartmentalize a multiphysics problem into sub-problems can be of great help, and there's possibly space for FSI to find its role in Digital Twins too.

1.2 Thesis outline

The following dissertation will be structured in chapters, as follows:

- In Chap. *Cyber Physical Systems and Digital Twins in turbomachinery and CFD*, a brief introduction about Cyber Physical Systems and Digital twins is given. In particular, from the definitions the reader is guided towards the possible applications of the technologies in the field of turbomachinery, their current limitations and the possible ways to overcome such boundaries;

- In Chap. *The role of FSI-CFD in Digital Twins*, starting from a quick overview on Computational Fluid Dynamics and Fluid Structure Interaction analysis, the issues encountered when trying to fit at runtime a numerical simulation with a CPD or DT are highlighted. Consequently, a description of a possible role for high definition, multiphysics simulations in the CPS and DT paradigms is given, unlocked by Machine Learning algorithms and Artificial Intelligence.

- In Chap. *Mathematical modeling and methodologies in FSI simulations* an overview of the mathematical, numerical and physical model implemented in the developed software FEMpar is provided. The review will take place from the basis of the Finite Elements Methods, and then the formulation for every system of differential equations required for a consistent FSI analysis is described and discussed: the Residual Based Variational Multiscale formulation for the Navier-Stokes equation, the Total Lagrangian formulation of the principle of virtual works for the non linear elastic problem, the Solid Extension Moving Mesh Technique for the moving mesh subsystem, as well as the Generalized-$alpha$ second order time integration scheme. Additionally, a final consideration about the coupling scheme will be provided, to give a better insight on what has been already discussed in the previous chapter;

- In Chap. *FEMpar: a custom FSI simulation software* a detailed description of the developed software FEMpar is provided, where the main components of the

software are illustrated: in particular, every stage of the flow of the software is examinated, from the preprocessing tools, to the main solvers. Furthermore, additional detail on specific features of the software are provided as well;

- In Chap. *Applications*, several results of numerical simulations of Fluid Structure Interaction problems in turbomachinery are given, all obtained with the software FEMpar, and with the described models.

- In Chap. *Conclusions* a summary of the main outcomes of the dissertation are provided.

2. Cyber Physical Systems and Digital Twins in turbomachinery and CFD

In this chapter will be introduced a brief overview of the context of the Cyber Physical Systems and the Digital Twins, and the reasons why they're being extensively studied and modelled in the turbomachinery industry: from predictive maintenance to design exploration, from performance optimization to surrogate modeling, a wide variety of applications in turbomachinery can benefit from the paradigm of the Cyber Physical Systems, and more specifically from the Digital Twin.

2.1 Cyber Physical Systems, an overview

Cyber Physical Systems (CPS) are a set of interconnected parts in which a physical device to monitor, control or coordinate is embedded with computing, communicating and control capabilities [1]. CPS research is focusing on the abstraction and modeling of the concept of CPS itself, and its applications, which covers a wide variety of scientific reaserch areas, such as automotive [2], bio-engineering [3], health care [4], transportation systems [5], energy grid management [6].

Several definitions have been proposed over the years to define and describe a Cyber Physical System:

- **Lee et al.** [7] : "integrations of computation with physical processes";
- **Rajkumar et al.** [8] : "physical and engineered systems whose operations are monitored, coordinated, controlled, and integrated by a computing and communication core";
- **Poovendran et al.** [9] : "a promising new class of systems that deeply embed cyber capabilities in the physical world, either on humans, infrastructure or platforms, to transform interactions with the physical world";

- **Gunes et al.** [10] : "a term describing a broad range of complex, multidisciplinary, physically-aware next generation engineered system that integrates embedded computing technologies (cyber part) into the physical world";

According to Rajkumar et al. [8] the reasons for the rise in the interest, design and adoption of CPS in many fields can be explained as a combination of *pushing* and *pulling* factors:

- ***Pushing factors***: miniaturization and proliferation of low-cost and high capability sensors; increasing availability of efficient and small computer devices; abundance of wireless, large bandwidth connections; improved systems for energy management, harvesting, gathering;

- ***Pulling factors***: the technology basis required to have CPS in most of the adoption fields are seriously lacking, and the CPS vendors and designers are struggling in the process of realizing their concepts for a wide adoption;

Regardless of the variety of definitions, the core aspect of the CPS, that can be considered the defining characteristic and consequently its true definition, is the coupling of cyber capabilities and physical processes. Such coupling relies on a small amount of core concepts, which must stand all together and converge into the the CPS model. A CPS model should always include an *Embedded System* (ES), which is a computational system, with all its components, closely embedded in a physical system. As a consequence of the presence of the ES, and of the coupling between cyber and physical domains, a CPS is often represented by a *Hybrid system*, i.e. a class of mathematical models that can manage both continuous and discrete domains, to match the continuous signals from the physical device with the discrete domain of the computational systems. The computational portion of the CPS can be composed of several interactive computational objects, one of each specialized in a specific task, or with a precise objective, constituting a *Multiagent system*. In addition to that, the CPS must be a *Reliable system*, designed to perform continuously in spite of the failure or the malfunctioning of one of its components. Furthermore, CPS must be *Real time systems*, as they should be able to operate at least at the time scales which are more representative of the underlying physics of the physical device they're containing.

To better understand the definition of CPS within the specific field of turbomachinery and energy production industries, in Fig. 2.1 is reported an example of a CPS for wind turbines from Kunzemann et al. [11]. In the example, a wind turbine (or a wind farm) is connected to a central computing unit ("central analyzer and optimizer" in the figure) using multiple connections: the red dash line represents the stream of energy output of wind turbine(s), and its linked to the central unit via a demand predictor,

and an energy controller; the continuous black lines represent the data streams, and connect the turbine(s) to a supply predictor and an operation analyzer.

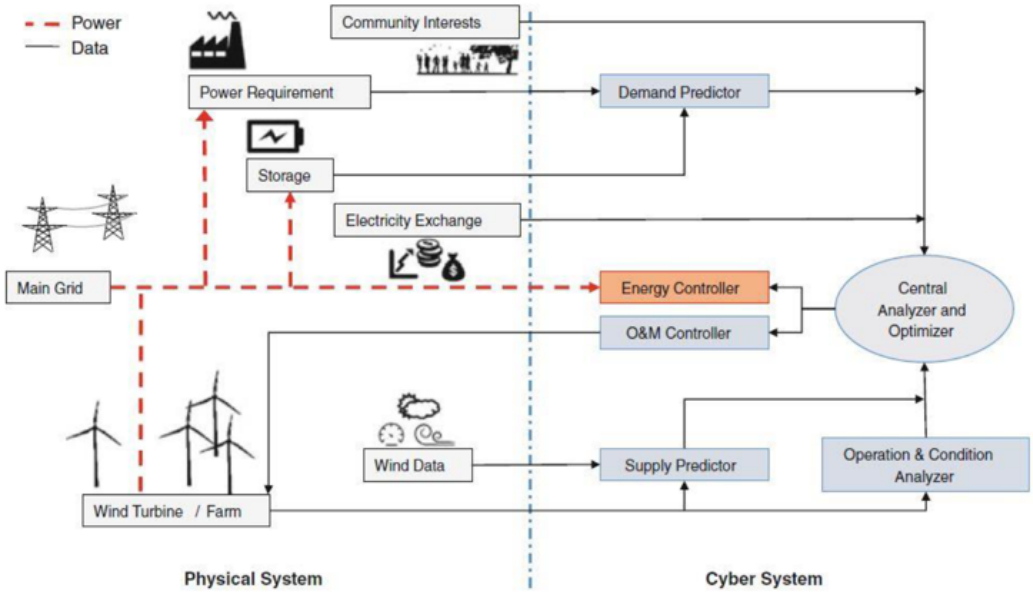

FIGURE 2.1: A CPS layout scheme applied to a wind turbine or a wind turbine farm by Kunzemann et al. [11]

In spite of the large research effort, the CPS paradigm is still facing several challenges. Some of the most important and crucial aspect that researchers are trying to solve and improve are regarding the safety and security of those systems (i.e. external attacks, disruptive failures), real time computational strategies, innovative control solutions for hybrid systems, smart data storage and management (the amount of data obtained by the autonomous sensors can be very large), validation and certification (standardized procedures shall be established to ensure an homogeneous adoption of CPS).

2.2 Digital Twins, an overview

A Digital Twin (DT) can be described, in general, as a digital equivalent of a physical product, composed of a physical entity (PE), its virtual counterpart or virtual entity (VE), and the data exchange between those entities. The main idea behind the DT is that the VE can be linked to the PE for the entire lifecycle of the product, and act as an effective mirror (a twin) of the PE, from which would be much easier to gather and process data, and predict emergent behaviours. Its origin is attributed to Grieves [12, 13], as a model for the Product Lifecycle Management, and further developed for aerospace vehicles and aircrafts [14, 15]. In its works, Grieves proposes a set of useful definitions, adopted to effectively describe each specific manifestation of the DT:

- **Digital Twin (DT)**: a set of virtual information configured to ideally represent a physical product in all its scales, all its behaviours. A perfect Digital Twin should be the one from which it's possible to obtain the same information that an observer could get by inspecting the real product. A digital twin can be a Digital Twin Prototype (DT) or a Digital Twin Instance (DTI) and they operate in a Digital Twin Environment DTE);
- **Digital Twin Prototype (DTP)**: a DT that describes a the prototype of physical artifact and its components, its materials, services and disposal;
- **Digital Twin Instance (DTI)**: a DT that describes a specific physical product, and it contains the unique information of that specific product;
- **Digital Twin Environment (DTE)**: the space in which the DT would operate, composed of integrated multi-domains physics, coupled in different ways;

In addition to the previous characterization of the DT, it can be useful to differentiate the DT by the different purposes that a DT could serve:

- **Predictive DT**: future behaviour and performance could be predicted by a DT, both as a DTP (for example by exploring the range of design tolerances that a specific instance of the product could express) and a DTI (by predicting a specific behaviour for a specific product);
- **Interrogative DT**: a DTI could be interrogated with respect to its current and past state in place of its real counterpart, regarding of its location and accessibility. Aggregates of the data obtained by interrogating the DT could provide, for example, useful correlations between sensors to build alerts on possible incoming failures.

With respect to the definition of the CPS, it is possible to assert that a DT is a subset of the CPS, and that many characteristics are in fact in common: for example, they both have a computing unit that performs algorithms and describe the current and past state of the physical counterpart; they both are driven by a closed loop control, where sensor's data are processed and sent back to actuators or to a terminal for decision making. More analogies could be found when referring to specific instances of both the DT and CPS, but a complete characterization is beyond the scope of the present work, and it can be found in literature [16]. An example of a layout for a Digital Twin architecture (deeply embedded in a more complex CPS) can be found in Fig. 2.2:

Fig. 2.2 shows the three main macro components of the CPS: the physical world (or *physical sphere*), composed of Physical Entities (PE), for example sensors, Radio-frequency identification (RFID) tags, embedded systems, and actuators; the cyber world

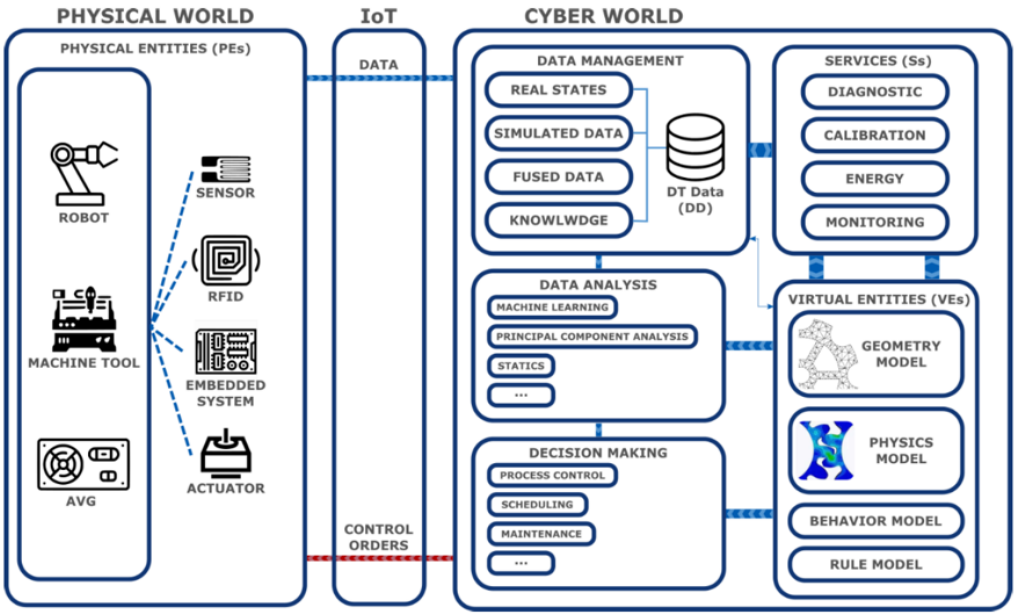

FIGURE 2.2: An example of a DT architecture

(or *cyber sphere*), composed of a large amount of logic blocks, each one dedicated to a specific task or objective; the internet of Things (IoT) that connects the physical world to the cyber world in both senses: it allows data to flow from sensors to the computing units, and it sends back the orders from the cyber world to the actuators. The blocks in the cyber sphere can perform different tasks, and in general it is not uncommon to recognize a set of most recurring components, as representend in the figure:

- **Data management block**: a server-like structure manages and stores the data obtained by both the physical sensors and the DT;
- **Data analysis block**: the stored data can be further manipulated and analyzed using different algorithms and methods, such as machine learning algorithms, statistics models etc;
- **Decision making block**: the data analysis allowed the software to take a decision with respect to the output of the analysis, and a specific command is sent to the actuators;
- **Services block**: a set of accessory tools and services that can help the user, or the operator, to understand the status of the DT and the CPS in general obtaining real time data regarding the on-going processes and the status of the CPS as a whole;
- **Virtual entities block**: It represent the digital counterpart of the PE, the DT itself, that is synchronized with the PE while it is operating. It is the main source of the

digital data in the CPS, and the data originated by the simulations are stored in the Digital Database, and used in real time to make decisions;

2.3 Digital Twins in Turbomachinery: challenges and applications in a lifecycle perspective

From the previous sections, it is easy to imagine that a generic Digital twin model, and the technologies required to implement it, could perfectly fit in the turbomachinery industry. When considering the entire lifecycle of a generic turbomachinery-related product (a turbine, an aeroengine, a specific component, a power producing plant or a power network system) it is possible to recognize the suitability of a Digital twin model to represent or support each specific phase of the lifecycle of the product, from the design stage to its disposal and recycle. Thus it could be theoretically possible to design a specific DT not only for each stage of the lifecycle of the product, in a horizontal direction, but also varying from the single component, to the product, up to a system of systems, in a vertical direction. This wide applicability of the model drastically increases its appeal from the perspective of every agent that participates in the lifecycle of the product: the manufacturer could reduce its design and manufacturing costs and time, the user could increase the reliability of a purchased product and eventually reduce maintenance and disposal costs. With this premise, it is much easier to recognize the significance of the possible improvements that could be reached in the research and developent, quality control, operations and predictive maintenance, by gradually adopting the use of DT in turbomachinery. More specifically, it is possible to subdivide the lifecycle of a generic turbomachinery product in five consecutive and cyclic phases [17], as in Fig. 2.3:

- ***Design phase***: at the design stage the product requirements are examined and an initial design is given, followed by an iterative study. At this stage, the definition of DT can be adopted even if a physical counterpart might not be available yet. However, when considering the cyclic nature of the lifecycle of the product, it is possible to imagine that a previous product could be adapted to initiate a new design operation, or its data could be analyzed *post-mortem* by a new virtual entity. In any case, the presence of a DT at the design stage could drastically reduce the time required to explore a large number of different designs and operating environments, by giving a quick feedback to the designer evaluating the performances of virtual products. An example application of DT in turbomachinery at the design stage is proposed by Baldassarre et al. [18] where they redesign an existing wind turbine using a DT;

- ***Experimental phase***: at the experimental stage the prototype is tested in all of its crucial aspects (e.g. performances, safety, reliability), but the high costs and high time consumption characterizing this stage are notorious. In fact, the set up of an experimental campaign on a real prototype requires a large amount of hardware involved (e.g. sensors, power gathering, data acquisition and transmission) of the highest quality, and usually long operating time, to ensure that the gathered data are as meaningful and rich as possible. At this stage, a DT could easily cut to the minimum the time required to acquire meaningful data and to reduce the unproductive time; virtual tests with DT could also provide higher definition data by using complex models, and due to the fact that virtual tests are not limited to a finite number of fixed-position sensors, they can provide data even where sensors could not be possibly placed due technological or operating constraints;

- ***Manufacturing phase***: at the manufacturing stage, a DT supported by a deeply interconnected set of sensors enabled by the Internet of Things (IoT), could provide useful optimization of both the manufacturing and assembly process, as well as the plant layout, even providing quick decision-making support for dynamic operations [19];

- ***Operation and maintenance phase***: at this stage, the current implementation and research of DT in turbomachinery flourishes, by gradually accompanying the shift occurring from reactive maintenance to proactive maintenance, reducing overall costs for the maintenance and failure occurrences, while still *de-facto* extending the lifespan of assemblies, products and components. When the product is delivered to the user, it can be packed together with an identical virtual copy, its Digital Twin, connected with sensors to its physical counterpart: the DT would constantly gather and process the data from the sensors mounted on the real device, and can give insights about faults or raise alarms for occurring failures and with enough anticipation reduce disruptive and expensive damages, as showed for example by Seshadri et al. [20] for a wind turbine. In this phase of the lifecycle the DT can be seen as a massive improvement of the Health and Monitor Systems of the product, by increasing the capability of the system with predictions, by ensuring a quicker response interventions, and by extending the quantities and phenomena monitored without increasing or changing the installed sensors (a DT, with its computing unit, sophisticated models, data and simulations could easily and reliably derive or interpolate additional variables without directly measuring them);

- ***Disposal or recycling stage***: at the end of life of the product, an already existing Digital Twin (or a Digital Twin developed *ad-hoc*) can serve as a reference for

further development of similar product, constituting an interactive library for a previous iteration of the device. A Digital Twin designed for a device or system no longer in service could still drastically accelerate the design process of another product of the same type, and improve the performance of the next one while at the same time reducing the costs of the design stage. In addition to that, clear advantages in terms of costs and space can be found when considering the differences between keeping in service an old machine at the end of its life for research purposes, and its digital counterpart (more specifically, the Virtual Entity associated with the physical device).

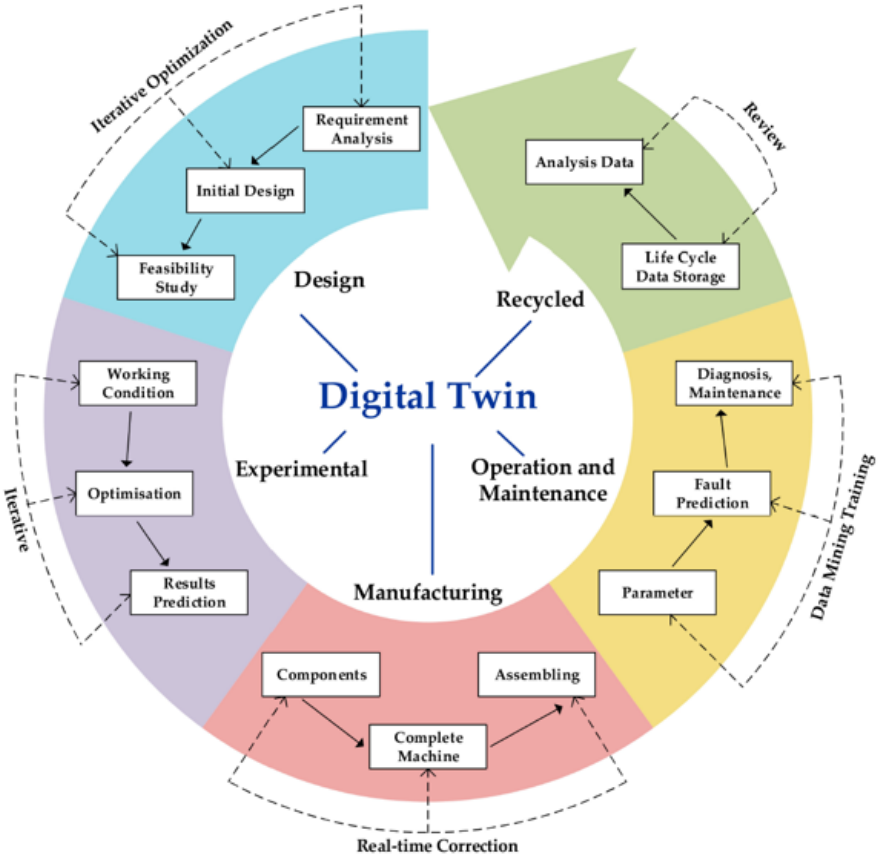

FIGURE 2.3: A generic lifecycle representation for turbomachinery with digital twin

As previously mentioned, when referring to the applicability of Digital Twins in turbomachinery it is also possible to differentiate the possible implementation of digital twins happening at different levels of the same product or technology. While not being the only case at which it is possible to have this characterization, the conversion cycle of wind energy can be take as a particularly fitting example to explain this idea. When considering wind energy, it is possible to implement a DT (or more in general, a CPS) at different levels. Whichever is the level at which a DT is implemented,

it could still accomplish the already mentioned wide variety of different objectives (design optimization, predictive maintenance etc.) at every stage of the lifecycle. The aforementioned levels could be schematized as follows:

- ***Component level***: each significative component of a wind turbine or the power network could be paired with a Digital Twin, regardless if the intention is to build a digital assembly or a digital network system, or to have an individual Digital Twin for that specific component. Significative components could be, for example, the wind turbine blade, the nacelle, the pitch control system, the generator, the mast;

- ***Assemble level***: a DT could be developed for the entire wind turbine, by linking together several modules, one for each specific component, which is describing the component behaviour with a mathematical model of unspecified complexity. However, due to the necessity to operate at real time, Reduced Order Models (ROMs) are mandatory when dealing with extremely complex behaviour, such as fluid dynamics or fluid structure interaction;

- ***System level***: a wind farm could represented as a DT by creating a communication network between the wind turbines so that the devices could share information valuable at that level (e.g. incoming wind gusts, operation parameters of nearby turbines);

- ***System of systems level***: a power grid (or a smart grid) could be linked to a Digital Twin to mitigate the physiological fluctuations of the power supply, if supported by a weather forecast system that gives feedback in real time.

With the objective to give a more specific example of a DT applied to turbomachinery, Fig. 2.4 shows a CPS with DT that aims to estimate fatigue life of a wind turbine: in the cyber sphere are found the DT (in this case is imagined a DT built with Ansys) used to estimate thee fatigue life and to constitutes a virtual database; in the physical sphere the sensors and the Supervisory Control and Data Acquisition system (SCADA) are gathering and sending data to the same backend software that interacts with the DT via the database. In the example, the DT is used to build a database that will be further manipulated by the backend software to estimate the fatigue life, when the inputs from the sensors and SCADA are given.

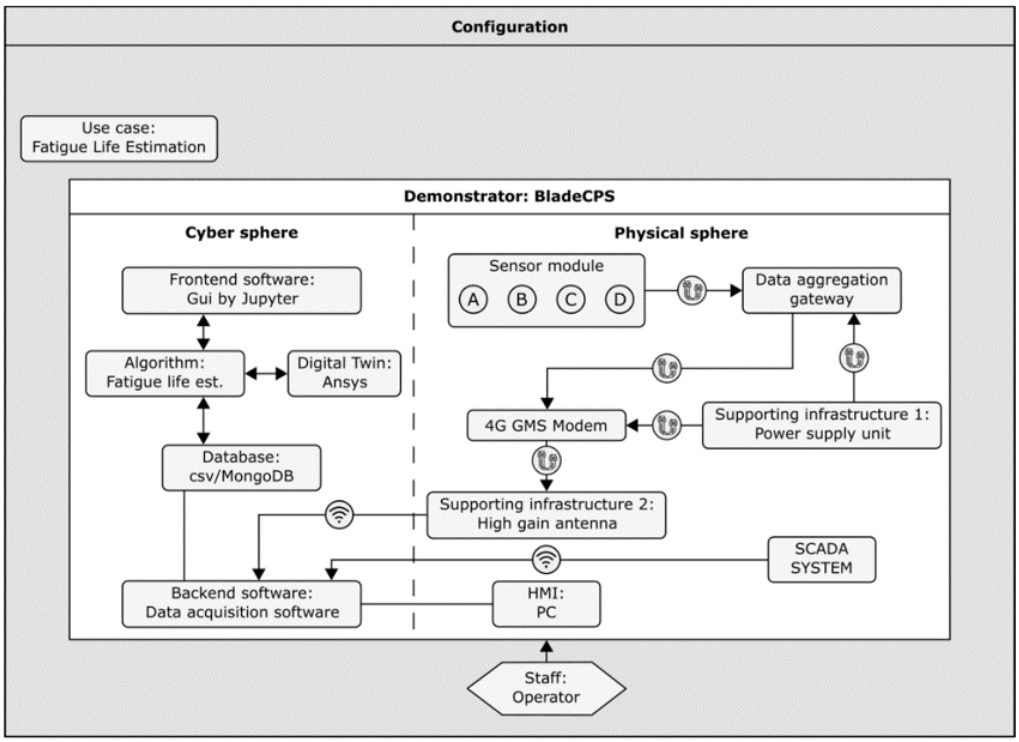

FIGURE 2.4: An example of a CPS with an embedded DT for the fatigue life estimation in wind turbines

3. The role of FSI-CFD in Digital Twins

3.1 An overview of CFD

In the last 30 years, the advances of computational fluid dynamics have been astounding, both in terms of fidelity and resolution, and in terms of performances. Several progresses have been made in many of the most challenging fields related to the numerical solution of the Navier-Stokes (NS) equations: the amount of different techniques, strategies and models proposed by the researchers in the last decades is still growing, and is far from being stagnant.

In general, a CFD simulation can be defined as the resolution of a discrete approximation of a system of equations of the family of the Navier-Stokes equations using numerical methods, to obtain precise knowledge of several quantities (usually velocity of the fluid, pressure, temperature and derived quantities) on the discretized domain. It is almost natural to see, from this definition, how wide and vary can be the field of CFD: several factors would intervene into constituting a proper flow simulation, and each combination of these factors represents a branch in the field of CFD. In fact, it is possible to characterize a CFD simulation by the system of equations that are being solved (e.g. compressible or incompressible NS equations, Stokes flows, Newtonian or non-Newtonian fluids, fluid-structure interaction), by the discretization adopted (e.g. finite differences, finite volumes, finite elements), by the presence of time variable in the equations (steady, unsteady), by the turbulence model adopted (one or two equations closure models), by the presence of a specific wall treatment, or the presence of a stabilization scheme, and also by the resolution of the mesh and consequently by the solution approach adopted (e.g. Reynolds Averaged Navier Stokes-RANS and related methods, Large Eddy Simulations-LES, Direct Navier Stokes-DNS). From this extremely simplified characterization, the variety and complexity of the CFD field in general is recognizable, and for such reason only a brief overview of the most common practices adopted in the field of the turbomachinery would be highlighted here, and especially from the perspective of the manufacturer.

It is no secret that, from the manufacturer point of view CFD is a reliable tool, a source of information to evaluate the performances of a specific design or feature of a

device, that could be otherwise available only after an extensive (and expensive) experimental campaign: the producer would have to account for the time consumption and costs of the prototype production, the experimental campaign, the data acquisition and analysis. At the current stage of the turbomachinery (and not only) industry, the aforementioned designed process is no longer adopted, and basically considered not affordable, and not needed for every new prototype or design proposal, at least at the early stages of the lifecycle of the product. In fact one of the most common design processes adopts CFD simulation as (almost) a complete substitute of a preliminary experimental campaign: a new device, or feature, is digitally reproduced using Computer Aided Design (CAD) software, and simulated in its core functionalities using CFD.

According to the aforementioned objective of the producer to always improve and optimize its production process, by constantly reducing time consumption and costs, the CFD simulations that are valuable for the manufacturer are the ones that are using faster solution strategies with the least amount of resources, while remaining consistent with the physics, even if sacrificing the resolution of every scale of the phenomena. It is for this reason that the most widely and common methods adopted to solve the NS equation are the RANS (Reynolds Averaged Navier Stokes) and URANS (Unsteady Reynolds Averaged Navier Stokes) models, often using turbulence modeling with one or two equation systems (such as SA [21], $\kappa - \epsilon$ [22], or $\kappa - \omega$ [23]) and a wall function at the wall [24]: such a combination is able to provide fast results using a relatively low amount of computational resources, with relatively coarse meshes so it is possible to model the lower scales phenomena (boundary layer resolution, small scale turbulence) using analytical equations provided by *ad-hoc* developed constants. Such a simulation could be handled by a personal computer, even a laptop, using a commercial (or open source) software, and carried on until convergence in hours or in a few days of computation. Other methods such as LES, are rarely used, in spite of the fact that they are able to reproduce with more fidelity a wider range of phenomena, as they require a finer discretization to do so, and consequently, more computational resources, and much more simulation time. DNS on the other hand, is immensely consuming both in terms of computational effort and computational time, and it is basically never used during design and production of turbomachinery, while its sometimes adopted by researchers to constitute extremely high fidelity data sets and benchmarks. Further information regarding RANS, LES and DNS, as well as a complete and comprehensive overview of several turbulence models can be found in **??**.

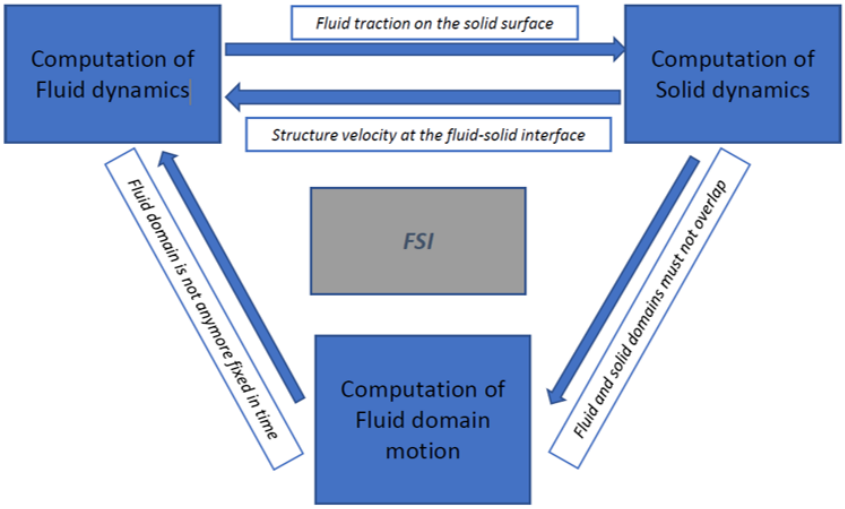

FIGURE 3.1: The logical components of a FSI problem: fluid dynamics, structural dynamics, fluid domain motion

3.2 An overview of FSI

The definition of a fluid-structure interaction problem (FSI) can be easily given without losing any generality. When looking at a natural phenomenon, or a technological device, where it is possible to clearly identify a solid component and a fluid component, and where, at the same time, there is reciprocal dependence between the solid part and the fluid part, it is possible to characterize that problem as a fluid-structure interaction problem. With "reciprocal dependence" it is specifically intended that the evolution in time of the shape, displacements, state of deformations and stress in the solid structure depends on the flow behaviour and the forces exchanged at the interface, and viceversa, the velocity and pressure fields of the fluid near the solid structure, and its dynamics, depend on the motion and dynamic behaviour of the structure. From the perspective of the subcomponents involved in the phenomena, it is a problem with three components: in fact the modifications in the shape of the fluid domain must also be taken in account, and for this reason an evaluation of the motion of the fluid domain must also be performed, as schematized in Fig. 3.1. From this general definition, it is quite easy to assert that a large number of problems could be included in the class of FSI problems, and lots of different applications of the FSI principles have been extensively applied in a wide variety of research fields: simulation of flapping wings [25], design and optimization of space parachutes [26], cardiovascular and blood-vessel simulations [27], simulation of large scale wind turbines [28], turbomachinery with passive-adaptive components [29, 30] or with a particularly slender design [31], simulation of erosion from the environment on wind turbines [32]. In spite of the large variety of the applications that can be described with FSI simulations, they all have in common

the core principle of FSI analysis: two (or more) different material phases are modelled by different equations, and they communicate with each other at the interface between the fluid and solid domains, by exchanging forces and by sharing the displacement at the interface. The overall model of the observed phenomenon will be strongly non linear and time-dependent in every aspect, and therefore an analytical solution for FSI problems of practical interest is quite hard to obtain, and for this reason, extremely rare: eventually an analytical solution can be obtained only for simplified applications with the aid of simplifying assumptions. On top of that, FSI problems do not scale together with the spatial scale: the dynamics of a FSI phenomena drastically changes its behaviour when changing the size of the problem, for example when passing by a full size device to its scaled model, an accurate collection of experimental data is not replicable in laboratory: therefore, a numerical simulation allows one to test full scale phenomena with ease.

For these reasons the furthest advances in FSI analysis have been obtained from computational FSI, in a constant strive to develop software and algorithms and numerical methods to reduce computational effort required by the simulations and increase the robustness and accurateness of the numerical results. It is not uncommon, for a FSI simulation software developed in compliance with the state of the art technology of computational FSI, to be able to accurately model the entire multi-physics FSI phenomenon at full scale, for complex geometries, with a sufficiently high definition to observe the insurgence of small scale fluid dynamics (i.e. turbulence) or high frequency vibrations.

3.2.1 FSI challenges

Unfortunately, the challenges encountered when setting up a proper and effective FSI computational simulations are built on top of the existing ones in the single-field problems (fluid dynamics simulations with CFD and structural dynamics simulations with FEA): the difficulties related to the selection of the proper model to represent the underlying differential equation, the compromise between the resolution of the discretization and the amount of computational sources for the simulation, the convergence issues, the proper setup of the essential and natural boundary conditions and the difficulty to effectively discretize complex geometries, are all found in FSI as they are found in CFD and FEA.

Domains management and reference frames

Usually in FSI the different computational domains are non overlapping, and a set of interface conditions of kinematic compatibility (the velocity of the fluid and the solid

must be the same at the interface) and traction continuity (the forces must be coherently transferred from the two different surfaces of the domains, at the interface), shall be set accordingly to the respective reference frames. The reference frames themselves have to be carefully chosen to accommodate the discrepancy between the consolidated habit of adopting a Lagrangian reference frame for the structural problem, and an Eulerian reference for the fluid dynamics, and the equations have to be adapted to this hybrid paradigm called Arbitrary Lagrangian Eulerian (ALE) framework.

Remeshing vs moving mesh

As previously anticipated, there's the additional problem of the mesh morphing (i.e. the requirement to evaluate the motion, the displacements, and the change in shape of the fluid domain), usually handled by moving mesh algorithms or by automatic re-meshing, to ensure the continuity of the discretization at the domain regardless of the large displacement that could occur in the structural domain, which is usually obtained by solving an additional subsystem, together with the fluid subsystem and the structural subsystem.

Interface discretization

The discretization of the interface itself represents a supplementary challenge: matching domains at the interface, for example, represent an easy and efficient solution from the implementation perspective, with no additional interpolation algorithm required, but they lack the flexibility that a non-matching interface solution algorithm could provide during the meshing phase, by assigning different discretization and different refinement levels to the different domains.

Time discretization and physical coupling

It is also crucial to mention the additional problem of the choice of the solution algorithm adopted to couple the subsystem from the perspective of the temporal discretization. In fact it is possible to have loose (or staggered) and strong (or monolithic) coupling between the fluid, structural and mesh morphing subsystem, when considering the time discretization. In the weak coupling the subsystems are not solved in the same time step, but sequentially: a solution is obtained for the fluid field, where the boundary conditions are directly derived from the displacement of the solid; then the new solid displacements are evaluated by stressing the structure with the forces derived from the interaction of the fluid at the interface; at the end, the mesh is updated accordingly to the structural displacements. In the strong coupling, on the other hand,

all the three subsystems are solved in the same time step, and the exchange of information at the interface is completed before the next time step. In general, it is safe to assume that a loosely coupled approach is more flexible from the point of view of the implementation, because it allows the use of different software or solvers to solve each subsystem (as long as effective, custom communication strategies and interpolation algorithms are implemented as well), but it suffers convergence issues under (not so uncommon) specific circumstances, such as a large mass imbalance between the solid and the fluid, or with fully enclosed flows. Strongly coupled algorithms, on the other hand, require a larger implementation effort due to the fact that an integrated FSI solver have to be developed, but they show in general a more robust convergence.

Numerical coupling

In spite of the implementation challenges provided by the strong coupling techniques, they could offer some form of flexibility when looking at the available algorithms to solve (and consequently couple) the linear algebra problems represented by each subsystem: when adopting the strongly coupled approach, it is possible to have a block-iterative coupling, a quasi-direct coupling, and a direct coupling. In the first strategy, the subsystem are linearized and solved sequentially within the same time step, as three separated linear problems; in the second strategy, the equations for the fluid dynamics and for the structural dynamics are included in the same matrix, while the mesh morphing is solved by a separated matrix inversion; in the third strategy, the FSI problem is written in a unique, large matrix, where every degree of freedom of each domain is directly connected with the corresponding ones in the other domains. The more the coupling in the matrix between degrees of freedom, the more robust the computational FSI simulation will be. However, ad-hoc preconditioning strategies and matrix entries ordering have to be developed in order to ensure convergence for the full matrix, which would otherwise be inconveniently sparse and particularly hard to invert, while a block-iterative approach represent a compromise between robustness and implementation difficulty: each subsystem can be solved independently, with the most suitable preconditioner and numerical solver.

Additional details on the algorithm adopted for the development of the software presented in this work can be found in Chapters 4 and 5. A more exhaustive description of the FSI paradigm, the solution strategies, its challenges and applications, can be found in the book by Bazilevs, Takizawa and Tezduyar [33], which is the main reference adopted for this work, for the development of the FSI simulation software developed, and for the applications presented in Chap. 6.

3.3 Connections between FSI and DT

By recalling some key concepts from Chap. 2, it is possible to identify in the FSI paradigm a number of features that could easily suit the requirements of a complex CPS or DT, and that the FSI paradigm itself could serve the purpose of constituting a key component of the Virtual Entity of the DT: the definition of the Digital Environment (DE) is a natural consequence of the interconnected layout of communications at the interface between the solid and the fluid domain; the Multi-Agent and multidisciplinary nature of the CPS is perfectly embodied by the contemporary presence of two different physics domains, governed by different systems of equations; the fact that a successfully designed, optimized, reliable software for FSI analysis could easily be adapted to be part of a Digital Twin, by direct or indirect implementation, without the necessity to completely redesign or develop a new software, or a new communication strategy, or a new solution algorithm. A multi-physics Digital Twin (specifically, a fluid structure interaction based Digital Twin, or related) can directly inherit the software structure, the software layout, and the optimization practices, directly from the well established experience of computational FSI.

3.4 Issues on the implementation of CFD and FSI in Digital Twins

The main focus of this section will be directed towards the interaction between CFD or FSI simulations, and the concept of the Digital Twin: why they can interact with each other, how the interaction can be made, which are the main issues of the interaction.

An ideal attempt to pair a CFD simulation with a Physical Entity (for example, a turbine) would be to gather the data obtained by a series of sensors mounted on the device and send the data to the computational unit, where the simulation software would use the data as an input (transmitted data could be the entire input set, or a subset of inputs) to observe (or enhance) the flow behaviour and derive a set of outputs by means of the simulation, which would be otherwise impossible to evaluate with the same degree of precision and fidelity, or even impossible to evaluate at all.

3.4.1 Limitations in real time applications

However, from the description given so far of CFD and FSI techniques in general, it appears almost impossible to find a niche for this technology in the Digital Twin paradigm, and Cyber-physical systems. In fact, one of the fundamental requirements mentioned in Sec. 2.1 when describing the CPS paradigm, was the requirements of

real time interaction between the Physical Entity and the Virtual Entity. As reported in Sec. 3.1, even when considering the less computational expensive solution strategy on a coarse mesh, it is almost impossible to have a real time interaction between a rotating machine and the computational unit in charge of carrying on the simulation. In fact, it is common that rotating machines experience high velocities in the guide vanes: smaller devices are often required to operate at high rotating speed to have the larges possible power output, while larger machines, in spite of lower rotating speed, still have high velocities at close to the tip of the blade, where the tangential component of the velocity is prevalent. In any case, turbomachinery can be safely considered devices which operate at high Reynolds numbers, and consequently, both the spatial and temporal discretization must be designed accordingly to the specific flow behaviour which one intends to simulate.

In general, it is safe to assume that for a sufficiently fine mesh and for an adequate time discretization, in a conventional CFD simulation using reasonable resources, *the time required to perform a simulation to observe the dynamic of a complex flow is in general larger than the observation time*. This claim has to be intended as a general observation and not a theorem, as specific results under specific circumstances may invalidate the sentence.

Nonetheless, there is currently a widespread effort by the research in the direction of eliminatingthe obstacles in the implementation of CFD and FSI simulations, as they are commonly used in the industry, in the Digital Twin paradigm. A possible solution which is emerging from some very recent papers would be to design a Digital Twin where the Virtual Entity is not directly performing a CFD flow simulation in real time, but instead it's processing the data obtained by the sensors using a system of Reduced Order Models obtained by a Machine Learning (ML) algorithm. Such models can be directly derived by the large amount of CFD data, and are able to evaluate a set of outputs which are comparable to the ones obtained by the CFD, differing in general for a small error. The presence of the error is traded for a reduction of the computational time by several orders of magnitudes, *de facto* unlocking the potential of the high flexibility and fidelity of CFD techniques for the Digital Twins. In Fig. 3.2 is presented the comparison between a direct, desirable, but not feasible implementation of a CFD simulation in DT, and an indirect implementation.

3.5 Artificial intelligence to unlock CFD and FEA in DT

Machine Learning (ML) proved itself to be a promising technology for data processing and analysis, especially when considering the large amount of data available from almost every digital sources, or electronic device. A wide variety of possible techniques

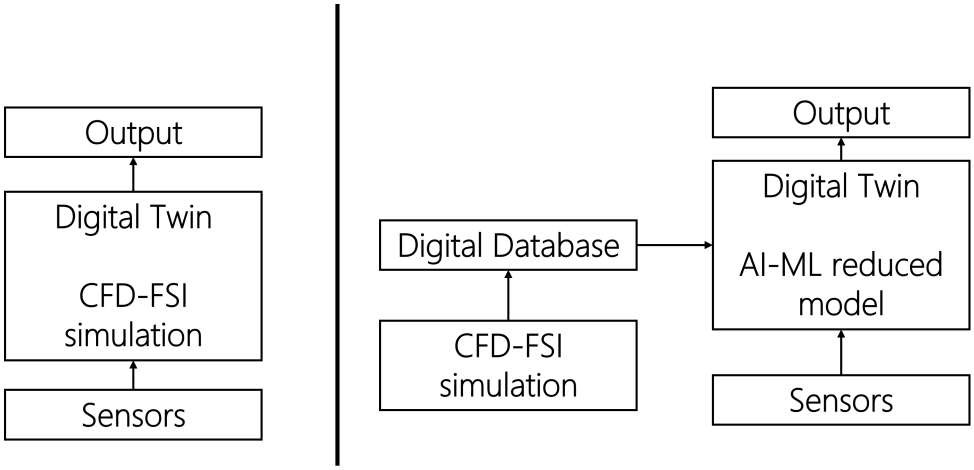

FIGURE 3.2: A comparison between an ideal (but not feasible in general) CFD implementation in DT, and a more realistic (but more complex) one using big data.

have been developed to efficiently and effectively process large amounts of data coming from very different sources. An exhaustive description of the most adopted and relevant techniques for ML, or a complete review of the ML advances are outside the scope of the present work and therefore they will not be reported. However, a complete description of ML techniques can be found in [34].

When considering the application of Machine Learning techniques to CFD, typical applications reported by a recent review from Brunton et al. [35] are often towards increasing the performance of the models or the computational performance of the simulations, or, on the other hand, towards design and topological optimization and flow control. In the first case, a dimensionality reduction is often performed on the data set, so the resulting reduced model is able to replicate some specific behaviours of the flow dynamics if the goal is to extract particular behaviours or flow structures [36]; when instead the aim is to predict the dynamics, in an attempt to bypass or accelerate CFD, the ML algorithm is designed to learn directly the solutions of the differential equations that define the observed system [37, 38], improving the simulation from different perspectives and with different strategies: improving convergence rate, increasing computational performance, optimizing the simulation parameters and improving closure models [39, 40, 41, 42, 43, 44, 45]. In the second case, optimization and flow control can be obtained by using ML algorithms on high fidelity CFD data, for example by constituting surrogate models to explore unconventional design spaces or to perform topology optimization to increase the performance of a device [46, 47, 48].

From the perspective of the structural analysis using Finite Elements (FEA), several alternative approaches to the implementation of ML in FEA simulation have been recently proposed, each one intervening at different levels of the simulation: from

the substitution of constitutive equations [49, 50], to the implementation of nonlinear structural model derived by data [51], or the optimization of the FEA simulation itself [52, 53, 54, 55]. Other authors however, directly propose the resolution of the structural dynamics entirely using Machine Learning algorithms, of different complexity [56, 57]: one of the most promising ones, from Liang et al. [57], proposes the adoption of a deep learning algorithm composed of an encoder and a decoder, substituting the entire Finite element simulation of a shell structure (shaped as a thoracic aorta) with an almost instantaneous operation of image recognition, while still obtaining an almost identical distribution of stresses and strain.

3.5.1 Reduced Order Modeling for FSI in CPS and DT

A common approach to feed data into a machine learning algorithm to obtain a reduced, data-driven model would be to use data directly obtained by the sensors: data obtained by a real observation of a natural phenomenon are expected to be of the highest fidelity possible, and consequently, perfectly able to be processed by a ML algorithm in an attempt to replicate a law of the physics under particular circumstances. However raw data are usually not perfectly suitable to be used for the design and development of a data driven model: they're often characterized by a non negligible noise and a consuming process of data cleaning is mandatory, and the completeness and quality of the data set is strongly dependent on the quality, reliability and positioning of the sensors. As suggested by Molinaro et al. [58], a solution could be the development of data-driven models (DDM) using CFD: instead of raw data from sensors, DDM could be built using simulation data, organized in a Synthetic Flow Database composed by a set of Figures of Merit (which could represent a well defined variable distribution, and it depends on the specific field of application: it could be the distribution of the aerodynamic coefficients, or the pressure losses, for example). The ML algorithm will explore all the Figures of Merit related to the specific operating conditions and will be able to give a desired output bypassing the resolution of the fluid dynamics equations. The procedure can be compared to a more classical approach such as the Model Order Reduction [36], as they share their ultimate purpose but not the workflow or the applicability range.

With the above considerations, it is possible to find a niche for CFD simulations, FEA simulations and Fluid-Structure interaction simulations in Digital twins: they all rely on very high fidelity models, using fine discretizations but unfortunately the computational effort required to obtain a solution is high and in general the simulation is time consuming. For those reasons, it is impossible to directly match a CFD or FEA or FSI software with the high frequency inputs coming from a set of sensors mounted on

an operating turbomachinery and expect the simulation software to be able to produce a solution output in real time. However, it is possible to set up a set of reduced order models, or a set of machine learning algorithms, that are designed with high quality simulation data obtained by the aforementioned software, which can be implemented as the Virtual Entity (or one the Virtual Entities) of the chosen Digital Twin. Such model (or models) is supposed to be able to fit the requirements of the Digital Twin expressed in earlier sections, such as the real time computation capabilities, the ability to describe multiple scales, and the ability to describe multiple physical systems. The first requirement is a natural consequence of a well designed and implemented reduced order model or surrogate model, while the second and the third one are a direct function of the quality of the models and solution strategies chosen for the CFD, FEA, or FSI analysis simulation software: the higher the richness, the resolution and quality of the data obtained by the simulations, the smaller would be the error for the output of the alternative model produced by machine learning.

3.6 Possible outcomes of FSI in DT

As a final remark of this section, it is useful to further investigate the implications of the adoption of FSI analysis in turbomachinery, and its role with respect to the digital twin technology, by means of the application examples provided in Chap. 6. The applications themselves will be discussed in much more details in the above mentioned chapter, however, it is useful to briefly anticipate the analysis and simulations that will be later presented, in order to emphasize the importance of FSI in specific turbomachinery niches, and also to specify how a DT could benefit the design and maintenance stages of a turbomachinery.

3.6.1 Multiphysics simulations provide multiphysics information

When considering the peculiarity and benefits of addressing an FSI analysis on top of a simple CFD analysis, the most immediate advantage is the increased amount of information that are available with a numerical simulation: on top of the simulation and representation of the unsteady flow field, a strongly coupled system can provide reliable information with respect to the structural dynamics, taking into account aeroelastic loads, and positive feedbacks that would be otherwise non predictable with traditional CFD, especially when considering non conventional materials, or blades with extreme aspect ratio. In Sec. 6.1.1, an FSI study is performed on 3D printed axial fan prototype, with particularly slender blades. It is shown how, for the same device, different operating conditions in terms of mass flow rate could provide completely different structural

dynamic response, up to possible disruptive conclusions. When using a traditional CFD approach, it is common to obtain useful information about the fan performances, efficiency, pressure distributions and pressure differences, drag and lift coefficients, turbulent structures, but with an FSI simulation it is possible to additionally obtain data regarding stress distribution, structural displacements and flow induced vibrations. A non-moving structure will always be limited in terms of the knowledge that can be obtained from an unsteady simulation: a blade characterized by a low stiffness material could be affected by structural displacements that would deeply modify the flow dynamics, and consequently the performance of the device.

3.6.2 Design exploration of passive morphing adaptivity

Passive flow control devices (such as flexible appendices at the trailing edge, in the tip region) have been extensively studied and applied in turbomachinery in the latest years, some of them with strongly coupled FSI solvers [29, 59]. A relative uncommon approach to passive adaptivity in turbomachinery is related to the design of devices characterized by completely flexible blades, built with hyperelastic materials described with a Neo-Hookean model [60]. Such design, inspired by boat sails and their ability to adapt to impulsive changes of the direction of the flow (see Fig. 3.3), is explored in Sec. 6.2.1 and 6.2.2: in the former, the behaviour of a reversible fan with a rotor composed of symmetric and flexible blades is observed; a 2D FSI analysis is performed on different material configuration and different constraints, to explore the possibilities and limitations of such design, and it brings up the large differences in performance and structural stability that are obtainable only by relaxing a constraint (and consequently by reducing the stiffness of the entire coupled system). In the latter, a Wells Turbine is redesigned from its original configuration using the same approach, by substituting the rigid blades with thin and flexible blades, using rigid clamps to constrain the blade at the hub.

3.6.3 Virtual sensors in prohibitive locations

When looking at small scale devices such as the ones proposed in Sec. 6.2.1 and 6.2.2, characterized in their functional core parts by completely flexible and rotating components, it is sometimes really difficult (sometimes even impossible) to place, connect, and maintain a large amount of sensors without compromising the functionality of the device, or the quality of the gathered data. As previously mentioned, this issue implies that some of the requirements for a functioning DT for such devices could be simply not fulfilled. An FSI simulation, on the other hand, could provide raw data (directly, or more reasonably, non-directly via a reduced order model) to a computing unit from

FIGURE 3.3: A boat sail can inspire the design of passive adaptive blades when the right materials are selected. On the left, the comparison between a rigid thin Wells turbine blade and its morphed counterpart, presented in Sec. 6.2.2

a large amount of locations, even the ones that could otherwise be not reachable by a physical sensor, without compromising the quality.

3.6.4 Data creation for multiscale and multiphysics problems

As previously stated, a strict requirement for machine learning algorithms (that could be used to connect the simulation data and the DT) is the amount of high quality data to feed the algorithm itself. As a rule of thumb, the higher the quality of the provided data, the lower the estimate error obtained by the algorithm, once properly trained. For such reason, the adoption of high order, LES-like, unsteady models (such as the RBVMS-Residual Based Variational MultiScale (RBVMS) model, for example, as will be properly described in Chap. 4) could help to constitute a reliable and rich database for further development of machine learning algorithms for fluid dynamics and FSI in turbomachinery. For these reasons are proposed in Sec. 6.1.1 and 2.1 two simulations using the RBVMS model in turbomachinery, one for the already mentioned 3D printed fan, the other for a large axial fan adopted in CSP plants to reduce water consumption in arid regions.

In addition to that, it is also possible to imagine that the database that could be built with FSI simulations is a multiphysics database, in the sense that it collects coupled information from different physical domains (the fluid domain and the structural, solid, domain): due to the inherently coupled nature of such a database, it would be useful to design a ML algorithm that could further reduce the dimensionality of the multiphysics

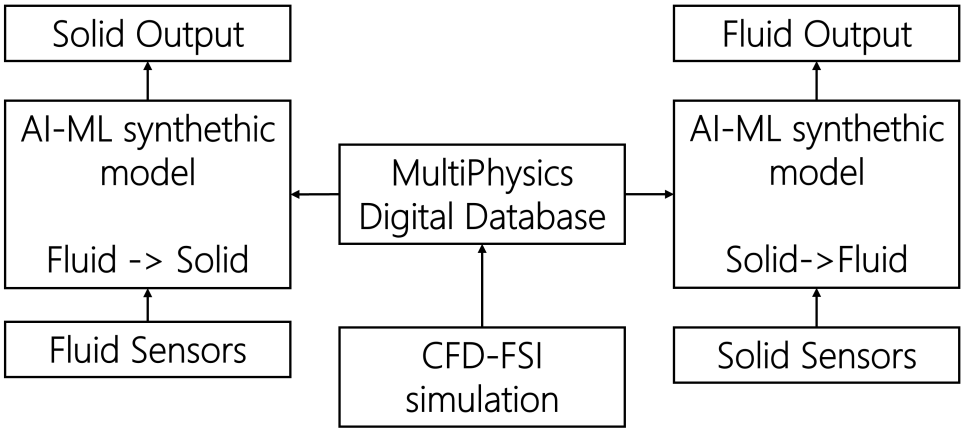

FIGURE 3.4: A possible application of a multiphysics database to reduce the dimensionality of a problem, or to bypass the solution of one field domain

problem, for example, to obtain as output a quantity of the solid domain from a fluid related input, or viceversa, as schematized in Fig. 3.4.

4. Mathematical modeling and methodologies in FSI simulations

The notation adopted in the description of the methods and algorithms in the following sections will follow the one proposed by Hughes [61] and Bazilevs, Tezduyar and Takizawa [33], where all the proposed models can be found for further studies. As a rule of thumb, scalar quantities will be written as lowercase, non bold letters, vectorial quantities will be written as lowercase, bold letters, while n-dimensional tensorial quantities will be written with uppercase, bold letters.

4.1 Finite Elements Method

4.1.1 Basics of FEM for boundary value problems

Finite Elements Methods (FEM) refers to a family of discretization techniques which are extremely useful and powerful to solve differential equations in boundary value problems (BVP) of the continuum, in a discretized, approximated domain. They are a well established technology for numerical simulations in structural mechanics, but they are widely used in fluid mechanics too, especially due to the fact that they can easily approximate differential equations for domains characterized by a particularly complex geometry. For a fixed spatial domain $\Omega \in \mathbb{R}^{n_{sd}}$, with boundary Γ (composed by the two complementary subsets Γ_g and Γ_h, respectively the essential and natural portion of the boundary), where are defined two function sets, S comprehending the trial functions, and V comprehending the test functions: each function member of S and V must possess the required continuity and differentiability level to represent the differential equations of the problem of interest. A generic, abstract, continuous, weak formulation of a BVP can be described as follows: find $u \in S$ such that $\forall w \in V$ stands the equation

$$B(w, u) - F(w) = 0 \tag{4.1}$$

with the condition that, in the trial and test function spaces are comprised the essential boundary conditions

$$u = g \ \ on \ \ \Gamma_g, \forall u \in S \tag{4.2}$$

$$w = 0 \ \ on \ \ \Gamma_g, \forall w \in V \tag{4.3}$$

where g is a prescribed function representing the Dirichlet (essential) condition of the BVP, B is a generic semi-linear form and F is a linear functional. When approximating a problem described as in Eq. (4.1) using FEM, the continuous domain Ω of the differential equation is approximated by a finite element domain Ω^h, composed of a set of sub-domains Ω^e (each bounded by a sub-boundary-domain Γ^b), such that

$$\Omega \approx \Omega^h = \bigcup_{e=i}^{n_{el}} \Omega^e \tag{4.4}$$

and

$$\Gamma \approx \Gamma^h = \bigcup_{e=i}^{n_{el}} \Gamma^b \tag{4.5}$$

Such discretization of the domain automatically defines the discrete counterparts of the sets S and V, respectively S^h and V^h, which are constructed by the locally supported finite element basis functions $N_A(x)$, where A is the index of the generic node of the mesh. In Petrov-Galerkin methods, both the trial and test functions are described by the same finite element basis functions, such that any finite element trial and test function can be represented as

$$u^h = v^h + g^h \ , \tag{4.6}$$

$$v^h = \sum_{A \in \boldsymbol{\eta}^s - \boldsymbol{\eta}^s_g} u_A N_A(\boldsymbol{x}) \ , \tag{4.7}$$

$$g^h = \sum_{A \in \boldsymbol{\eta}^s_g} g_A N_A(\boldsymbol{x}) \ , \tag{4.8}$$

$$w^h = \sum_{A \in \boldsymbol{\eta}^w} w_A N_A(\boldsymbol{x}) \ , \tag{4.9}$$

with the conditions

$$v^h = 0 \ \ on \ \ \Gamma^h_g \ , \tag{4.10}$$

$$g^h = \prod^h g \ \ on \ \ \Gamma^h_g \ , \tag{4.11}$$

$$w^h = 0 \ \ on \ \ \Gamma^h_g \ , \tag{4.12}$$

where the coefficients u_A are unknown and g_A are known, while $\boldsymbol{\eta}^s$, $\boldsymbol{\eta}^s_g$, $\boldsymbol{\eta}^w$ are respectively the subset of nodes of the discretized domain, the subset of nodes of the Dirichlet boundary, and the subset of nodes of the test functions. According to the proposed discretization, the continuous BVP of Eq. (4.1) is transformed in: find $u^h \in S^h$ such that $\forall w^h \in V^h$ stands the equation

$$B^h(w^h, u^h) - F^h(w^h) = 0 \tag{4.13}$$

that can be rewritten for every degree of freedom of every grid node as

$$B^h(N_A, u^h) - F^h(N_A) = 0 \;\; \forall A \in \boldsymbol{\eta}^w \tag{4.14}$$

Since B is in general a semilinear form, the solution of the matrix form of Eq. (4.14) must be found iteratively, tipically using a *Newton-Raphson* procedure from the initial guess $\mathbf{U}^0$ and for i iterations:

$$\boldsymbol{K}^i \Delta \mathbf{U}^i = -\boldsymbol{R}^i \tag{4.15}$$

with the update of the solution defined as

$$\boldsymbol{U}^{i+1} = \boldsymbol{U}^i + \mathbf{U}^i \tag{4.16}$$

The solution update is carried on until the right hand side of Eq. (4.15) is equal to zero.

4.1.2 Finite element basis function

The construction of the finite element basis functions that constitute the function sets S and V is a procedure that takes place at the individual element level, and only at a later stage, these are assembled together building up the interpolation of the entire grid. It is crucial to point out that while every element geometrical shape might differ in every aspect from one another, they can all be derived from some basic element shape in the parametric domain. For such reason a brief overview of the element shape function of linear hexahedral and tetrahedral elements in 3D will be given in this section, as these are the only implemented elements in the software FEMpar.

Element basis function: definition and properties

As stated before, the basis function $N_a(\boldsymbol{\xi})$, $a = 1, .., n_{en}$ for each of the element nodes n_{en} is defined in the parametric domain of the element $\hat{\Omega}^e$ with coordinates $\boldsymbol{\xi} \in \mathbb{R}^{n_{sd}}$. The parametric description of the element is allowed by the invertible and continuously differentiable mapping function between the parametric coordinates $\boldsymbol{\xi}$ and physical

coordinates $\boldsymbol{x}$

$$\boldsymbol{x}(\boldsymbol{\xi}) = \sum_{a=1}^{n_{en}} \boldsymbol{x}_a N_a(\boldsymbol{\xi}), \tag{4.17}$$

In addition to the requirements for the mapping function, it is mandatory that $N_a(\boldsymbol{\xi})$ satisfies the partition of unity property, and it's desirable that it satisfies the interpolation property. The solution funtion $u \in S$ is required to be written in the parametric domain as

$$u^h(\boldsymbol{\xi}) = \sum_{a=1}^{n_{en}} u_a N_a(\boldsymbol{\xi}), \tag{4.18}$$

and if the same basis function is used for both the interpolation of the geometry and the solution, the element is considered an isoparametric element [62]. In order to compute the derivatives of the solution in the paramentric element domain, the derivatives of the element shape function are required. Those are directly obtained by the Jacobian matrix of the element mapping

$$\frac{\partial \boldsymbol{x}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} = \sum_{a=1}^{n_{en}} \boldsymbol{x} \frac{\partial N_a(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \tag{4.19}$$

so it is possible to obtain the element shape function derivatives with respect to the physical coordinates through the inverse transposed Jacobian element (the invertibility of the mapping function allows the invertibility of the Jacobian)

$$\frac{\partial N_a(\boldsymbol{\xi})}{\partial \boldsymbol{x}} = (\frac{\partial \boldsymbol{x}}{\partial \boldsymbol{\xi}})^{-T} \frac{\partial N_a(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \tag{4.20}$$

which will be widely adopted in the evaluation of the left hand side and right hand side of the FEM equations for fluid dynamics and structural mechanics boundary value problems.

It is possible to build finite element shape functions N_a of order p by making use of the Lagrange polynomials: for a linear ($p = 1$) element $p+1$ element nodes are required in each parametric direction

$$l_i^p(\xi) = \prod_{j=1, j\neq i}^{p+1} \frac{(\xi - \xi_j)}{\xi_i - \xi_j}, \qquad for \;\; i = 1, 2..., p+1 \tag{4.21}$$

Isoparametric 3D hexahedron shape functions

Starting from a unit cube centered at the origin of the parametric coordinate system described by ξ, η, ζ, the 3D parametric shape functions for a trilinear hexahedron are

defined as

$$N_a(\boldsymbol{\xi}) = l_i^p(\xi) l_j^p(\eta) l_k^p(\zeta) \qquad for \;\; i,j,k = 1,2,...p+1 \tag{4.22}$$

resulting in the following shape functions and derivatives

$$N_1(\boldsymbol{\xi}) = \frac{(1-\xi)(1-\eta)(1-\zeta)}{8}, \qquad N_2(\boldsymbol{\xi}) = \frac{(1+\xi)(1-\eta)(1-\zeta)}{8},$$

$$N_3(\boldsymbol{\xi}) = \frac{(1+\xi)(1+\eta)(1-\zeta)}{8}, \qquad N_4(\boldsymbol{\xi}) = \frac{(1-\xi)(1+\eta)(1-\zeta)}{8},$$

$$N_5(\boldsymbol{\xi}) = \frac{(1-\xi)(1-\eta)(1+\zeta)}{8}, \qquad N_6(\boldsymbol{\xi}) = \frac{(1+\xi)(1-\eta)(1+\zeta)}{8},$$

$$N_7(\boldsymbol{\xi}) = \frac{(1+\xi)(1+\eta)(1+\zeta)}{8}, \qquad N_8(\boldsymbol{\xi}) = \frac{(1-\xi)(1+\eta)(1+\zeta)}{8},$$

$$\frac{\partial N_1(\boldsymbol{\xi})}{\partial(\xi)} = -\frac{(1-\eta)(1-\zeta)}{8}, \quad \frac{\partial N_1(\boldsymbol{\xi})}{\partial\eta)} = -\frac{(1-\xi)(1-\zeta)}{8}, \quad \frac{\partial N_1(\boldsymbol{\xi})}{\partial\zeta)} = -\frac{(1-\xi)(1-\eta)}{8},$$

$$\frac{\partial N_2(\boldsymbol{\xi})}{\partial(\xi)} = +\frac{(1-\eta)(1-\zeta)}{8}, \quad \frac{\partial N_2(\boldsymbol{\xi})}{\partial\eta)} = -\frac{(1+\xi)(1-\zeta)}{8}, \quad \frac{\partial N_2(\boldsymbol{\xi})}{\partial\zeta)} = -\frac{(1+\xi)(1-\eta)}{8},$$

$$\frac{\partial N_3(\boldsymbol{\xi})}{\partial(\xi)} = +\frac{(1+\eta)(1-\zeta)}{8}, \quad \frac{\partial N_3(\boldsymbol{\xi})}{\partial\eta)} = +\frac{(1+\xi)(1-\zeta)}{8}, \quad \frac{\partial N_3(\boldsymbol{\xi})}{\partial\zeta)} = -\frac{(1+\xi)(1+\eta)}{8},$$

$$\frac{\partial N_4(\boldsymbol{\xi})}{\partial(\xi)} = -\frac{(1+\eta)(1-\zeta)}{8}, \quad \frac{\partial N_4(\boldsymbol{\xi})}{\partial\eta)} = +\frac{(1-\xi)(1-\zeta)}{8}, \quad \frac{\partial N_4(\boldsymbol{\xi})}{\partial\zeta)} = -\frac{(1-\xi)(1+\eta)}{8},$$

$$\frac{\partial N_5(\boldsymbol{\xi})}{\partial(\xi)} = -\frac{(1-\eta)(1+\zeta)}{8}, \quad \frac{\partial N_5(\boldsymbol{\xi})}{\partial\eta)} = -\frac{(1-\xi)(1+\zeta)}{8}, \quad \frac{\partial N_5(\boldsymbol{\xi})}{\partial\zeta)} = +\frac{(1-\xi)(1-\eta)}{8},$$

$$\frac{\partial N_6(\boldsymbol{\xi})}{\partial(\xi)} = +\frac{(1-\eta)(1+\zeta)}{8}, \quad \frac{\partial N_6(\boldsymbol{\xi})}{\partial\eta)} = -\frac{(1+\xi)(1+\zeta)}{8}, \quad \frac{\partial N_6(\boldsymbol{\xi})}{\partial\zeta)} = +\frac{(1+\xi)(1-\eta)}{8},$$

$$\frac{\partial N_7(\boldsymbol{\xi})}{\partial(\xi)} = +\frac{(1+\eta)(1+\zeta)}{8}, \quad \frac{\partial N_7(\boldsymbol{\xi})}{\partial\eta)} = +\frac{(1+\xi)(1+\zeta)}{8}, \quad \frac{\partial N_7(\boldsymbol{\xi})}{\partial\zeta)} = +\frac{(1+\xi)(1+\eta)}{8},$$

$$\frac{\partial N_8(\boldsymbol{\xi})}{\partial(\xi)} = -\frac{(1+\eta)(1+\zeta)}{8}, \quad \frac{\partial N_8(\boldsymbol{\xi})}{\partial\eta)} = +\frac{(1-\xi)(1+\zeta)}{8}, \quad \frac{\partial N_8(\boldsymbol{\xi})}{\partial\zeta)} = +\frac{(1-\xi)(1+\eta)}{8}$$

Isoparametric 3D tetrahedron shape functions

The 3D parametric shape functions of a tetrahedron can be derived from a tetrahedron described with the natural coordinates (directly derived from the triangular coordinates [61]) system r, s, t, u, where r, s, t are the independent coordinates such that $r+s+t \leq 1$ and $0 \leq r \leq 1$, $0 \leq s \leq 1$, $0 \leq t \leq 1$ and u is a dependent coordinate such that

$u = 1 - r - s - t$, obtaining

$$N_a(\boldsymbol{r}) = T_i(r)T_j(s)T_k(t)T_l(u) \qquad for \;\; i,j,k,l = 1,2,...p+1 \tag{4.23}$$

where

$$T_i(r) = \begin{cases} l_i^{i-1}(2r/r_i - 1) & \text{if } i > 1 \\ 1 & \text{if } i = 1 \end{cases} \tag{4.24}$$

resulting in the following shape functions and derivatives

$$N_1(\boldsymbol{r}) = r, \quad N_2(\boldsymbol{r}) = s, \quad N_3(\boldsymbol{r}) = t, \quad N_4(\boldsymbol{r}) = u,$$

$$\frac{\partial N_1(\boldsymbol{r})}{\partial(r)} = 1, \quad \frac{\partial N_1(\boldsymbol{r})}{\partial(s)} = 0, \quad \frac{\partial N_1(\boldsymbol{t})}{\partial(t)} = 0,$$

$$\frac{\partial N_2(\boldsymbol{r})}{\partial(r)} = 0, \quad \frac{\partial N_2(\boldsymbol{r})}{\partial(s)} = 1, \quad \frac{\partial N_2(\boldsymbol{t})}{\partial(t)} = 0,$$

$$\frac{\partial N_3(\boldsymbol{r})}{\partial(r)} = 0, \quad \frac{\partial N_3(\boldsymbol{r})}{\partial(s)} = 0, \quad \frac{\partial N_3(\boldsymbol{t})}{\partial(t)} = 1,$$

$$\frac{\partial N_4(\boldsymbol{r})}{\partial(r)} = -1, \quad \frac{\partial N_4(\boldsymbol{r})}{\partial(s)} = -1, \quad \frac{\partial N_4(\boldsymbol{t})}{\partial(t)} = -1$$

4.2 ALE Methods

One of the most crucial decision to make in the numerical simulations of continuum mechanics is the correct choice of an adequate kinematic description of the continuum itself, especially when dealing with multidimensional problems of fluid dynamics, solid mechanics and fluid structure interaction. The discretization of the problem must be set up according to the problem which is intented to be solved, and not every kinematic description can describe the physics of each category of problem, in an appropriate and efficient way at the same time: the choice of discretization will highly influence the reliability of the numerical simulations when dealing with mesh distortions, or interfaces between solid and fluid [63]. Historically, two main kinematic descriptions have been adopted for the numerical simulations in continuum mechanics, in general, the Eulerian description and the Lagrangian description [64]:

- ***Lagrangian description***: a mesh node is bound to the associated material particle, the node tracks the motion of the material particle; it is the *de-facto* standard kinematic description for structural mechanics.
  - Pros: suitable for material with history-dependent constitutive relations, easy surface and interfaces tracking;

  - Cons: unable to handle high distortions of the continuum without remeshing;

- ***Eulerian description***: the mesh node is fixed while the continuum moves, the node is not bound to the the material particle; it is mainly used in fluid dynamics;

  - Pros: good at describing distortions in the continuum;
  - Cons: low precision interface definition, low resolution of the flow details;

When dealing with FSI problems, the attempt of adopting a hybrid kinematic description, in order to keep the advantages of the classical descriptions while lowering the impact of their disadvantages in the numerical simulations, is almost natural. In that sense, the ***Arbitrary Lagrangian-Eulerian*** (ALE) description can be seen as the perfect solution: in the ALE description a node of the grid can be moved as in the Lagrangian description, kept fixed in its place as in the Eulerian description, or moved in an independent direction to minimize the distortions in the mesh while still preserving the desired resolution in the discretization of complex surfaces or interfaces.

4.2.1 Finite Elements Method in ALE

In the next sections, a brief overview of the mathematical models implemented in the finite element software will be given, by adopting the ALE formulation for the equations of incompressible fluid dynamics, nonlinear structural dynamics and moving mesh algorithm. Before that, it appears useful to highlight the main differences between the standard Finite Element discretization for non moving domains against moving domains. The main difference is given by the fact that the motion of the discretized fluid domain (i.e. the mesh) does not follow the kinematic laws of the of the fluid domain that it represents, but it instead it has its own motion, displacements, velocity and acceleration. For this reason the definition of additional spatial reference coordinates $\hat{\boldsymbol{x}}$ is required, in order to describe and represent the domain kinematics. In particular, the fluid domain at a given time t Ω_t can be described by the reference coordinates $\hat{\boldsymbol{x}}$ of the reference domain $\hat{\boldsymbol{\Omega}}$ as

$$\Omega_t = \left\{ \boldsymbol{x} | \, \boldsymbol{x} = \boldsymbol{phi}(\hat{\boldsymbol{x}}, t) \forall \hat{\boldsymbol{x}} \in \hat{\boldsymbol{\Omega}}, t \in (0, T) \right\} \tag{4.25}$$

that leads to the mapping

$$\boldsymbol{phi}(\hat{\boldsymbol{x}}, t) = \hat{\boldsymbol{x}} + \hat{\boldsymbol{y}}(\hat{\boldsymbol{x}}, t) \, , \tag{4.26}$$

with $\hat{\boldsymbol{y}}$ displacement of the reference domain. Such displacement can be derived keeping the reference coordinates fixed to obtain the fluid domain velocity

$$\hat{\boldsymbol{u}} = \left.\frac{\partial \hat{\boldsymbol{y}}}{\partial t}\right|_{\hat{x}} \tag{4.27}$$

Such displacement velocity will be used in the ALE formulation of the Navier Stokes equations to take into account the motion of the fluid domain. The consistency between integration performed in the reference domain and the ones in the fluid domain at the current time (which will be widely used in the Navier-Stokes equation in the next section) is ensured by the Piola transformation:

$$\int_{\Omega_t} \boldsymbol{\nabla} \cdot \gamma d\Omega = \int_{\hat{\Omega}} \boldsymbol{\nabla}_{\hat{x}} \cdot \hat{\gamma} d\hat{\Omega} \tag{4.28}$$

in which γ is a generic vector quantity, while $\hat{\gamma}$ is the same vector quantity in the reference configuration, defined as

$$\hat{\gamma} = \hat{J}\hat{\boldsymbol{F}}^{-1}\gamma \tag{4.29}$$

with

$$\hat{\boldsymbol{F}} = \frac{\partial \boldsymbol{x}}{\partial \hat{\boldsymbol{x}}} = \boldsymbol{I} + \frac{\partial \hat{\boldsymbol{y}}}{\partial \hat{\boldsymbol{x}}} \tag{4.30}$$

the deformation gradient with respect to the reference configuration and

$$\hat{J} = det\hat{\boldsymbol{F}} \tag{4.31}$$

its Jacobian.

4.3 Variational Multiscale Method (RBVMS) in ALE

Variational MultiScale methods for the solution boundary value problems have been introduced in the late 90's by T.J.R. Hughes [65, 66], specifically for those phenomena which are existing and expressing their behaviour on multiple scales, and successfully applied to fluid dynamics in the solution of Navier-Stokes equations in [67, 68, 69, 70] as an alternative to the widely adopted and well known LES (Large Eddy Simulation) methods. The VMS method relies on the decomposition of the equations in two scales, large scales (or coarse scale, related to the element size of the grid) that are in general numerically solved, and fine scales (or sub-grid scales) that are modelled in consistency with the coarse scales. The approach is substantially different from the LES paradigm due to the fact that the equations are not filtered, but manipulated through a variational

projection on different scales. Early versions of the theory still adopted *ad-hoc*, traditional, eddy viscosity terms added at the equations at the fine scale level to model them. Bazilevs introduced a new concept in the variational multiscale theory, by presenting the RBVMS, i.e. Residual Based Variational Multiscale method [71]. The introduction of the RBVMS was intended as a step forward with respect to the traditional VMS approach, with the objective to solve specific issues of the former attempts, i.e. the use of traditional eddy viscosity to represent fine-scale dissipation, that was considered inefficient, and the ineffectiveness of energy transferring between scales when the discretization is very coarse. The main concepts behind this new variant of the model can be summarized as follows:

- The splitted (coarse and fine scales) exact solution is substituted in the Navier-Stokes equations and the equation is then projected into coarse-scale and fine-scale subspaces;
- The analytical expression of the fine scales is approximated through the adoption of an infinite perturbation series expansion (rapidly converging due to the smallness of the fine scales) and the linearized operator of the Navier-Stokes system, inverted via the Green's function [72];
- The fine scale modeling relies on the presence of the residuals of the coarse scale Navier-Stokes equations, in the sense that if the coarse solution is able to approximate the real solution, the coarse scale residual will be small and the resulting fine scale solution will be consequently small;

An exhaustive description of the model, as well as the demonstrations required to assert the quality and consistency of the model itself, are beyond the scope of the present work; nonetheless, a brief introduction to the final form of the RBVMS equations in the ALE reference frame, implemented in FEMpar, is still useful and is given below for the sake of completeness.

From the ALE form of the Navier-Stokes equations of incompressible flows

$$\rho \left(\left. \frac{\partial \boldsymbol{u}}{\partial t} \right|_{\hat{x}} + (\boldsymbol{u} - \hat{\boldsymbol{u}}) \cdot \boldsymbol{\nabla} \boldsymbol{U} - \boldsymbol{f} \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\boldsymbol{u}, p) = 0 \tag{4.32}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} \tag{4.33}$$

it is possible to obtain the variational (weak) formulation of the Navier-Stokes equations to find $\boldsymbol{u} \in S_u$ and $p \in S_p$ such that $\forall \boldsymbol{w} \in V_u$ and $q \in V_p$

$$\begin{aligned}\int_{\Omega_t} \boldsymbol{w} \cdot \rho \left(\left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{\hat{x}} + (\boldsymbol{u} - \hat{\boldsymbol{u}}) \cdot \boldsymbol{\nabla} \boldsymbol{U} - \boldsymbol{f} \right) d\Omega + \int_{\Omega_t} \varepsilon(\boldsymbol{w}) : \boldsymbol{\sigma}(\boldsymbol{u}, p) d\Omega \\ - \int_{(\Gamma_t)_h} \boldsymbol{w} \cdot \boldsymbol{h} d\Gamma + \int_{\Omega_t} q \boldsymbol{\nabla} \cdot \boldsymbol{u} d\Omega = 0\end{aligned} \tag{4.34}$$

the equation holds. From this equation it is possible to perform the multiscale decomposition to coarse scale and fine scale via a direct sum decomposition based on the weighting and solution function spaces

$$S_u = S_u^h \oplus S_{u'}' \tag{4.35}$$

$$S_p = S_p^h \oplus S_{p'}' \tag{4.36}$$

$$V_u = V_u^h \oplus S_{u'}' \tag{4.37}$$

$$V_p = V_p^h \oplus V_{p'}' \tag{4.38}$$

where the h superscript refers to the coarse, element-wise scale, while the prime superscript refers to the fine scale. A direct consequence of the decomposition of the function spaces is the decomposition of the function member of the same spaces, such that, also for the solution and weight functions holds

$$\boldsymbol{u} = \boldsymbol{u}^h + \boldsymbol{u}' \tag{4.39}$$

$$p = p^h + p' \tag{4.40}$$

$$\boldsymbol{w} = \boldsymbol{w}^h + \boldsymbol{w}' \tag{4.41}$$

$$q = q^h + q' \tag{4.42}$$

where $\boldsymbol{u}'$ and $\boldsymbol{p}'$, the fine scale counterpart of the solution in terms of velocity and pressure, are modelled as a function of the residual of the large scales and the stabilization coefficients

$$\boldsymbol{u}' = \frac{\tau_{SUPS}}{\rho} - \boldsymbol{r}_M(\boldsymbol{u}^h, p^h), \tag{4.43}$$

$$p' = -\rho \nu_{LSIC} r_C(\boldsymbol{u}^h), \tag{4.44}$$

where the residuals of the momentum equation $\boldsymbol{r}_M$ and of the continuity equation r_C are defined, with respect to the coarse scales, as

$$\boldsymbol{r}_M(\boldsymbol{u}^h, p^h) = \rho \left(\left. \frac{\partial \boldsymbol{u}^h}{\partial t} \right|_{\hat{x}} + (\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h) - \boldsymbol{f}^h \right) - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\boldsymbol{u}^h, p^h), \tag{4.45}$$

$$r_C(\boldsymbol{u}^h) = \boldsymbol{\nabla} \cdot \boldsymbol{u}^h, \tag{4.46}$$

In the above equations, the parameters τ_{SUPS} and ν_{LSIC} are intended as stabilization parameters, specifically the former as the Streamwise Upwind Pressure Stabilizing parameter and the latter as Least Squared Incompressibility Constraint parameter. More informations about them and how they were developed for finite element theory can be found in [73, 74, 75, 76, 77, 78]. In the FEMpar implementation, they are evaluated as

$$\tau_{SUPS} = \left(\frac{4}{\Delta t^2} + \boldsymbol{u}^h \cdot \boldsymbol{G} \boldsymbol{u}^h + C_I \nu^2 \boldsymbol{G} : \boldsymbol{G} \right)^{-1/2} \tag{4.47}$$

$$\nu_{LSIC} = (tr\boldsymbol{G} \tau_{SUPS})^{-1} \tag{4.48}$$

where $\boldsymbol{G}$ is the metric tensor, $tr\boldsymbol{G}$ its trace and C_I is a constant that depends on the element topology [79]. According to [71], it is possible to substitute the decomposition above directly into Eq. (4.34), obtaining the semi-discrete RBVMS formulation of incompressible flows:

$$\begin{aligned}
&\int_{\Omega_t} \boldsymbol{w}^h \cdot \rho \left(\frac{\partial \boldsymbol{u}^h}{\partial t}|_{\hat{x}} + (\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h) \cdot \boldsymbol{\nabla} \boldsymbol{U}^h - \boldsymbol{f}^h \right) d\Omega + \int_{\Omega_t} \boldsymbol{\varepsilon}(\boldsymbol{w}^h) : \boldsymbol{\sigma}(\boldsymbol{u}^h, p^h) d\Omega \\
&- \int_{(\Gamma_t)_h} \boldsymbol{w}^h \cdot \boldsymbol{h}^h d\Gamma + \int_{\Omega_t} q^h \boldsymbol{\nabla} \cdot \boldsymbol{u}^h d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{\Omega_t^e} \tau_{SUPS} \left((\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h) \cdot \boldsymbol{\nabla} \boldsymbol{w}^h + \frac{\boldsymbol{\nabla} q^h}{\rho} \right) \cdot \boldsymbol{r}_M(\boldsymbol{u}^h, p^h) d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{\Omega_t^e} \rho \nu_{LSIC} \boldsymbol{\nabla} \cdot \boldsymbol{w}^h r_C(\boldsymbol{u}^h) d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{\Omega_t^e} \tau_{SUPS} \boldsymbol{w}^h \cdot (\boldsymbol{r}_M(\boldsymbol{u}^h, p^h) \cdot \boldsymbol{\nabla} \boldsymbol{u}^u) d\Omega \\
&- \sum_{e=1}^{n_{el}} \int_{\Omega_t^e} \frac{\boldsymbol{\nabla} \boldsymbol{w}^h}{\rho} : (\tau_{SUPS} \boldsymbol{r}_M(\boldsymbol{u}^h, p^h)) \otimes (\tau_{SUPS} \boldsymbol{r}_M(\boldsymbol{u}^h, p^h)) d\Omega = 0
\end{aligned} \tag{4.49}$$

4.4 Weakly enforced essential boundary conditions

In order to improve the accuracy of simulations where the mesh size was kept coarse at the boundary layers (i.e. in presence of unresolved boundary layers), Bazilevs introduced in the RBVMS formulation the weakly enforced essential boundary conditions [80]: in this formulations, essential boundary conditions are removed from the trial and test function spaces, and a series of surface integrals at the boundaries are added on the left hand side of the equation (in practice, they're no longer known variables but they're resolved during the inversion of the matrix as any other variable in the fluid field):

$$
\begin{aligned}
&-\sum_{b=1}^{n_e b}\int_{\Gamma^b\cap(\Gamma_t)_g} \boldsymbol{w}^h\cdot\boldsymbol{\sigma}(\boldsymbol{u}^h,p^h)\boldsymbol{n}d\Gamma \\
&-\sum_{b=1}^{n_e b}\int_{\Gamma^b\cap(\Gamma_t)_g} (2\mu\varepsilon(\boldsymbol{w}^h\boldsymbol{n}+q^h\boldsymbol{n})\cdot(\boldsymbol{u}^h-\boldsymbol{g}^h)d\Gamma \\
&-\sum_{b=1}^{n_e b}\int_{\Gamma^b\cap(\Gamma_t)_g^-} \boldsymbol{w}^h\cdot\rho((\boldsymbol{u}^h-\hat{\boldsymbol{u}}^h)\cdot\boldsymbol{n})(\boldsymbol{u}^h-\boldsymbol{g}^h)d\Gamma \\
&+\sum_{b=1}^{n_e b}\int_{\Gamma^b\cap(\Gamma_t)_g} \tau^B_{TAN}(\boldsymbol{w}^h-(\boldsymbol{w}^h\cdot\boldsymbol{n})\boldsymbol{n})\cdot((\boldsymbol{u}^h-\boldsymbol{g}^h)-((\boldsymbol{u}^h-\boldsymbol{g}^h)\cdot\boldsymbol{n})\boldsymbol{n})d\Gamma \\
&+\sum_{b=1}^{n_e b}\int_{\Gamma^b\cap(\Gamma_t)_g} \tau^B_{NOR}(\boldsymbol{w}^h\cdot\boldsymbol{n})((\boldsymbol{u}^h-\boldsymbol{g}^h)\cdot\boldsymbol{n})d\Gamma
\end{aligned}
\tag{4.50}
$$

where $\tau^B_{TAN}=\tau^B_{NOR}$ are stabilization parameters set equal to $\frac{{C_I}^B\mu}{h_n}$, with ${C_I}^B$ constant and h_n equal to the size of the element in the direction normal to the wall. With this numerical approach, the solution can converge to the same solution obtained using classical Dirichlet conditions if $h_n \longrightarrow 0$ [80, 33].

4.5 Non-Linear Elasticity using Total Lagrangian formulation

The tradition of solving discretely the linear elasticity problem of the continuum using finite elements is well established, and from the foundations of this widely adopted methodology grew the research effort that eventually led to the development of finite elements methods for fluid dynamics and fluid structure interaction. A less widespread

model is the non linear elasticity model, also known as Total Lagrangian Formulation, where the displacements are evaluated in an incremental fashion, from a reference configuration up to the current configuration. The TL formulation considers in the stiffness matrix a series of geometrical non linearities (i.e. unrelated to the model adopted to describe the material, but only relative to the state of the geometrical deformation, through the displacements), and which is particularly suitable for slender and thin structures, where large (purely geometrical) deformations coexist with low stress distributions [81, 33].

In this section, a brief introduction to the equations implemented in the solid mechanics solver of FEMpar will be given, accordingly to the notation already adopted in Sec. (4.3) and based on the works of Hughes, Bazilevs, Tezduyar and Takizawa [61, 33]. For further information about the formulation, and for a more detailed mathematical description, it is useful to refer to [61, 33].

More precisely, the reference configuration (i.e. the initial configuration of the solid at time $t = 0$) can be described as $\Omega_0 \in \mathbb{R}^n_{sd}$ bounded by Γ_0 and its coordinates will be referred as $\boldsymbol{X}$, while for the current configuration $\Omega_t \in \mathbb{R}^n_{sd}$ bounded by Γ_t at time $t > 0$, and its coordinates, the notation $\boldsymbol{x}$ will be adopted. In general, lowercase indices will be used for quantities referring to the reference configuration, while uppercase indices will refer to quantities in the current configurations. The displacement vector between the two configuration is $\boldsymbol{y}$, and thus a map between coordinates can be defined:

$$\boldsymbol{x}(\boldsymbol{X}, t) = \boldsymbol{X} + \boldsymbol{y}(\boldsymbol{X}, t) \tag{4.51}$$

From the principle of virtual work, which states that for a rigid body the sum of the internal and external work is zero

$$\delta W = \delta W_{int} + \delta W_{ext} = 0 \tag{4.52}$$

where δ is the variation with respect to the virtual displacement $\boldsymbol{w}$, it is possible to obtain a variational formulation for the structural mechanics. In order to obtain such formulation, it is necessary to introduce a series of more complex quantities, such as the Green-Lagrange strain tensor $\boldsymbol{E}$ and the second Piola-Kirchhoff stress tensor $\boldsymbol{S}$, obtained by manipulating the deformation gradient $\boldsymbol{F}$ and the Cauchy-Green deformation tensor $\boldsymbol{C}$:

$$\boldsymbol{F} = \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{X}} = \boldsymbol{I} + \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{X}} , \qquad \boldsymbol{C} = \boldsymbol{F}^T \boldsymbol{F}$$
$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}) , \qquad \boldsymbol{S} = \mathbb{C}\boldsymbol{E}$$

in which $\boldsymbol{I}$ is the identity tensor, and $\mathbb{C}$ is the constitutive relation of the material. The above definition of S holds for the constitutive model of St. Venant-Kirchhoff, where

$$\mathbb{C}_{IJKL} = \left(\kappa - \frac{2}{2}\mu\right)\delta_{IJ}\delta KL + \mu(\delta_{IK}\delta_{JL} + \delta_{IL}\delta_{JK}), \tag{4.53}$$

with κ, λ and μ defined as functions of Young's modulus E and Poisson's ratio ν:

$$\begin{aligned}\kappa &= \lambda + \frac{2}{3}\mu \\ \mu &= \frac{E}{2(1+\nu)} \\ \lambda &= \frac{\nu E}{(1+\nu)(1-2\nu)}\,.\end{aligned}$$

With the aid of such defined quantities it is possible to express the variational formulation of the principle of virtual work, where the objective is to find $\boldsymbol{y} \in S_y$ such that $\forall \boldsymbol{w} \in V_y$ holds the equation:

$$\int_{\Omega_t} \boldsymbol{w} \cdot \rho \boldsymbol{a} d\Omega + \int_{\Omega_0} \delta \boldsymbol{E} : \boldsymbol{S} d\Omega - \int_{\Omega_t} \boldsymbol{w} \cdot \rho \boldsymbol{f} d\Omega - \int_{(\Gamma_t)_h} \boldsymbol{w} \cdot \boldsymbol{h} d\Gamma = 0 \tag{4.54}$$

In the Eq. (4.54) $\boldsymbol{y}$ and $\boldsymbol{w}$ are respectively the trial functions and the test functions, members of the sets S_y and V_y, defined in the complementary subsets of the boundary $(\Gamma_t)_{gi}$ (where $y_i = g_i$, with g_i a specified function) and $(\Gamma_t)_{hi}$ (where $w_i = 0$). Unfortunately the above formulations is obtained by considering stresses from different configurations in time, and such strategy is not optimal. It is possible to write an alternative form of the same equation which adopts stresses entirely referred to a single configuration, alternatively the reference or the current configuration. In the present work, as stated before, the Total Lagrangian formulation is preferred for the description of structural mechanics, and consequently the following equation is implemented in FEMpar, obtained by manipulating Eq. (4.54):

$$\int_{\Omega_0} \boldsymbol{w} \cdot (\rho_0(\boldsymbol{a} - \boldsymbol{f}) - \boldsymbol{\nabla}_X \boldsymbol{w} : \boldsymbol{P}) d\Omega + \int_{(\Gamma_0)_h} \boldsymbol{w} \cdot (\boldsymbol{P}\hat{\boldsymbol{n}} - \hat{\boldsymbol{h}}) d\Gamma = 0 \tag{4.55}$$

In Eq. (4.55) there is the introduction of a couple of quantities which are convenient when adopting the reference configuration as the main configuration, such as the first Piola-Kirchhoff tensor $\boldsymbol{P} = \boldsymbol{F}\boldsymbol{S}$, the gradient evaluated with respect to the reference configuration $\boldsymbol{\nabla}_X \boldsymbol{w}$, the traction vector in the reference configuration $\hat{h}$ and the normal unit vector in the reference configuration $\hat{n}$.

To solve for the non linearities in the equation, it is possible to perform a linearization

of the expression of the virtual work under the hypothesis of small perturbation $\boldsymbol{y}$ of the displacement around the deformed state $\bar{\boldsymbol{y}}$, where the bar is denoting that the quantity is evaluated at the deformed state:

$$\delta W(\boldsymbol{w}, \bar{\boldsymbol{y}}) + \frac{d}{d\epsilon}\delta W\left(\boldsymbol{w}, \bar{\boldsymbol{y}} + \epsilon \boldsymbol{y}\right)|_{\epsilon=0} = 0 \tag{4.56}$$

From the linearization of the internal virtual work (see [33]) it is possible to obtain a new description of the internal work terms as

$$\frac{d}{d\epsilon}\delta W_{int}\left(\boldsymbol{w}, \bar{\boldsymbol{y}} + \epsilon \boldsymbol{y}\right)|_{\epsilon=0} = \int_{\Omega_0} \left(\bar{\boldsymbol{F}}^t \boldsymbol{\nabla}_X \boldsymbol{w} : \bar{\mathbb{C}}\bar{\boldsymbol{F}}^t \boldsymbol{\nabla}_X \boldsymbol{y} + \boldsymbol{\nabla}_X \boldsymbol{w} : \boldsymbol{\nabla}_X \boldsymbol{y}\bar{\boldsymbol{S}}\right) d\Omega \tag{4.57}$$

from which emerges the definition of the tangent stiffness tensor composed of a sum of two elements, the first one referred as the material stiffness, and the second one as the geometric stiffness. More concisely, in index notation, Eq. (4.57) can be expressed as

$$\int_{\Omega_0} w_{i,J}\bar{D}_{iJkL}y_{k,L}d\Omega \tag{4.58}$$

where appear the components of the tangent stiffness tensor $\bar{D}_{iJkL}$:

$$\bar{D}_{iJkL} = \bar{F}_{iI}\bar{\mathbb{C}}_{IJKL}\bar{F}_{kK} + \delta ik\bar{S}_{JL} \tag{4.59}$$

Following the linearization of all the terms of Eq. (4.55), the linearized problem (solved inside a Newton-Raphson algorithm) can be stated as: from the structural displacement $\bar{\boldsymbol{y}}$ find $\boldsymbol{y} \in S_y$ such that $\forall \boldsymbol{w} \in V_y$

$$\begin{aligned}
&\int_{\Omega_0} \boldsymbol{w} \cdot \rho_0 \bar{\boldsymbol{a}} d\Omega + \int_{\Omega_0} \boldsymbol{\nabla}_X \boldsymbol{w} : \bar{P} d\Omega - \int_{\Omega_0} \boldsymbol{w} \cdot \rho_0 \bar{\boldsymbol{f}} d\Omega - \int_{(\Gamma_0)_h} \boldsymbol{w} \cdot \bar{\boldsymbol{h}} \\
&+ \int_{\Omega_0} \boldsymbol{w} \cdot \rho_0 \boldsymbol{a} d\Omega + \int_{\Omega_0} \left(\bar{\boldsymbol{F}}^t \boldsymbol{\nabla}_X \boldsymbol{w} : \bar{\mathbb{C}}\bar{\boldsymbol{F}}^t \boldsymbol{\nabla}_X \boldsymbol{y} + \boldsymbol{\nabla}_X \boldsymbol{w} : \boldsymbol{\nabla}_X \boldsymbol{y}\bar{\boldsymbol{S}}\right) d\Omega \qquad (4.60) \\
&- \int_{\Omega_0} \boldsymbol{w} \cdot \rho_0 \boldsymbol{f} d\Omega - \int_{(\Gamma_0)_h} \boldsymbol{w} \cdot \hat{\boldsymbol{h}}
\end{aligned}$$

(4.61)

4.6 Solid-Extension Mesh Moving Technique (SEMMT)

When dealing with fluid structure interaction simulations of complex geometries, where the solid is characterized by a complex interface, which is moving according

to the ALE framework, it is mandatory to update the mesh in order to accomodate the displacements of the solid, especially at the fluid-solid interface. To achieve this, the main strategies that can be adopted are moving the fluid mesh (effectively deforming the elements of the mesh), remeshing (by generating new elements, nodes, and connectivity structures on the entire mesh, or on a portion), or a combination of both.

In general, the remeshing strategy is able to ensure the generation of a new mesh with high element quality, through the adoption of automated meshing algorithm that can be called whenever needed. However, a non negligible increase in the computational effort and in the computational time must be considered, due to the fact that automated mesh generation algorithm can be quite costly, especially if in addition to the new connectivity, new mesh node or elements have to be generated, introducing searching and interpolation operations, to project the solution from the old mesh to the new one. For this reason it is common to reduce at the minimum the frequency of remeshing, or to entirely avoid such operation.

A cost-efficient, high effective mesh-moving strategy for matching interfaces, that is able to drastically reduce the frequency of remeshing, up to making (in some cases) a remeshing algorithm not needed at all, is the SEMMT (Solid-Extension Mesh Moving Technique), developed by Jhonson and Tezduyar [82, 83, 84], an automated moving mesh algorithm where the elements of the fluid mesh are treated as an extension of the solid elements: the fluid elements are characterized by a Jacobian based stiffness matrix, and the displacements of their nodes are evaluated by solving a simple linear elastic problem: find the mesh node displacement $\hat{\boldsymbol{y}} \in S_m$ such that $\forall \boldsymbol{w} \in V_m$

$$\int_{\Omega_{\tilde{t}}} \boldsymbol{\epsilon}(\boldsymbol{w}) \cdot \boldsymbol{D^h} \boldsymbol{\epsilon}(\hat{\boldsymbol{y}}(t) - \hat{\boldsymbol{y}}(\tilde{t})) d\Omega = 0 \tag{4.62}$$

where $\Omega_{\tilde{t}}$ is the fluid domain at a given time $\tilde{t} < t$, $\hat{\boldsymbol{y}}$ denotes the fluid mesh displacements, $\boldsymbol{D}^h$ is the isotropic material elastic tensor defined with artificial Lamè parameters specific for the mesh motion

$$\begin{aligned} \mu^h &= \frac{E_m^h}{2(1+\nu_m)} \\ \lambda^h &= \frac{\nu_m E_m^h}{(1+\nu_m)(1-2\nu_m)} \\ E_m^h &= E_m \left(\frac{J_{x\xi}}{J_{x\xi}^0} \right)^{-\chi} \end{aligned}$$

where E_m and ν_m are the user defined mesh's Young's modulus and Poisson's ratio,$\chi > 0$ is a real parameter, $J_{x\xi}$ is the Jacobian determinant of the parametric mapping for the

mesh element and J^0 a scaling constant. Typical values are $\chi = 1.0$,$\nu_m = 0.3$ and $E_m =$, even thought the last one does not have any impact on the discrete solution (unless adopting a direct coupling solution scheme). Eq. (4.62) can be solved by imposing that at the interface the normal displacements of the solid mesh is equal to the normal displacement of the fluid mesh, and that the test function is equal to zero in the normal direction of the moving part of the fluid interface Γ_t:

$$\begin{aligned} \hat{\boldsymbol{y}} \cdot \boldsymbol{n} &= \boldsymbol{y} \cdot \boldsymbol{n} \quad on\ (\Gamma_t)_g \\ \boldsymbol{w} \cdot \boldsymbol{n} &= 0 \quad on\ (\Gamma_t)_g \end{aligned} \tag{4.63}$$

In the SEMMT scheme, a higher stiffness is assigned to lower volume elements automatically via the definition of the mesh Young's modulus E_m^h using the determinant of the Jacobian. Most likely, smaller elements will be placed near the interface, and in general next to the solid surfaces, in order to appropriately describe the fluid behaviour close to the boundary layer. Such elements will not be distorted by the moving mesh algorithm, and the mesh quality imposed by the user at the mesh design stage will be retained entirely, while allowing, at the same time, large deformations of the fluid mesh cells in the far field, where elements are larger and consequently less stiff.

4.7 ALE formulation for FSI

The fluid structure interaction formulation can be assembled directly from the previous descriptions of fluid dynamics, structural mechanics and moving mesh algorithm. The resulting equation can be solved using several solution and coupling strategies, but the starting point is the same for every declination of the fluid structure interaction problem.

The solution domain of the FSI problem is the union of the non overlapping fluid and solid domains

$$\Omega_i = (\Omega_1)_i \cup (\Omega_2)_i \qquad\qquad (\Omega_1)_i \cap (\Omega_2)_i = 0$$

where the subscript $i = (0, t)$ refers to a generic configuration in time of the moving domain, while the subscript 1 refers to the fluid domain and the 2 to the solid domain. In the following equations, the subscript I will refer to the interface domain and interface-specific quantities, while the E subscript will refer to the "elsewhere" keyword, and consequently will define a portion or boundary of the domains by excluding the related interface.

The ALE-VMS formulation, with TL and SEMMT with matching interfaces, of the FSI problem can be obtained by adding the equations developed in the previous sections, specifically Eq.s (4.50),(4.55) and (4.62), into a single equation

$$
\begin{aligned}
&\int_{\Omega_t} \boldsymbol{w}_1^h \cdot \rho \left(\frac{\partial \boldsymbol{u}^h}{\partial t}|_{\hat{x}} + (\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h) \cdot \boldsymbol{\nabla} \boldsymbol{u}^h - \boldsymbol{f}^h \right) d\Omega + \int_{\Omega_t} \varepsilon(\boldsymbol{w}_1^h) : \boldsymbol{\sigma}_1^h d\Omega \\
&- \int_{(\Gamma_t)_h} \boldsymbol{w}_1^h \cdot \boldsymbol{h}_{1E}^h d\Gamma + \int_{\Omega_t} q_1^h \boldsymbol{\nabla} \cdot \boldsymbol{u}^h d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{(\Omega_1)_t^e} \tau_{SUPS} \left((\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h) \cdot \boldsymbol{\nabla} \boldsymbol{w}_1^h + \frac{\boldsymbol{\nabla} q_1^h}{\rho} \right) \cdot \boldsymbol{r}_M(\boldsymbol{u}^h, p^h) d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{(\Omega_1)_t^e} \rho \nu_{LSIC} \boldsymbol{\nabla} \cdot \boldsymbol{w}_1^h r_C(\boldsymbol{u}^h) d\Omega \\
&- \sum_{e=1}^{n_{el}} \int_{(\Omega_1)_t^e} \tau_{SUPS} \boldsymbol{w}_1^h \cdot (\boldsymbol{r}_M(\boldsymbol{u}^h, p^h) \cdot \boldsymbol{\nabla} \boldsymbol{u}^u) d\Omega \\
&- \sum_{e=1}^{n_{el}} \int_{(\Omega_1)_t^e} \frac{\boldsymbol{\nabla} \boldsymbol{w}_1^h}{\rho} : (\tau_{SUPS} \boldsymbol{r}_M(\boldsymbol{u}^h, p^h)) \otimes (\tau_{SUPS} \boldsymbol{r}_M(\boldsymbol{u}^h, p^h)) d\Omega \\
&+ \int_{(\Omega_2)_0} \boldsymbol{w}_2^h \cdot (\rho_2)_0 \left(\frac{\boldsymbol{d^2 y^h}}{dt^2} - \boldsymbol{f}_2^h \right) + \int_{(\Omega_2)_0} \boldsymbol{\nabla}_X \boldsymbol{w}_2^h : \boldsymbol{P}^h d\Omega \\
&- \int_{(\Gamma_2)_h} \boldsymbol{w}_2^h \cdot \boldsymbol{h}_{2E}^h d\Gamma \\
&+ \int_{(\Omega_1)_{\tilde{t}}} \boldsymbol{\epsilon}(\boldsymbol{w}_3^h) \cdot \boldsymbol{D^h} \boldsymbol{\epsilon} \left(\hat{\boldsymbol{y}}(t) - \hat{\boldsymbol{y}}(\tilde{t}) \right) d\Omega = 0
\end{aligned}
\tag{4.64}
$$

The Eq. (4.64) requires additional constraints to be solved, specifically the traction continuity and the kinematic compatibility of the velocities of the fluid and solid are required at the fluid-structure interface

$$\boldsymbol{\sigma}_1 \boldsymbol{n}_1 + \boldsymbol{\sigma}_2 \boldsymbol{n}_2 = 0 \quad on \Gamma_I \tag{4.65}$$

$$\boldsymbol{u}^h = \frac{d \boldsymbol{y}^h}{dt} \quad on \Gamma_I \tag{4.66}$$

as well as the identity of the test functions at the interface,

$$\boldsymbol{w}_1^h = \boldsymbol{w}_2^h \quad on \Gamma_I \tag{4.67}$$

and the kinematic compatibility between the mesh displacement and the solid displacement

$$\hat{\boldsymbol{y}}^h = \boldsymbol{y}^h \quad on\Gamma_I \tag{4.68}$$

$$\boldsymbol{w}_3^h = 0 \quad on\Gamma_I \tag{4.69}$$

The variational formulation expressed by Eq. (4.64) can be expressed in the vector form by substituting the trial and test functions of the fluid mechanics, structural, and moving mesh problems with their discrete counterpart, expressed as

$$\boldsymbol{y}^h(\boldsymbol{X},t) = \sum_{\eta^s_{struct}} \boldsymbol{y}_A(t)N_A(\boldsymbol{X}), \qquad \hat{\boldsymbol{y}}^h(\hat{\boldsymbol{x}},t) = \sum_{\eta^s_{mesh}} \hat{\boldsymbol{y}}_A(t)N_A(hat\boldsymbol{x}),$$

$$\boldsymbol{w}_2^h(\boldsymbol{X},t) = \sum_{\eta^s_{struct}} (\boldsymbol{w}_2)_A(t)N_A(\boldsymbol{X}), \qquad \boldsymbol{w}_3^h(\hat{\boldsymbol{x}},t) = \sum_{\eta^s_{mesh}} (\boldsymbol{w}_3)_A(t)N_A(\hat{\boldsymbol{x}}),$$

with η specific nodal index set for each equation, in every member of the full problem equation, obtaining the residual formulation in the vector form

$$\boldsymbol{N}_{1M} = [(N_{1M})_{A,i}] = \boldsymbol{N}_{1M}(\dot{\boldsymbol{U}},\boldsymbol{U},P,\ddot{\boldsymbol{Y}},\dot{\boldsymbol{Y}},\boldsymbol{Y},\ddot{\hat{\boldsymbol{Y}}},\dot{\hat{\boldsymbol{Y}}},\hat{\boldsymbol{Y}}) = 0, \tag{4.70}$$

$$\boldsymbol{N}_{1C} = [(N_{1C})_A] = \boldsymbol{N}_{1C}(\dot{\boldsymbol{U}},\boldsymbol{U},P,\ddot{\boldsymbol{Y}},\dot{\boldsymbol{Y}},\boldsymbol{Y},\ddot{\hat{\boldsymbol{Y}}},\dot{\hat{\boldsymbol{Y}}},\hat{\boldsymbol{Y}}) = 0, \tag{4.71}$$

$$\boldsymbol{N}_2 = [(N_2)_{A,i}] = \boldsymbol{N}_2(\dot{\boldsymbol{U}},\boldsymbol{U},P,\ddot{\boldsymbol{Y}},\dot{\boldsymbol{Y}},\boldsymbol{Y},\ddot{\hat{\boldsymbol{Y}}},\dot{\hat{\boldsymbol{Y}}},\hat{\boldsymbol{Y}}) = 0, \tag{4.72}$$

$$\boldsymbol{N}_3 = [(N_3)_{A,i}] = \boldsymbol{N}_3(\dot{\boldsymbol{U}},\boldsymbol{U},P,\ddot{\boldsymbol{Y}},\dot{\boldsymbol{Y}},\boldsymbol{Y},\ddot{\hat{\boldsymbol{Y}}},\dot{\hat{\boldsymbol{Y}}},\hat{\boldsymbol{Y}}) = 0, \tag{4.73}$$

where $(N_{1M})_{A,i}$ is the Navier-Stokes momentum equation evaluated at each fluid node A in all the spatial dimensions $i = 1,...,n_{sd}$, $(N_{1C})_A$ is the Navier-Stokes continuity equation, $(N_2)_{A,i}$ is the structural mechanics equation and $(N_3)_{A,i}$ is the moving mesh equation, and where $\boldsymbol{U}$ represents the solution vector of the velocity of the fluid and its time derivatives, $\boldsymbol{P}$ the fluid pressure, $\boldsymbol{Y}$ the displacements of the structure and its time derivatives and $\boldsymbol{Y}$ the displacement of the mesh and its time derivatives. Specifically,

Eq.s (4.70 - 4.73) are rewritten as

$$
\begin{aligned}
(N_{1M})_{A,i} &= \int_{(\Omega_1)_t} N_A \boldsymbol{e}_i \cdot \rho \left(\frac{\partial \boldsymbol{u}^h}{\partial t}|_{\hat{x}} + (\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h) \cdot \boldsymbol{\nabla} \boldsymbol{u}^h - \boldsymbol{f}^h \right) d\Omega \\
&+ \int_{\Omega_t} \varepsilon N_A \boldsymbol{e}_i : \boldsymbol{\sigma}_1^h d\Omega - \int_{(\Gamma_t)_h} N_A \boldsymbol{e}_i \cdot \boldsymbol{h}_{1E}^h d\Gamma \\
&+ \sum_{e=1}^{n_{el}} \int_{(\Omega_1)_t^e} \tau_{SUPS} \left((\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h) \cdot \boldsymbol{\nabla} N_A \boldsymbol{e}_i + \frac{\boldsymbol{\nabla} q_1^h}{\rho} \right) \cdot \boldsymbol{r}_M(\boldsymbol{u}^h, p^h) d\Omega \\
&+ \sum_{e=1}^{n_{el}} \int_{(\Omega_1)_t^e} \rho \nu_{LSIC} \boldsymbol{\nabla} \cdot N_A \boldsymbol{e}_i r_C(\boldsymbol{u}^h) d\Omega \\
&- \sum_{e=1}^{n_{el}} \int_{(\Omega_1)_t^e} \tau_{SUPS} N_A \boldsymbol{e}_i \cdot (\boldsymbol{r}_M(\boldsymbol{u}^h, p^h) \cdot \boldsymbol{\nabla} \boldsymbol{u}^u) d\Omega \\
&- \sum_{e=1}^{n_{el}} \int_{(\Omega_1)_t^e} \frac{\boldsymbol{\nabla} N_A \boldsymbol{e}_i}{\rho} : (\tau_{SUPS} \boldsymbol{r}_M(\boldsymbol{u}^h, p^h)) \otimes (\tau_{SUPS} \boldsymbol{r}_M(\boldsymbol{u}^h, p^h)) d\Omega = 0 \,, \qquad (4.74)
\end{aligned}
$$

$$
(N_{1C})_A = \int_{(\Omega_1)_t} N_A \boldsymbol{\nabla} \cdot \boldsymbol{u}^h d\Omega + - \sum_{e=1}^{n_{el}} \int_{(\Omega_1)_t^e} \frac{\boldsymbol{\nabla} N_A}{\rho} \cdot \boldsymbol{r}_M(\boldsymbol{u}^h, p^h) d\Omega = 0 \,, \qquad (4.75)
$$

$$
\begin{aligned}
(N_2)_{A,i} &= \int_{(\Omega_2)_0} N_A \boldsymbol{e}_i \cdot (\rho_2)_0 \left(\frac{\boldsymbol{d^2 y^h}}{dt^2} - \boldsymbol{f}_2^h \right) + \int_{(\Omega_2)_0} \boldsymbol{\nabla} N_A \boldsymbol{e}_i : \boldsymbol{P}^h d\Omega \\
&- \int_{(\Gamma_2)_h} N_A \boldsymbol{e}_i \cdot \boldsymbol{h}_{2E}^h d\Gamma = 0 \,, \qquad (4.76)
\end{aligned}
$$

$$
(N_3)_{A,i} = \int_{(\Omega_1)_{\tilde{t}}} \boldsymbol{\epsilon}(N_A \boldsymbol{e}_i) \cdot \boldsymbol{D^h} \boldsymbol{\epsilon} \left(\hat{\boldsymbol{y}}(t) - \hat{\boldsymbol{y}}(\tilde{t}) \right) d\Omega = 0 \qquad (4.77)
$$

4.8 Generalized alpha time integration

The differential equations which are functions of time, not only require a spatial discretization to be solved, but also the discretization of the time evolution. Unsteady simulations rely on an appropriate time discretizazion, and consequently on a time integration algorithm, where the time interval $[0, T]$ is partitioned in time steps, which are analogous to dt in the continuum time integrals. Several algorithms have been applied in the unsteady simulations for the Navier-Stokes equations, such as the Runge-Kutta scheme [85], and for the structural mechanics equations, e.g. the Newmark's method [86]. A broader family of time integration scheme, named Generalized-α (G-α), was initially introduced by Chung and Hulbert for structural dynamics [87]. Jansen et al.

further developed the method to be applied to the Navier-Stokes equations [88], while Bazilevs et al. adapted the method for the full FSI equation set [89]. The peculiarity of this integration scheme is the possibility to be adapted (generalized, as the name suggests) to a wide variety of different time discretization schemes, both implicit and explicit, by performing the tuning of the constant parameters $\alpha_m, \alpha_f, \beta, \gamma$. The scheme is able to ensure unconditionally stable second order accuracy when

$$\gamma = \frac{1}{2} + \alpha_m - \alpha_f$$
$$\beta = \frac{1}{4}(1 + \alpha_m - \alpha_f)^2$$
$$\alpha_m \geq \alpha_f \geq \frac{1}{2}$$

that are usually parametrized by ρ_∞, known as the spectral radius for infinitely large time step size, in order to damp high frequency dissipation, and which is suggested to be set as 0 for both the Navier-Stokes and structural mechanics equations [89], obtaining

$$\alpha_m = \frac{1}{2}\left(\frac{3 - \rho_\infty}{1 + \rho_\infty}\right) = \frac{1}{2}$$
$$\alpha_f = \frac{1}{1 + \rho_\infty} = \frac{1}{2}$$

In the G-α scheme, the unknowns of the FSI equations are evaluated in two stages in a predictor multicorrector algorithm with a Newton-Raphson method.

At a given time n, the first operation to be performed is the evaluation (the prediction) of the $n+1$ quantities at the predictor stage. At the predictor stage the quantities at $n+1$, for the first Newton-Raphson iteration (superscript 0) are evaluated as a function of the previous time step quantities and the parameters. A "same-velocity" predictor

yields, for all the unknowns of the FSI problem:

$$\dot{\boldsymbol{U}}_{n+1}^{0} = \frac{(\gamma-1)}{\gamma}\dot{\boldsymbol{U}}_{n} \tag{4.78}$$

$$\boldsymbol{U}_{n+1}^{0} = \boldsymbol{U}_{n} \tag{4.79}$$

$$\boldsymbol{P}_{n+1}^{0} = \boldsymbol{P}_{n} \tag{4.80}$$

$$\ddot{\boldsymbol{Y}}_{n+1}^{0} = \frac{(\gamma-1)}{\gamma}\ddot{\boldsymbol{Y}}_{n} \tag{4.81}$$

$$\dot{\boldsymbol{Y}}_{n+1}^{0} = \dot{\boldsymbol{Y}}_{n} \tag{4.82}$$

$$\boldsymbol{Y}_{n+1}^{0} = \boldsymbol{Y}_{n} + \Delta t_{n}\dot{\boldsymbol{Y}}_{n} + \frac{\Delta t_{n}^{2}}{2}\left((1-2\beta)\ddot{\boldsymbol{Y}}_{n} + 2\beta\frac{(\gamma-1)}{\gamma}\ddot{\boldsymbol{Y}}_{n}\right) \tag{4.83}$$

$$\ddot{\hat{\boldsymbol{Y}}}_{n+1}^{0} = \frac{(\gamma-1)}{\gamma}\ddot{\hat{\boldsymbol{Y}}}_{n} \tag{4.84}$$

$$\dot{\hat{\boldsymbol{Y}}}_{n+1}^{0} = \dot{\hat{\boldsymbol{Y}}}_{n} \tag{4.85}$$

$$\hat{\boldsymbol{Y}}_{n+1}^{0} = \hat{\boldsymbol{Y}}_{n} + \Delta t_{n}\dot{\hat{\boldsymbol{Y}}}_{n} + \frac{\Delta t_{n}^{2}}{2}\left((1-2\beta)\ddot{\hat{\boldsymbol{Y}}}_{n} + 2\beta\frac{(\gamma-1)}{\gamma}\ddot{\hat{\boldsymbol{Y}}}_{n}\right) \tag{4.86}$$

An iterative solver inverts the tangent matrices associated with the linearization of the FSI equations at an intermediate time level, defined as a function of the G-α parameters. At that specific time step, the quantities are defined as:

$$\dot{\boldsymbol{U}}_{n+\alpha_m} = \dot{\boldsymbol{U}}_{n} + \alpha_m(\dot{\boldsymbol{U}}_{n+1} - \dot{\boldsymbol{U}}_{n}) \tag{4.87}$$

$$\boldsymbol{U}_{n+\alpha_f} = \boldsymbol{U}_{n} + \alpha_f(\boldsymbol{U}_{n+1} - \boldsymbol{U}_{n}) \tag{4.88}$$

$$\boldsymbol{P}_{n+1} = \boldsymbol{P}_{n+1} \tag{4.89}$$

$$\ddot{\boldsymbol{Y}}_{n+\alpha_m} = \ddot{\boldsymbol{Y}}_{n} + \alpha_m(\ddot{\boldsymbol{Y}}_{n+1} - \ddot{\boldsymbol{Y}}_{n}) \tag{4.90}$$

$$\dot{\boldsymbol{Y}}_{n+\alpha_f} = \dot{\boldsymbol{Y}}_{n} + \alpha_f(\dot{\boldsymbol{Y}}_{n+1} - \dot{\boldsymbol{Y}}_{n}) \tag{4.91}$$

$$\boldsymbol{Y}_{n+\alpha_f} = \boldsymbol{Y}_{n} + \alpha_f(\boldsymbol{Y}_{n+1} - \boldsymbol{Y}_{n}) \tag{4.92}$$

$$\ddot{\hat{\boldsymbol{Y}}}_{n+\alpha_m} = \ddot{\hat{\boldsymbol{Y}}}_{n} + \alpha_m(\ddot{\hat{\boldsymbol{Y}}}_{n+1} - \ddot{\hat{\boldsymbol{Y}}}_{n}) \tag{4.93}$$

$$\dot{\hat{\boldsymbol{Y}}}_{n+\alpha_f} = \dot{\hat{\boldsymbol{Y}}}_{n} + \alpha_m(\dot{\hat{\boldsymbol{Y}}}_{n+1} - \dot{\hat{\boldsymbol{Y}}}_{n}) \tag{4.94}$$

$$\hat{\boldsymbol{Y}}_{n+\alpha_f} = \hat{\boldsymbol{Y}}_{n} + \alpha_f(\hat{\boldsymbol{Y}}_{n+1} - \hat{\boldsymbol{Y}}_{n}) \tag{4.95}$$

A direct substitution of Eq.s (4.87-4.94) in Eq.s (4.74 - 4.77), with the additional substitution of the quantities derivatives at $n+1$ with the Newmark's formulas

$$\boldsymbol{U}_{n+1} = \boldsymbol{U}_n + \Delta t_n \left((1-\gamma)\boldsymbol{U}_n + \gamma \dot{\boldsymbol{U}}_n \right) \tag{4.96}$$

$$\dot{\boldsymbol{Y}}_{n+1} = \dot{\boldsymbol{Y}}_n + \Delta t_n \left((1-\gamma)\ddot{\boldsymbol{Y}}_n + \gamma \ddot{\boldsymbol{Y}}_n \right) \tag{4.97}$$

$$\boldsymbol{Y}_{n+1} = \boldsymbol{Y}_n + \Delta t_n \dot{\boldsymbol{Y}}_n + \frac{\Delta t_n^2}{2} \left((1-2\beta)\ddot{\boldsymbol{Y}}_n + 2\beta \ddot{\boldsymbol{Y}}_{n+1} \right) \tag{4.98}$$

$$\dot{\hat{\boldsymbol{Y}}}_{n+1} = \dot{\hat{\boldsymbol{Y}}}_n + \Delta t_n \left((1-\gamma)\ddot{\hat{\boldsymbol{Y}}}_n + \gamma \ddot{\hat{\boldsymbol{Y}}}_n \right) \tag{4.99}$$

$$\hat{\boldsymbol{Y}}_{n+1} = \hat{\boldsymbol{Y}}_n + \delta t_n \dot{\hat{\boldsymbol{Y}}}_n + \frac{\Delta t_n^2}{2} \left((1-2\beta)\ddot{\hat{\boldsymbol{Y}}}_n + 2\beta \ddot{\hat{\boldsymbol{Y}}}_{n+1} \right) \tag{4.100}$$

will yield the complete linearized FSI system of equation solved for the increments of the unknowns $\Delta\dot{\boldsymbol{U}}^i_{n+1}, \Delta\boldsymbol{P}^i_t, \Delta\ddot{\boldsymbol{Y}}^i_{n+1}, \Delta\ddot{\hat{\boldsymbol{Y}}}^i_{n+1}$, where the i is the $i-th$ Newton Raphson non linear iteration. A detailed overview on the system will be presented and discussed in the following section.

When the solutions of the nonlinear iterations converged up to a desired tolerance, the solution of the quantities at a previous time step can be updated with their increments:

$$\dot{\boldsymbol{U}}^{i+1}_{n+1} = \dot{\boldsymbol{U}}^i_{n+1} + \Delta\dot{\boldsymbol{U}}^i_{n+1} \tag{4.101}$$

$$\boldsymbol{U}^{i+1}_{n+1} = \boldsymbol{U}^i_{n+1} + \gamma\Delta t_n \Delta\dot{\boldsymbol{U}}^i_{n+1} \tag{4.102}$$

$$\boldsymbol{P}^{i+1}_{n+1} = \boldsymbol{P}^i_{n+1} + \Delta\boldsymbol{P}^i_{n+1} \tag{4.103}$$

$$\ddot{\boldsymbol{Y}}^{i+1}_{n+1} = \ddot{\boldsymbol{Y}}^i_{n+1} + \Delta\ddot{\boldsymbol{Y}}^i_{n+1} \tag{4.104}$$

$$\dot{\boldsymbol{Y}}^{i+1}_{n+1} = \dot{\boldsymbol{Y}}^i_{n+1} + \gamma\Delta t_n \Delta\ddot{\boldsymbol{Y}}^i_{n+1} \tag{4.105}$$

$$\boldsymbol{Y}^{i+1}_{n+1} = \ddot{\boldsymbol{Y}}^i_{n+1} + \beta\Delta t_n^2 \Delta\ddot{\boldsymbol{Y}}^i_{n+1} \tag{4.106}$$

$$\ddot{\hat{\boldsymbol{Y}}}^{i+1}_{n+1} = \ddot{\hat{\boldsymbol{Y}}}^i_{n+1} + \Delta\ddot{\hat{\boldsymbol{Y}}}^i_{n+1} \tag{4.107}$$

$$\dot{\hat{\boldsymbol{Y}}}^{i+1}_{n+1} = \dot{\hat{\boldsymbol{Y}}}^i_{n+1} + \gamma\Delta t_n \Delta\ddot{\hat{\boldsymbol{Y}}}^i_{n+1} \tag{4.108}$$

$$\hat{\boldsymbol{Y}}^{i+1}_{n+1} = \ddot{\hat{\boldsymbol{Y}}}^i_{n+1} + \beta\Delta t_n^2 \Delta\ddot{\hat{\boldsymbol{Y}}}^i_{n+1} \tag{4.109}$$

$$\tag{4.110}$$

A more exhaustive formulation of the G-α algorithm and its application for the FSI problem can be found in [33].

4.9 FSI system of equations

The non-linear system of equations given by Eq.s (4.74 - 4.77) can be solved iteratively, in the time step, by adopting a two-step predictor-multicorrector algorithm, in a Newton-Raphson method fashion. At the beginning of the time step $t = n+1$, for the non-linear iteration $i = 0, 1..., (i_{max} - 1)$, at the predictor stage an initial guess of the solution at $t = n + 1$ is obtained as a function of the previous, already evaluated, time step n. Inside the non-linear iteration, at the multicorrector stage, the quantities are evaluated at the intermediate time step as in Eq.s (4.87-4.94), from which the linearized system is assembled and written in terms of tangent matrices, and solved for the increments of the unknown quantities for i non linear iterations. At the end of the non linear iteration, the solution is updated by adding the last evaluated increment to the solution of the previous non linear iteration.

Inside the Newton-Raphson loop, for each iteration at the multi-corrector stage, the quantities are evaluated at the intermediate time step as described in Eq.s (4.87-4.94). Consecutively, the residual formulation expressed by Eq.s (4.74-4.77) is completely linearized into a system of linear equations solved for the increments of the unknowns of every subsystem:

$$\left.\frac{\partial \boldsymbol{N}_{1M}}{\dot{\boldsymbol{U}}_{n+1}}\right|_i \Delta\dot{\boldsymbol{U}}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{1M}}{\boldsymbol{P}_{n+1}}\right|_i \Delta\boldsymbol{P}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{1M}}{\ddot{\boldsymbol{Y}}_{n+1}}\right|_i \Delta\ddot{\boldsymbol{Y}}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{1M}}{\ddot{\hat{\boldsymbol{Y}}}_{n+1}}\right|_i \Delta\ddot{\hat{\boldsymbol{Y}}}_{n+1} = -\boldsymbol{N}^i_{1M}\,, \quad (4.111)$$

$$\left.\frac{\partial \boldsymbol{N}_{1C}}{\dot{\boldsymbol{U}}_{n+1}}\right|_i \Delta\dot{\boldsymbol{U}}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{1C}}{\boldsymbol{P}_{n+1}}\right|_i \Delta\boldsymbol{P}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{1C}}{\ddot{\boldsymbol{Y}}_{n+1}}\right|_i \Delta\ddot{\boldsymbol{Y}}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{1C}}{\ddot{\hat{\boldsymbol{Y}}}_{n+1}}\right|_i \Delta\ddot{\hat{\boldsymbol{Y}}}_{n+1} = -\boldsymbol{N}^i_{1C}\,, \quad (4.112)$$

$$\left.\frac{\partial \boldsymbol{N}_{2}}{\dot{\boldsymbol{U}}_{n+1}}\right|_i \Delta\dot{\boldsymbol{U}}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{2}}{\boldsymbol{P}_{n+1}}\right|_i \Delta\boldsymbol{P}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{2}}{\ddot{\boldsymbol{Y}}_{n+1}}\right|_i \Delta\ddot{\boldsymbol{Y}}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{2}}{\ddot{\hat{\boldsymbol{Y}}}_{n+1}}\right|_i \Delta\ddot{\hat{\boldsymbol{Y}}}_{n+1} = -\boldsymbol{N}^i_{2}\,, \quad (4.113)$$

$$\left.\frac{\partial \boldsymbol{N}_{3}}{\dot{\boldsymbol{U}}_{n+1}}\right|_i \Delta\dot{\boldsymbol{U}}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{3}}{\boldsymbol{P}_{n+1}}\right|_i \Delta\boldsymbol{P}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{3}}{\ddot{\boldsymbol{Y}}_{n+1}}\right|_i \Delta\ddot{\boldsymbol{Y}}_{n+1} + \left.\frac{\partial \boldsymbol{N}_{3}}{\ddot{\hat{\boldsymbol{Y}}}_{n+1}}\right|_i \Delta\ddot{\hat{\boldsymbol{Y}}}_{n+1} = -\boldsymbol{N}^i_{3} \quad (4.114)$$

In Eq.s (4.111-4.114), each component can be expressed as a tangent matrix evaluated at the intermediate time level, and in particular, using the index notation:

$$\begin{aligned}
\frac{\partial \boldsymbol{N}_{1M}}{\dot{\boldsymbol{U}}_{n+1}} &= \alpha_m \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} N_A \rho N_B d\Omega \delta_{ij} \\
&+ \alpha_m \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} \tau_{SUPS} \left(\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h \right) \cdot \boldsymbol{\nabla} N_A \rho N_B d\Omega \delta_{ij} \\
&+ \alpha_f \gamma \Delta t_n \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} N_A \rho \left(\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h \right) \cdot \boldsymbol{\nabla} N_B d\Omega \delta_{ij} \\
&+ \alpha_f \gamma \Delta t_n \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} \boldsymbol{\nabla} N_A \cdot \mu \boldsymbol{\nabla} N_B d\Omega \delta_{ij} \\
&+ \alpha_f \gamma \Delta t_n \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} \boldsymbol{\nabla} N_A \cdot \boldsymbol{e}_i \mu \boldsymbol{\nabla} N_B \cdot \boldsymbol{e}_j d\Omega \delta_{ij} \\
&+ \alpha_f \gamma \Delta t_n \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} \tau_{SUPS} \left(\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h \right) \cdot \boldsymbol{\nabla} N_A \rho \left(\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h \right) \cdot \boldsymbol{\nabla} N_B d\Omega \delta_{ij} \\
&+ \alpha_f \gamma \Delta t_n \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} \rho \nu_{LSIC} \boldsymbol{\nabla} N_A \cdot \boldsymbol{e}_i \mu \boldsymbol{\nabla} N_B \cdot \boldsymbol{e}_j d\Omega \delta_{ij} \,, \qquad (4.115) \\
\frac{\partial \boldsymbol{N}_{1M}}{\boldsymbol{P}_{n+1}} &= - \int_{\Omega_{t_{(n+\alpha_f)}}} \boldsymbol{\nabla} N_A \cdot \boldsymbol{e}_i N_B d\Omega \\
&+ \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} \tau_{SUPS} \left(\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h \right) \cdot \boldsymbol{\nabla} N_A \boldsymbol{\nabla} N_B \cdot \boldsymbol{e}_i d\Omega \delta_{ij} \,, \qquad (4.116) \\
\frac{\partial \boldsymbol{N}_{1C}}{\dot{\boldsymbol{U}}_{n+1}} &= \alpha_f \gamma \Delta t_n \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} N_A \boldsymbol{\nabla} N_B \cdot \boldsymbol{e}_j d\Omega \\
&+ \alpha_f \gamma \Delta t_n \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} \tau_{SUPS} \boldsymbol{\nabla} N_A \cdot \boldsymbol{e}_j \left(\boldsymbol{u}^h - \hat{\boldsymbol{u}}^h \right) \cdot \boldsymbol{\nabla} N_B d\Omega \\
&+ \alpha_m \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} \tau_{SUPS} \boldsymbol{\nabla} N_A \cdot \boldsymbol{e}_j N_B d\Omega \,, \qquad (4.117) \\
\frac{\partial \boldsymbol{N}_{1C}}{\boldsymbol{P}_{n+1}} &= \int_{(\Omega_1)_{t_{(n+\alpha_f)}}} \frac{\tau_{SUPS}}{\rho} \boldsymbol{\nabla} N_A \cdot \boldsymbol{\nabla} N_B d\Omega \,, \qquad (4.118)
\end{aligned}$$

$$\frac{\partial \boldsymbol{N}_2}{\ddot{\boldsymbol{Y}}_{n+1}} = \alpha_m \int_{(\Omega_2)_0} N_A(\rho_2)_0 N_B d\Omega \delta_{ik} + \alpha_f \beta \Delta t_n^2 \int_{(\Omega_2)_0} \frac{\partial N_A}{\partial X_J} D^X_{iJkL} \frac{\partial N_B}{\partial X_L} d\Omega \,, \tag{4.119}$$

$$\frac{\partial \boldsymbol{N}_3}{\ddot{\hat{\boldsymbol{Y}}}_{n+1}} = \int_{(\Omega_1)_0} \frac{\partial N_A}{\partial X_J} D^{hX}_{iJkL} \frac{\partial N_B}{\partial X_L} d\Omega \,. \tag{4.120}$$

It should be noted that the integrals are performed in the spatial domain at time $t_(n+\alpha_f)$ for the fluid domain while the structural and moving mesh tangent matrices are evaluated in the reference configuration according to the Total Lagrangian formulation. A diligent reader would notice that not all the members of Eq.s (4.111-4.114) are part of the linearized system described by the matrices of Eq.s (4.115 - 4.120). In fact $\frac{\partial \boldsymbol{N}_{1M}}{\partial \ddot{\boldsymbol{Y}}_{n+1}}$, $\frac{\partial \boldsymbol{N}_{1C}}{\partial \ddot{\boldsymbol{Y}}}$, $\frac{\partial \boldsymbol{N}_2}{\boldsymbol{P}_{n+1}}$, $\frac{\partial \boldsymbol{N}_3}{\boldsymbol{P}_{n+1}}$ and $\frac{\partial \boldsymbol{N}_2}{\partial \ddot{\hat{\boldsymbol{Y}}}_{n+1}}$ are not evaluated as they represent a derivation by a quantity which is not part of the specific subsystem or domain, while $\frac{\partial \boldsymbol{N}_{1M}}{\partial \ddot{\hat{\boldsymbol{Y}}}_{n+1}}$ and $\frac{\partial \boldsymbol{N}_{1C}}{\partial \ddot{\hat{\boldsymbol{Y}}}_{n+1}}$, the so called *shape derivatives*, are proven to have negligible effects on the convergence of the system [89], and are consecutively removed to save computations.

4.10 Strongly coupled, block iterative solution strategy

The solution strategy adopted for the FSI solver in FEMpar to solve the Eq.s (4.70 - 4.73) linearized in Eq.s (4.115 - 4.120) is a strongly coupled, block iterative algorithm, as it was briefly described in Sec. 3.2. Strong coupling FSI techniques have been developed and applied in last decades to study a wide range of problems, from full-size wind turbines [90], to parachutes [26] or bio-engineering applications [91]. Less often are found applications of time-running, non-linear and fully coupled FSI, in the turbomachinery field. Indeed, a "weak coupling" approach (see e.g. [92]) or modal-based analysis (e.g. [93]) is often preferred in turbomachinery FSI analysis, because the stiffness of the blades is generally very high and the aeroelastic feedback is often a secondary effect with respect to the inertial forces. The algorithm can provide the robustness in the solution given by the strongly coupling approach, where all the sub-systems are solved in the same time step, and the flexibility and easiness to use guaranteed by the segregated solvers called sequentially: in fact it is possible to use in the FSI solution algorithm the same solvers developed for the standalone simulations of fluid dynamics and solid mechanics, just by calling them when needed; furthermore, each matrix of the coupled system can be individually preconditioned, and no custom, ad-hoc preconditioner has to be designed and implemented. Unfortunately, such flexibility is obtained with a series of shortcomings: a direct approach, where one unique large matrix is assembled by adding the contribution of all the equations (or a quasi-direct approach, where the fluid system and solid system are written in one matrix, while the moving mesh is

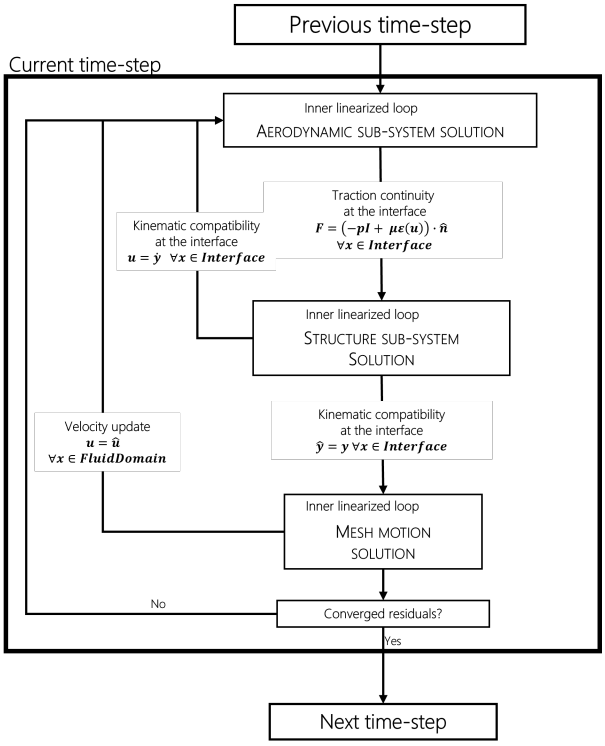

FIGURE 4.1: Block iterative solver, strongly coupled approach

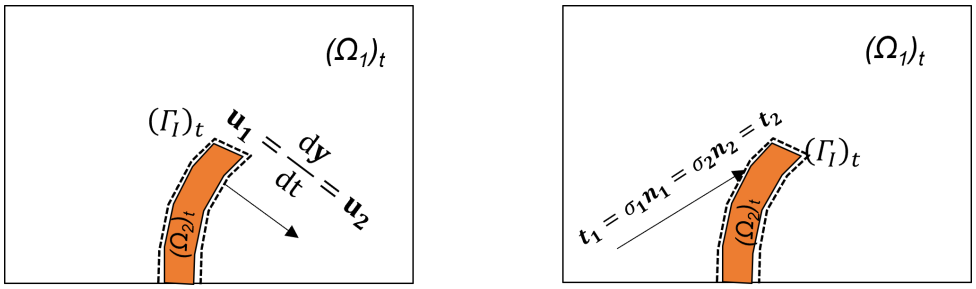

FIGURE 4.2: Traction continuity and kinematic compatibility conditions

solved separately) is in general more robust and affected by very few (in some cases, by none) convergence problems, especially when there's a large difference between the mass of the fluid and the mass of solid. However, the advantages of the segregated approach, at such an early stage of the development, are considered extremely valuable and for this reason, the block iterative approach is chosen.

In the block-iterative approach, the sub systems (the "blocks") are called sequentially inside the time step, and multiple times according to the number of requested non-linear iterations. A scheme of the algorithm can be found in Fig. 4.1.

The aforementioned interface conditions, the traction continuity and the kinematic compatibility represented in Eq.s (4.65) and (4.69) must be established between the three subsystem according to the scheme in Fig. 4.1, to ensure a consistent representation of the FSI dynamics. A schematic representation of the interface conditions is presented

in Fig. 4.2.

5. FEMpar: a custom FSI simulation software

As a core part of the present work, the FSI simulation software FEMpar has been entirely developed from scratch in the last three years. FEMpar is a C++ MPI software for CFD simulations, structural analysis and fully coupled FSI simulations. It was specifically designed to resolve specific issues that limited the applicability, functionality and scalability of its predecessor Xenios++. Xenios++ was the last iteration of the CFD software Xenios, developed by the research group of Prof. Rispoli in the last 30 years, and it was able to perform CFD and FSI simulations producing good results for flows in turbomachinery [94] , passive morphing [29], multiphase combustion [95], particle tracking [96], fluid structure interaction [31] and erosion analysis in turbomachinery [97]. While the quality of the models implemented in Xenios++ were never questioned, it suffered a series of flaws that could not be resolved if not by completely redesigning the entire software:

- the memory management of the software was not suitable to achieve good scalability for large size problems in large computational clusters, due to the strategy adopted to handle the decomposed domain;
- the external libraries for the FE discretization, such as the open source FE library LibMesh [98], in spite of the fact of being well written and extremely useful to speed up the development time, are a limitation when for research interests the developer wants to have the complete control over the discretization model, the integration scheme, the solution strategy. These libraries in fact are very complete but only a small fraction of them were effectively used in Xenios++, while the rest of the library was unused and it was constraining the development;
- the long development history of Xenios++, from its Fortran77 origins to the last C++ version, lead to a software characterised by several layers of subsequent development, that resulted in numerous sections of the code where implementations of new models were extremely difficult if not impossible at all;

For such reasons the development of FEMpar was focused from its beginning on the solution of the critical issues of its predecessor, while conserving most of the mathematical models and the backbone of the solution algorithm. The resulting software, with the models implemented as described in Chap. 4, is able to perform CFD, FEA, and FSI simulations for complex geometries (turbomachinery rotors, thin structures, internal flows of enclosed devices) describing complex, unsteady and nonlinear physics phenomena such as fluid structure interaction and turbulence on multiple spatial scales, with the great advantage in terms of meshing using tetrahedral elements in unstructured meshes. The core algorithm of the software has been successfully tested at the CINECA HPC infrastructure, in a scalability test of 1024 cores. The software has been successfully used in FSI simulations in [31, 30, 99].

In spite of the fact that the state of the art of available software for CFD and FEA simulations presents a large number of valid alternatives to perform reliable, high resolution simulations, such as for example the Ansys suite [100], STARCCM [101], NUMECA [102] with commercial licenses, and OpenFOAM [103] with open source licence, it is not uncommon that academic research in numerical analysis for fluid Dynamics and structural dynamics is supported by custom, *ad-hoc* developed software. Even though it might seem, at a first glance, a counter intuitive and inefficient choice, it presents a non negligible, large number of advantages:

- complex multi-physics phenomena are not always handled by commercial software, which are usually more prone to satisfy the requirements of the industry, and not of an academic audience;
- the flexibility provided by simulation software widely adopted in the industry is often not needed by every research group in the same way;
- the complete control over libraries, dependencies, version control, compiling methods and installation procedures allows an easy installation on older machines, sometimes with very different architectures and sizes, and enhance portability on different operating systems;
- it is often required to implement specific, innovative models that sometimes do not fit the already existing software architecture, and could not be added to the software;
- in an academic environment, a custom software can be also considered as a tool to support scientific programming and numerical analysis lectures, or student projects, so that students could improve their knowledge by practicing on every aspect of the numerical analysis, from design and development of specific features, to testing and practical use.

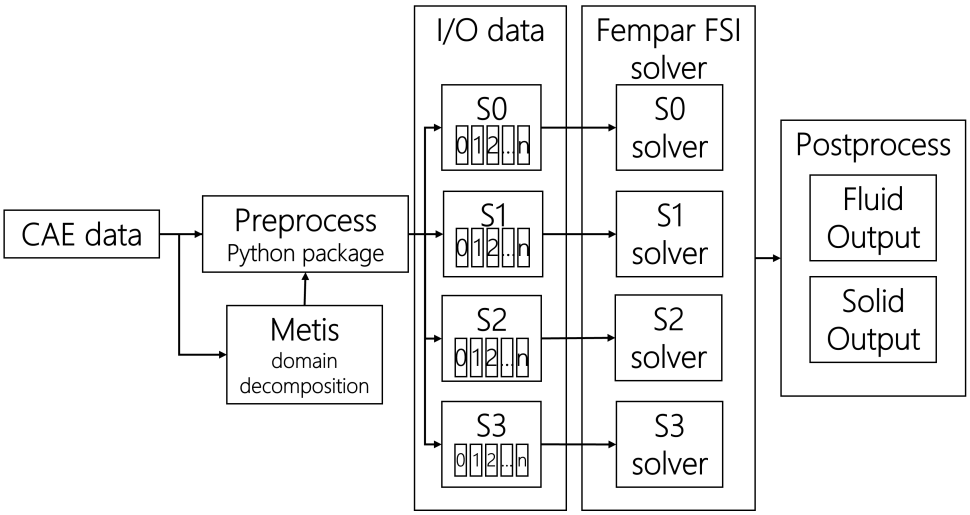

FIGURE 5.1: Flow diagram for FEMpar

In the following sections, a brief overview of the software will be presented, with its main components and the strategies adopted for the development of FEMpar.

5.1 Software structure

FEMpar is composed of two separate and different parts, differentiated for their main purpose and consequently by the language adopted for them: a set of preprocessing tools written in Python handle the I/O, the domain decomposition and the setup of the boundary conditions, while the main FSI solver in C++ handles the discretization, the time integration, the FE assembly loop and the inversion of the matrix for the solution of the linearized system of equations.

The preprocessing is a one-time operation, and the solver can be executed as a standalone program where input options are passed by a simple command line, or by an input file. The choice of separating the program in this two main components led to a more flexible and easier setup during the debug, test and validation stages, and also during the daily use of FEMpar, because it is fairly common to use the same computational domain and boundary conditions for different operating conditions, or to repeat the same computation for debug and test purposes. In Fig. 5.1, a block diagram of the main components of FEMpar and their main characteristics is presented. FEMpar can be executed, according to the user's needs, as a CFD software for the incompressible Navier-Stokes equations, as a FEA structural solver, or as a FSI analysis software. The user can select each case starting from the preprocessing tools, and its inputs will be carried on up to the C++ solver.

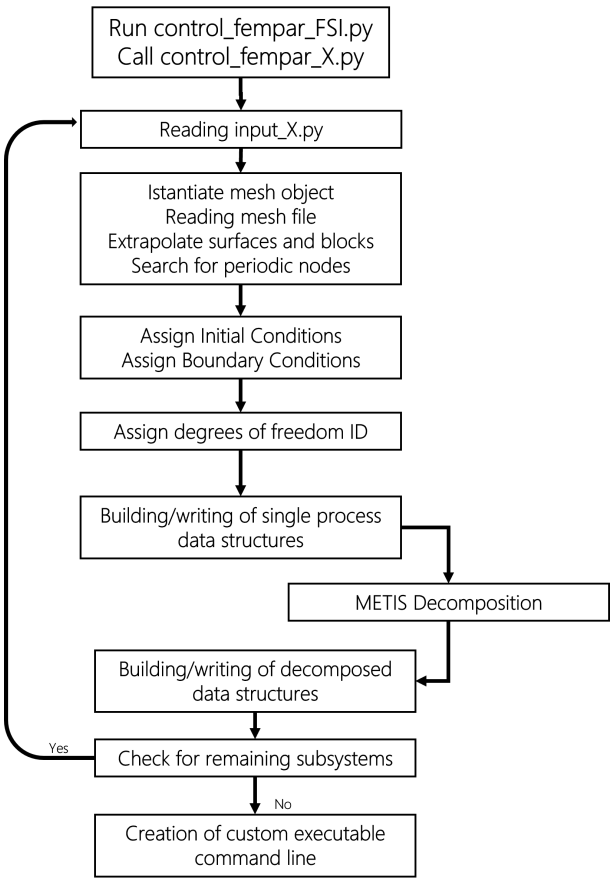

FIGURE 5.2: Flow diagram for FEMpar preprocessing

5.1.1 Preprocessing tools

The preprocessing tools consist of a Python package that makes extensive use of the libraries numpy and Pandas [104] due to the extreme speedup obtainable with the vectorized operations allowed by those libraries, while not giving up on the friendliness and easiness to use of the Python language. The package consists of a set of files, most of which differentiated by the system of equation they refers to (in particular, the *X* in the file name should be replaced with the specific ID, from 0 to 2, for fluid, turbulent, structural, moving mesh preprocessing script):

- `inputX.py`: a series of input files, where the user specifies the mesh files, the number of CPUs for the C++ execution and several physical and numerical quantities, that will be required at later stages;

- `control_fempar_FSI.py`: a main file, where each controller is called accordingly to the simulation that the user is interested in performing: with respect to the flags specified in the input files and in the main file script, the user can execute the preprocessing routine for a pure fluid dynamics unsteady simulation, a pure solid

dynamics unsteady simulation or a strongly coupled FSI unsteady simulation in the ALE domain;

- `control_fempar_X.py`: a series of controller files, one for each physical and logical subsystem, where the user specifies the initial conditions, and the boundary conditions addressing the topological entities (i.e. volumes and surfaces obtained by a meshing software) read from the mesh file, and where the preprocessing routine is effectively executed;
- `utils3D.py`: a file where a parent class is managing all the mesh information and storing all the preprocessing utilities and functions, and its objects are instantiated by all the other files as mesh objects directly related to the specific subsystem;
- a compiled installation of the software METIS [105] for a fast and reliable element-based domain decomposition;

The preprocessing routine writes all the data structure needed for each subsystem in a separate folder, one for each subsystem. Each preprocessing run sets up the data structures needed for both a serial (single processor unit) simulation and a parallel MPI (multiple processor units) simulation: this is not only done to save time by avoiding to perform multiple preprocessing runs in the test and debug phases (since it is way more convenient to check the implementation of a new feature in serial, and only later check the parallel , but also because some serial data structures are required for the parallel simulation and preprocessing.

5.1.2 Assignment of initial conditions and boundary conditions

At the current stage of development, the boundary conditions are assigned at the preprocessing stage in Python. This is done for several reasons: it would be pointless to assign boundary conditions multiple times, at the beginning of each FEMpar execution (for example, if the simulation needs to be interrupted and then restarted from the last iteration, there would be no reason at all to assign the boundary conditions again); it is easier to manipulate large datasets using pandas and numpy, than to create ad-hoc classes and objects in C++, as well the function required for each dataset manipulation; due to the fact that no Graphic User Interface (GUI) is provided to the user at the current stage of development, the presence of an interpreted (i.e. non compiled), high level language as Python allows the user to constantly modify the controller files (where the boundary conditions are assigned) to suite the specific simulation, reducing the debug and test time to a minimum. After the input files are read, a Dataframe is built for each specific boundary condition (or initial condition), by collecting any surface or volume previously defined in the mesh software with a characteristic and unique name.

Multiple surfaces or volume can be assigned to the same boundary or initial condition, as long as an indication of the specific degrees of freedom to be constrained, and the assigned values, are given as function parameters. In the last version of FEMpar, it is possible to assign in this routine any Dirichlet and Neumann condition: constant velocity profile, turbulent velocity profile, rotating domains, relative velocity profiles for rotating domains, slip and no slip conditions, structural displacements, distributed forces.

Additional conditions can be applied using a combination of preprocessing tools and runtime operations: for example, the conservation of the mass flow rate can be imposed by exporting a desired surface from the preprocessing in a file containing the surface connectivity and an id referring to the specific side of the 3D element that is lying on that surface, and at runtime a finite element integration of the velocity on that surface would evaluate at each iteration the mass flow rate, and assign to the domain a volume force that would compensate for each loss due to the viscous effects, and frictions at the wall. Using analogous methods is possible to have simulations, for example, with inlet profiles that are variable in time, or with weakly enforced boundary conditions.

5.1.3 FEMpar solvers structure

The core of the software FEMpar revolves around its C++ solvers and classes for fluid dynamics, solid dynamics, moving mesh and fluid structure interaction. At the current stage of development, FEMpar presents a total of four solvers, called by a main file according to the requested simulation, and supported by several classes, designed to handle complex data structures or specific features of the finite elements method:

- `main.cpp`: main file where each sequence of solver is called accordingly to the requested simulation through a parse of the command line arguments;
- `solver_unsteady.cpp`: solver for the fluid dynamics sub-system, can be executed as a standalone program when the only CFD simulation (RANS or LES-like with VMS) is needed;
- `solver_unsteady_1.cpp`: solver for the fluid dynamics turbulence sub-system, can be executed as a standalone program when the user desires to enrich the RANS simulation with a turbulence model, such as the one-equation model of Spalart and Almaras [21] (models that relies on two or more equations, such as $\kappa - \epsilon$ [22]) or $\kappa - \omega SST$ [23]) , can be easily implemented);

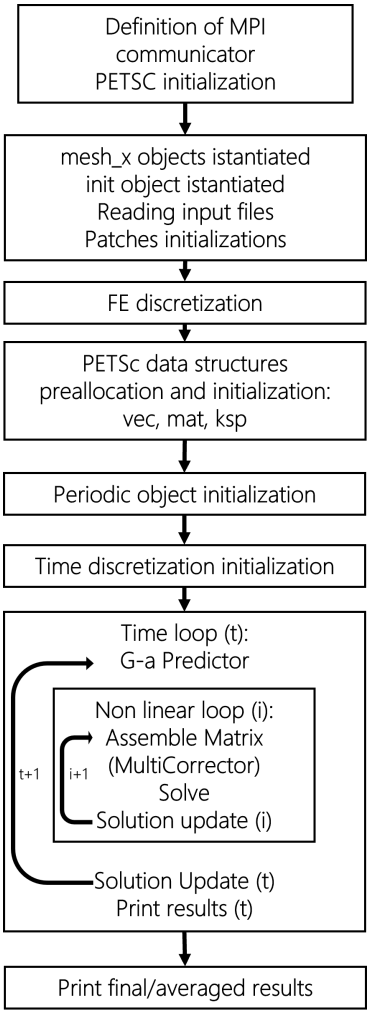

FIGURE 5.3: Flow diagram for FEMpar FSI solver

- `solver_unsteady_2.cpp`: solver for the solid dynamics sub-system, can be executed as a standalone program when the only non-linear elastic structural simulation is needed;
- `solver_unsteady_FSI.cpp`: solver for the strongly coupled FSI with block iterative coupling algorithm, can be executed as a standalone program when the user desires to perform a fluid structure interaction simulation using all the models available to the other software; this specific solver is able to call the assembly and solve functions of the fluid and solid solvers, and it implements the moving mesh subsystem at the end; in Fig. 5.3 is reported the flow diagram for the FSI solver: the other solvers follow a similar and simpler diagram;
- `AssembleMatrix_X.cpp`: class that handles the finite elements assembly algorithm, where each function is specialized with respect to the specific model that the user would like to adopt; in FEMpar there are four different assemble matrix classes, one for each sub-system (where the X in the file name can be 0-1-2-3 according to the sub-system the class is referring to);
- `Fem.cpp`: class that instantiates and manages all the data structures required for the discretization using finite elements (e.g. weight functions for the 3D element and its faces, weight function derivatives, gaussian quadrature);
- `models.cpp`: class that collect all the mathematical and physical models implemented in FEMpar, for all solvers and subsystems;
- `MeshFMP.cpp`: class that instantiates, manages, and manipulates the mesh data, its decomposition, and the solution vectors, making the mesh and its data available for all solvers when needed; a mesh object must be instantiated for each subsystem and will be accessed by all the other classes; in addition to that, the class also reads the files obtained from the preprocessing and sets up and writes the solution files for the postprocessing;
- `periodic.cpp`: class that manages the periodic (also known as cyclic) condition for the computational domain; it can handle rotational and translational periodicity (in multiple directions at the same time);
- `Init.cpp`: classes that checks and parses the command line used to execute FEMpar; each option is stored with a string and a flag, and by combining the two information the solvers are designed to switch between different behaviours and routines, and consequently to perform different simulations;
- `IO_toolsFMP.cpp`: classes that collects a series of functions extensively used to allocate, set, and deallocate most of the data structures used in FEMpar;

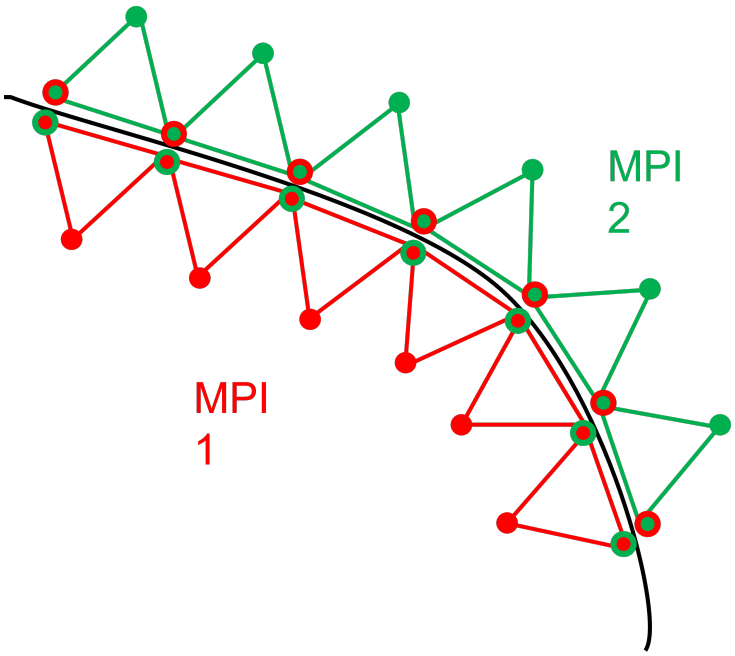

FIGURE 5.4: Non overlapping domain decomposition

5.1.4 Domain decomposition strategy

In FEMpar the domain decomposition algorithm is entirely handled by METIS [105], which performs a fast and efficient domain decomposition using graphs partitioning. In particular, the domain decomposition strategy adopted in FEMpar can be defined as non-overlapping from the perspective of the element, as represented in Fig. 5.4. In such a way it is possible to guarantee a correct assembly of the matrix with no need of the so called "ghosted nodes" [106], simplifying by a large amount both the pre-processing routines and the finite element assembly. As will be discussed in a later section, the finite element assembly strategy adopted in FEMpar by means of the PETSc library is completely parallel, in the sense that each process can assemble the portion of the matrix that is given by its own decomposed portion of the domain. For this reason, the repeated nodes at the interfaces between two sub-domains are only counted once for each element, and since the matrix assembly algorithm for finite elements is elements-based, they are contributing to the final matrix with the correct amount of weight, as they are never counted twice for the same element.

At the interface between solid and fluid, the nodes are required to be matching. However, in FEMpar it is not mandatory that the mesh decomposition is likewise matching: fluid nodes that belongs to a specific process could be matched with solid nodes from a different process, as shown in Fig. 5.5, as well as an example of a decomposed domain. This is achieved using a map between shared nodes that is managed by a master process.

At the end of the simulation, or more in general, every time an output file is required

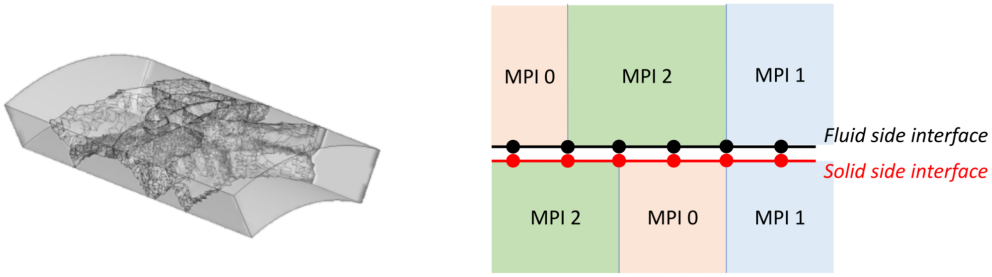

FIGURE 5.5: Example of decomposition at the FSI interface

to be printed for visualization or data postprocessing, a recomposition routine is on charge of gathering all the data, to write an easily manageable output file. The output files are separated in fluid output and solid output.

5.1.5 Matrix preallocation

In order to guarantee an efficient matrix inversion, PETSc requires a perfect allocation of the non-zero entries for the left-hand-side matrix, so that a correct memory allocation is assigned for the assembly and inversion operations. The requirement is directly dependent on the data structures chosen to represent sparse matrices in PETSc, and it is mandatory. If not allocated correctly, the assembly matrix could be slowed down up to 100 times, and even saturate the available memory of the computer during the process. PETSc is able to perform dynamic allocation of the memory required by the matrix, however for large sparse matrix the required time and memory for the process would make the feature simply not usable. For this reason, a parallel preallocation routine has been developed for FEMpar, so that the software is able to have perfect allocation in all the supported scenarios.

In particular, the algorithm is designed to work accordingly to the the domain decomposition performed by METIS: at the preprocessing stage, the information about the specific degrees of freedom (in terms of range of matrix rows indices) which are competent to each subdomain is stored in a file; with this operation, the matrix is sliced in sub-matrices, and each sub-matrix is handled by a single MPI process (this is done in order to reduce the amount of communications between processes, that would increase the time required for the simulation, specifically the time required by the inversion of the matrix); consequently the file is read by a master process, that will assign in a single array the ownership of every node to each subprocess, and that will count for each node the amount of unconstrained degrees of freedom that are represented by that node; the master process then loops over the neighbours of each node, and counts and assigns the amount of non-zero entries which are competent to the diagonal portion (subset of the matrix where both rows and columns represent degrees of freedom that

belong to that specific submatrix) and off-diagonal portion (subset of the matrix where the rows represent degrees of freedom that belong to that specific submatrix, while the columns represent degrees of freedom which belongs to different submatrices) of each submatrix.

5.1.6 Periodic boundary conditions implementation

In order to reduce the size of the computational domain, and consequently the number of elements and nodes required to appropriately discretize the surfaces and volumes defining the domains, it is possible, in some specific applications, to have the definition of a periodic (or cyclic) boundary condition, between a couple of matching surface domains, that could be referred as *master* and *slave* (in no specific order). With such condition, it is possible to impose that two topologically different nodes, one on the *master* patch and the other on the *slave* patch, are corresponding in the logical space of degrees of freedom in the matrix, and consequently, they're corresponding in every aspect of their nature as if they represent the same node: they describe the same fluid velocity, or pressure, or stress or displacement, while still being described by different spatial coordinates in the topological space. This feature has been commonly used in CFD for decades to virtually enlarge the computational domains, when the computing power was not sufficient to elaborate full size domains in a reasonable time, hence symmetries and periodicities has to be exploited to be able to obtain a solution, and it's still widely used today. In addition to that, it must be pointed out that several notorious and highly adopted benchmark or test applications for CFD would not be feasible at all without this specific *escamotage*: above all, the well known channel flow simulation (as an example, the one presented in [107]) strictly requires a periodicity at the sides to represent an indefinitely wide channel in the cross-flow direction. In FEMpar, the periodic condition is expressed by a logical connection between degrees of freedom represented by different nodes in different locations, by placing the same, identical degree of freedom in the two nodes. A strict requirement for this feature to work as intended is that the *master* patch and the *slave* patch and their discretization (their meshes) have to be exactly matching after a rigid motion (rigid translation in a straight direction, or rigid rotation about an axis). The matching is ensured and built via a research algorithm with customizable tolerances. The only critical issue is to ensure that the periodic nodes are not counted twice during the finite element assembly process, as they actually represent the same degree of freedom. Three different periodic conditions are included in FEMpar, and can be imposed at the preprocessing stage:

- *Translational in one direction*: the flow that is crossing the *master* patch, apparently exiting the domain, is reinserted back in the domain from the *slave* patch, or

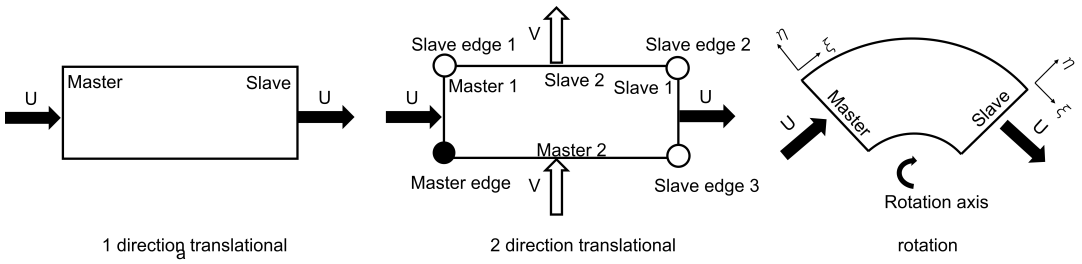

FIGURE 5.6: Periodic conditions implemented in FEMpar (from left to right): translational in one direction, translational in two directions, rotational

viceversa. Each vectorial quantity mantains all its components and direction, while every scalar quantity mantains its signed value. No further adjustement is required to ensure the continuity of the domain, the sole identity of the degree of freedom is sufficient for the purpose;

- *Translational in two directions*: two couples of translational periodicity patches coexists at the same time: there are two *master* and two *slave* patches that can separately ensure continuity. In addition to the mono-directional periodic condition, the presence of a *master* edge and three separate *slave* edges (each linked with the same *master* edge) is required to ensure continuity of the field: identical quantities will be represented at the edges, as if *master* edge were the center of the periodicity.

- *Rotational*: the periodicity is defined between two patches which are matching when transformed by a rigid rotation along a given axis. The coupling between degrees of freedom is obtained similarly to the previous cases, however an additional adjustment has to be made to maintain the consistency of the terms in the matrix: in fact it has to be recalled that the Navier-Stokes equations in FEM are written for an isoparametric element, in the parametric reference frame. Consequently, every vectorial quantities that appear in the equations has to be projected in the Frenet-Serret reference frame (normal versor, tangent versor, and binormal versor) of the 2D element on the periodic patches at the assembly matrix stage, and later, when the solution is found, they are rotated back in the cartesian reference frame before the writing of the output files.

6. Applications

In this chapter, a series of different applications for the CFD and FSI simulations in turbomachinery using FEMpar is presented. For all the applications, a closed loop approach is selected for the solver algorithm, as well as the strong coupling technique already discussed in Chap. 4, which is crucial in order to be able to catch consistently the feedback between the unsteady aerodynamic field and structural response. All the presented applications are extracted from recently published papers.

The chapter is divided in two main parts:

- In the first part, are presented two applications that can be recollected under the definition of aeroelastic analysis: in *Strongly coupled Fluid Structure Interaction simulation of a 3D printed fan rotor* [31], the software XENIOS++ (the predecessor of FEMpar) was used to perform a 3D simulation for an axial fan characterized by low solidity, high aspect ratio, and blades in PLA (PolyLactic Acid), obtained with a 3D printer. The simulation highlighted that due to aerolastic coupling, the behaviour of the fan changed dramatically under different operating conditions; in *Unsteady flow simulation of an axial fan for dry cooling in a CSP plant using the Variational Multiscale Method* [99], the RBVMS method is used with the software FEMpar to perform a CFD simulation for a large axial fan, and the results are compared with experimental results.

- In the second part, two applications are presented as examples of an attempt to introduce passive morphing adaptivity for turbomachinery blades: in *Morphing of reversible axial fan blade: a FSI-FEM study* [108], a 2D FSI simulation performed with FEMpar explores the design of an axial fan with reversible blades obtained by adopting a thin, symmetric profile using an hyperelastic material; in *FSI analysis and simulation of flexible blades in a Wells turbine for wave energy conversion* [30], an analogous approach is adopted for a 2D simulation of a flexible blade in a Wells turbine. In these last two applications, it is more evident how an FSI strongly coupled simulation could be able to express and to represent complex multiphysics dynamics, and dramatically increase the amount of information

available at the design stage, for a given structural configuration, or operating condition.

6.1 Aeroelastic analysis

6.1.1 Strongly coupled Fluid Structure Interaction simulation of a 3D printed fan rotor

Additive manufacturing represents a new frontier in the design and production of rotor machines. This technology drives the engineering research framework to new possibilities of design and testing of new prototypes, reducing costs and time. On the other hand, the fast additive manufacturing implies the use of plastic and light materials, often including a certain level of anisotropy due to the layered deposition. The most used material for the fast printing of scaled models are PLA (Polylactic Acid) and ABS (Acrylonitrile butadiene styrene), which confer to the 3D printed samples a high-flexible structure due to a lower stiffness with respect to the materials used in the final production. For these prototypes, the fluid-structure interaction between the flow field and the blade structure appears more relevant and more complex, introducing a wide variety of non-negligible and non easily predictable aeroelastic effects and flow induced vibrations for high aspect ratio rotors. In this section, it is presented a study regarding the aeroelastic response of a small sample fan blade, printed using PLA material. Scope of the work is to study both the structure and flow field dynamics, where strong coupling is considered on the simulation. The blade has been tested in two operating points, to see the aero-mechanical dynamics of the system in stall and normal operating condition. The computational fluid-structure interaction (FSI) technique is applied to simulate the coupled dynamics.

Fan details and numerical setup

Fan and blade description A FSI simulation is performed on a 3D printed PLA fan blade prototype. The blade is designed in order to have high aspect ratio and to be particularly slender: the thickness ratio spans from $t/c = 0.08$ at the root to 0.04 at the tip, and the symmetric airfoil profile is twisted by 11 degrees along a radius of 0.19 m. The entire rotor is composed by three of such designed blades in a cylindrical shroud, in an unusual configuration with an extremely low solidity. It is well known that such a low solidity would result in a poor operative design, but it can emphasize the fluid structure interaction by increasing the aerodynamic load on each blade.

Fig. 6.1 shows a 3D view of the blade, while some details of the blade design can be found in Table 6.4.

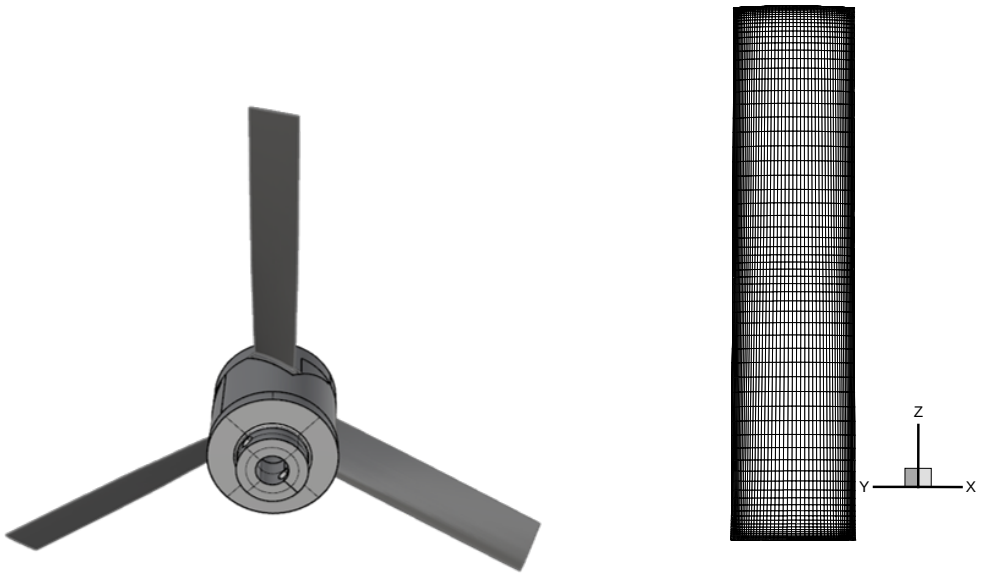

FIGURE 6.1: CAD of the fan (left); mesh of the blade (right).

TABLE 6.1: Blade details.

height [m]	0.15
hub radius [m]	0.04
shroud radius [m]	0.2
twist angle [deg]	11
chord [m]	0.035
thickness [m]	0.0015

With the support of the ISMN laboratory at CNR in Montelibretti (RM), are presented the pictures in Fig. 6.2 of a cross section of the PLA blade obtained with a SEM microscope. The section was obtained by submerging the blade in liquid nitrogen and consequently exposing the material with a brittle fracture at midspan, crosswise, and spanwise at midchord. From the observation of the section it is safe to assume that most of the blade is composed by completely fused materials, and that the anisotropic deposition of the PLA during the 3D print process is affecting the structural dynamic to a minor extent, so that on a first approximation it is possible to model the material as an homogeneous material, while being conservative with respect to the bending stress,since the modeled material leads to a slightly stiffer solid structure.

Computational domain and boundary conditions A periodic domain is set for the simulation, defined by the union of the solid domain and the fluid domain, the latter corresponding to a third of the entire fan, delimited by the hub and shroud surfaces, by the inflow and outflow surfaces and the master and slave periodic surfaces.

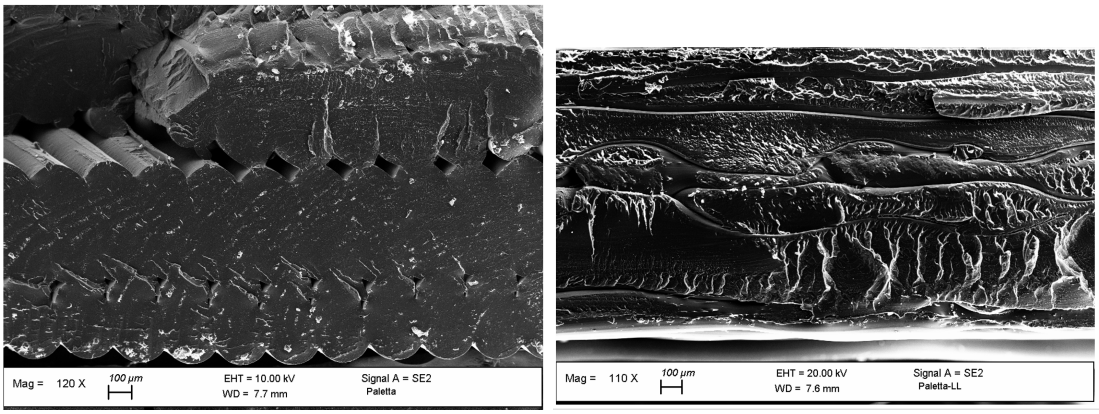

FIGURE 6.2: Cross section view of the 3D printed fan blade obtained with the SEM microscope at ISMN-CNR

The fluid domain is extended downwind from the blade center by a length equal to four times the rotor radius, and upwind by one rotor radius. The resulting mesh (Fig. 6.1) counts about 1.1M nodes: further details about the computational domain can be found on 6.8.

A no-slip boundary condition is imposed at walls, therefore the first cell at the blade surface is set in order to have $y^+ = 1$. Indeed, considering a dynamic viscosity of $\mu = 18.6 \times 10^{-6}\ Pa\ s$, the fan is experiencing a maximum Reynolds number of $Re = 5 \times 10^5$ at the blade tip. The rotor is supposed to have a constant rotational velocity of $\omega = 125,66\ rad/s$. Both systems mare solved in the frame of reference relative to the rotor (non-inertial frame). Hence, the centrifugal and Coriolis are treated as body forces for the fluid and solid systems, while both inlet and outlet surfaces boundary conditions are set with the relative velocity:

$$\boldsymbol{u}_{rel} = \boldsymbol{\omega} \times \boldsymbol{x}|_{inlet} + V_{inlet}\boldsymbol{y} \tag{6.1}$$

where V_{inlet} is the absolute free-stream velocity.

The sides of the fluid domain are considered master and slave surfaces where a rotational periodic condition is set, allowing to perform calculation on a single blade.

The blade is constrained to the hub with a cantilever constraint.

As first approximation the PLA will be considered as a linear and isotropic material, with Young's modulus $E = 3500\ MPa$, shear modulus $G = 1287\ MPa$ and density $\rho_2 = 1252\ kg/m^3$ as proposed by [109].

Test set-up The study aims to observe the fluid-structure interaction of a fan blade in different working conditions, assuming as parameter the inlet flow rate. The fluid structure interaction is investigated in two operating points defined by $Q =$

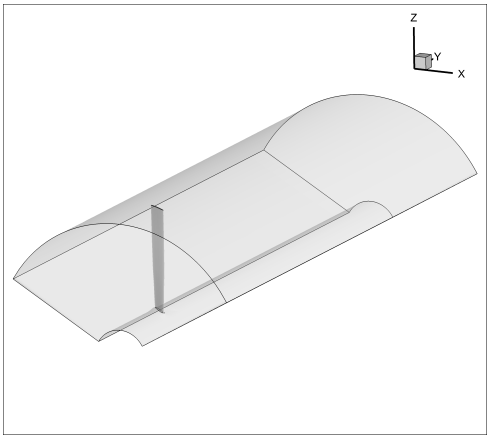

FIGURE 6.3: Computational domain for the 3D printed fan.

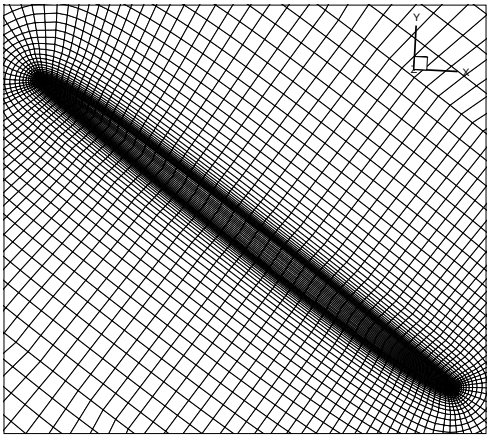

FIGURE 6.4: Detail of the grid at blade's tip.

$\{0.325m^3/s, 0.975m^3/s\}$, the first being an unstable operating point, while the second should correspond to a stable flow condition.

A time step of 5×10^{-4} s is set for both simulations; inside the time loop, 10 non linear iterations are used for the outer nonlinear loop. At the end of the time step, a new time iteration is performed until the blade's deflection magnitude stabilizes on a medium value.

TABLE 6.2: 3D printed fan computational domain characteristics.

Element type	HEX8
Total number of nodes	1154914
Total number of elements	1118610
Fluid elements	1055682
Solid elements	62928
Interface nodes	9948
Average y^+ at wall	1

Frame of reference The frame of reference used to define the solution domain is an orthogonal right-handed set of unit vectors, defined as follows:

- O (origin): hub center, below airfoil center of gravity
- Z (blade axis): vector parallel to the blade axis, directed from the hub to the tip.
- Y (rotation/thrust axis): vector lying on a plane normal to the rotor disc plane, parallel to the rotation axis of the fan and directed opposite to the fan angular velocity vector.
- X (in-plane axis): vector lying on the rotor disc plane, defined to close the orthogonal right-hand system.

Results

In the following sections the simulation with $Q = Q1 = 0.325\ m^3/s$ will be referred as "TEST 1" and with "TEST 2" the simulation with $Q = Q2 = 0.975\ m^3/s$. For sake of clarity, the results will be presented by separating aerodynamic results and structural dynamics results.

Aerodynamic results The resulting aerodynamic field is shown in Fig. 6.5, in the form of iso-surfaces of the magnitude of the cross product between velocity and vorticity, taken at the final time step of the simulation. Iso-surfaces allow to instantly visualize the shape of the fluid field and exploit the critical aspects. It is possible to notice that for "TEST 1" (Fig. 6.5, top) the blade works in a condition of separated flow around the blade, and the imposed flow condition must be considered as an off-design point, as expected. In opposition to that, it is immediate to observe that "TEST 2" inlet condition leads to a much more stable flow, with lower vorticity production and a flow attached to the most part of the blade (Fig. 6.5, bottom). The imposed flow condition can be considered as an expected design point, in spite of the leading edge separation in the near-tip region, where the vorticity structures production is relevant and affects the flow field in all the tip region.

A brief view of the time history of the integral axial force on the blade can be found as a plot in Fig. 6.6. For "TEST 1", the separated flow on the suction side leads to an unsteady and chaotic variation of the force, with an average value higher in magnitude with respect to the "TEST 2". In "TEST 2", the attached flow produces a settled oscillating force on the blade. The oscillation derives only from the FSI coupling between the two dynamics. This effect will be much more evident on the displacement field, as presented in the next section.

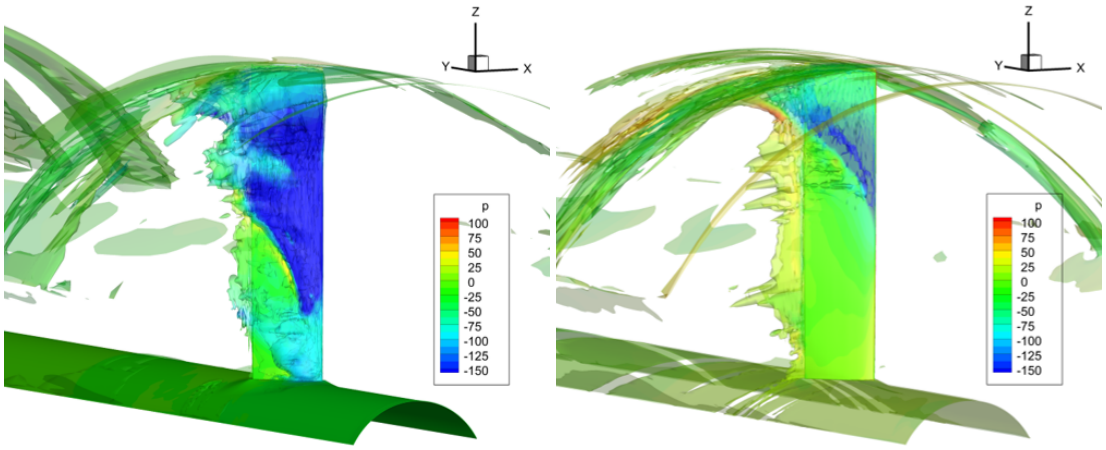

FIGURE 6.5: Velocity cross vorticity magnitude iso-surface: top, $Q = 0.325\ m^3/s$; bottom, $Q = 0.975\ m^3/s$.

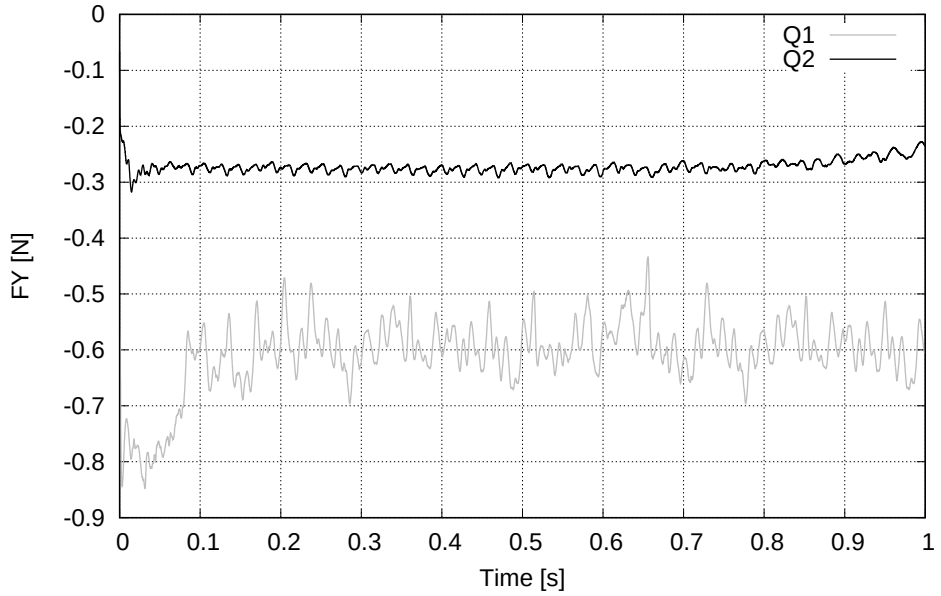

FIGURE 6.6: Time history of the integral axial force (Y-axis) on the blade.

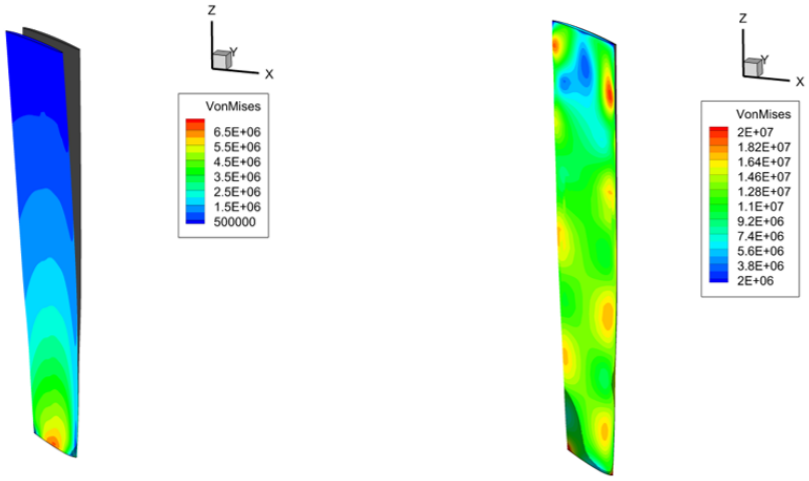

FIGURE 6.7: Maximum Von Mises stress on the blade [Pa] and deformed shape. From top to bottom: $Q = \{0.325\ m^3/s, 0.975\ m^3/s\}$.

Structural results The maximum structural displacement and its corresponding Von Mises stress distribution on the blade can be observed in Fig. 6.7 for both operating conditions, while in Fig. 6.8 is reported the plot of the displacement over time of the l.e. and t.e. tip nodes in the Y direction (V_{inlet} direction), for both tests.

In "TEST 1", a relevant blade deflection is observed in terms of maximum (Fig. 6.7) structural displacement: this behaviour is imputable to the high pressure difference between pressure side and suction side due to the separated flow, and the most significant force on the blade is the drag force. Although a chaotic and unstable flow with high vorticity production is developing, only a low frequency flapping motion is observable, showing that such flow instabilities have not excited any elastic mode of the structure, and the response is a simple bending oscillation at the first flapping frequency due to the average force. The structural stress is concentrated at the root of the blade, where the cantilever constraint is placed.

In "TEST 2" the flapping bending is initially smaller with respect to "TEST 1" in terms of average value (Fig. 6.8), while a high frequency torsional dynamics is evident and divergent. The observation of the evolution in time of this unstable torsional dynamic can suggest the beginning of a flutter at this flow condition. The Von Mises stress distribution drastically change from the "TEST 1" condition: a lower pressure difference is affecting the blade's sides, and the highest values of stress grows, due to the positive aeroelastic feedback up to values that are probably destructive for the blade structure. The high value of stress presented in Fig. 6.7 is possible as the model does not account for fracture, fatigue or yield (the latter is not present for this material) and the maximum value reached is just related to the time step at which the simulation has been interrupted.

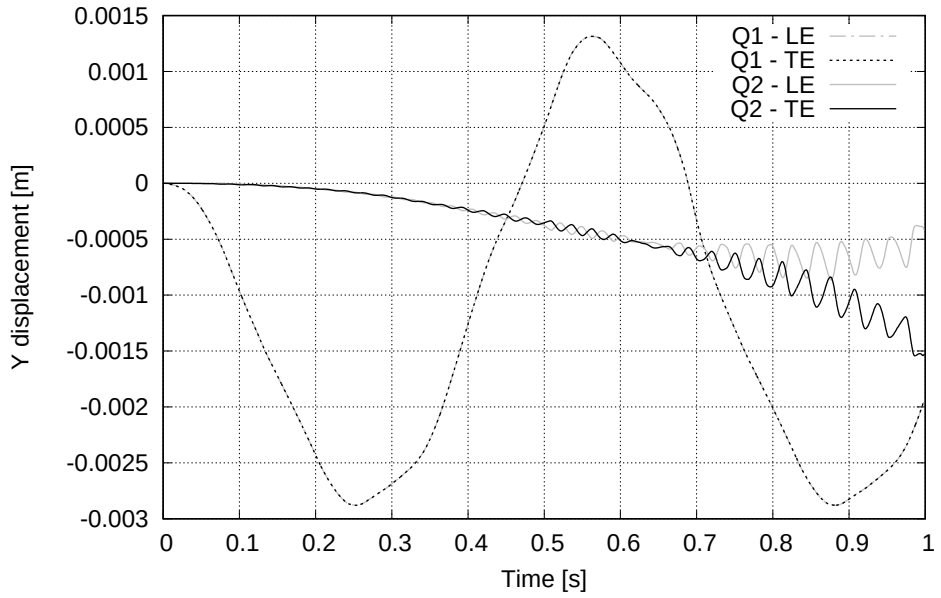

FIGURE 6.8: Time history of tip displacement in Y direction ($Q1 = 0.325\ m^3/s, Q2 = 0.975\ m^3/s$).

Final remarks

In this section is presented a computational coupled simulation for the fluid-structure interaction of 3D printed PLA fan blade in two different operating conditions: a install off-design point, and a design operating point, differing for the imposed inlet flow rate condition. The choice has been made to observe the aeroelastic response in two opposite situations of regular and and completely separated flow. The numerical solution gives a quantification of the characteristics of the structural and fluid dynamics over time (bending displacement, stresses, pressure field, vorticity) of the interaction. It's highlighted how in this case an unstable stalled flow field would produce a average high load on the blade, but also a stable aero-structural response. While, despite a lower average value of the aerodynamic load, a in-design point can lead to a positive feedback between the aerodynamic and structure dynamics, which can bring the blade to destructive unstable vibration. Indeed, the emerging of a possible torsional flutter in one of the design points is highlighted in the results.

Validation and experimental measurements

Experimental measurements on the aerodynamic performance of the printed blade has been carried out in the laboratory of the University of Rome La Sapienza, at the office in Latina (LT). The reduced size of the blade, the low solidity and blade number used for this sample case, allowing to reply, at this state of the study, only the working condition for TEST 1 ($Q = Q1 = 0.325\ m^3/s$). Table 6.3 reports the comparison between

experimental and simulation results in terms of static pressure jump and power at the shaft.

	Pressure difference $[Pa]$	Power $[W]$
FSI simulation	2.65	20.49
Experiment	3.23	23.32

TABLE 6.3: Data comparison at $Q = Q1 = 0.325\ m^3/s$

The FSI solver algorithm has been performed in [110] by running the benchmark proposed by [111].

6.1.2 Unsteady flow simulation of an axial fan for dry cooling in a CSP plant using the Variational Multiscale Method

In the following section is presented an unsteady simulation of an axial fan with low pressure and high flow rate, developed during the MinWaterCSP EU project, specifically designed and optimized for Concentrating Solar Power (CSP) plants with the aim of reducing water consumption at the condenser stage. An earlier version of the fan, referred as M-fan, was initially developed to serve the hybrid cooling system, to boost the overall heat transfer performance with dry cooling. A Computational Fluid Dynamics (CFD) analysis of the unsteady flow is performed on a redesigned and scaled version of the M-fan, referred as T-fan, specifically designed and optimized to reduce noise emissions and improve efficiency with respect to the M-fan. Finally, a brief discussion on the simulation results of the RBVMS model will be conducted by comparing the simulation results for the T-fan with experimental data gathered on the same fan.

Large axial fans in CSP plants

Concentrating solar power (CSP) plants are located in regions of high solar resource. A region with high solar resource often has a limited or constrained supply of water. The use of dry-cooling for Rankine cycle CSP plants is therefore preferred, due to their sustainable rate of water consumption. Axial flow cooling fans are used in mechanical draft air-cooled condensers (ACCs) to induce airflow over the condenser fin tube bundles. The operation of the cooling fans inside an ACC are affected by the presence of high ambient temperatures and wind, both phenomena that occur frequently in regions of high solar resource. The performance of the ACC, and the power plant, is therefore closely related to the performance of the cooling fans [112]. The Horizon2020 MinWaterCSP project was defined with the purpose of reducing the water consumption of CSP power plants. For CSP plants making use of wet cooling a substantial portion

of water is consumed by the condenser. As explained previously, dry-cooling would therefore be preferred, when considering the water consumption of a plant. However, wet-cooling power plants have a performance advantage over dry-cooled plants due to their ability to run at a lower steam vacuum pressure. To compensate for this disadvantage in performance, the MinWaterCSP project proposed the use of a novel hybrid-condenser, along with a purpose-designed cooling fan to reduce the cooling water consumption of a CSP plant, while simultaneously improving its performance [113]. Wilkinson [114] designed the M-fan for use in an ACC, within the requirements of the MinWaterCSP project. The fan was designed with a diameter of 24 ft., in order to suit the full scale ACC test facility at Stellenbosch University. The design was however required to conform to numerous performance and manufacturing constraints. Most importantly, it had to exhibit a design fan static pressure rise of only 117 Pa, in order to match the modern trend of high flow rate, low pressure rise ACC fin tube bundles. The M-fan was tested experimentally in 1.542 m diameter scale on an ISO 5801, Type A test facility at Stellenbosch University. The M-fan exhibited a fan static efficiency of 59 % at a tip clearance of 0.002 m on the scale test facility [115]. As part of the MinWaterCSP project, the noise characteristic of the cooling fan also had to be considered. Due to time constraints the design of a reduced noise fan could only be performed following the completion of the M-fan design. Consequently it was decided to make use of the opportunity to remove some of the constraints applied to the design of the M-fan, in order to see if the efficiency of the fan could be improved, while simultaneously reducing its noise emissions. This led to the development of the so-called T-fan, which will be described in the next section.

The simulation of the noise emission requires an accurate prediction of the unsteady flow field to catch the field pressure fluctuations and load, which is the main cause of the noise. The objective of this specific research is to evaluate the possibility to simulate the unsteady flow field with a high resolution stabilized finite element formulation of the 3D unsteady flow field equations, as a first part of an ongoing study which will lead to an evaluation of the noise production of the fan in further studies. The accuracy of the simulation will be verified by comparison with the experimental data.

Blade description

As mentioned above, the fan object of this study, referred as the T-fan, is the result of a redesign of the 24ft axial fan developed during the MinWaterCSP project [113], in order to reduce the noise and improve efficiency. The blade profile chosen for the T-fan is a NASA LS 413, with a modified camber angle distribution, and 8 blades are mounted on the fan: a picture of the scale model of T-fan can be found in Fig. 6.9.

FIGURE 6.9: Scaled model of the T-fan, 3D printed, mounted on the test facility at Stellenbosch University.

While the M-fan was designed according to an hub-to-tip ratio optimization from van Niekerk [116] and a minimization of the kinetic energy at the fan exit through the velocity distribution [117], the T-fan was redesigned by Bonanni et al. using a Hybrid Design methodology, a methodology which is not dependent from the hub-to-tip ratio of the fan [118]. This methodology relies on the coupling between the vortex distribution and the aerodynamic response of a blade profile, ensuring a direct connection between the profile performance and the load distribution. The aim of this new design was to reduce the load at the near-hub region of the blade to avoid high angle of attack and flow recirculations, especially at low flow rates, reducing noise emission and improving the flow stability. Volponi and Bonanni [118] chose the power law vortex distribution to obtain this specification. As a consequence of the new Euler Work distribution imposed on the blade span, between 7% and 40% of the blade height, the T-fan is more loaded than the M-fan presenting a higher pitch, while globally the T-fan is less twisted than the M-fan. Previous comparisons between the M-fan and the T-fan quantified an improvement in efficiency by approximately 2%[119]. In Figg. 6.10,6.11,6.12 are displayed the comparison between the M-fan and the T-fan in terms of diffusion factor DF [120], normalized lift and drag coefficient C_L^*, C_D^* using Min-Max normalization over the normalized radius $r^* = r/r_{tip}$. The particular configuration of the T-fan we present in this work is the most promising configuration of the T-fan in terms of efficiency, according with the results obtained during the MinWaterCSP project, with a $0.002m$ tip clearance, corresponding to 0.13% of the outer diameter and 1.17% of the blade chord, and a 31° setting blade angle. In Fig. 6.13 it is indicated the comparison of the spanwise distribution of pitch over the normalized radius for both the T-fan and the M-fan, where the pitch angle is the angle between the chord and the rotation plane. Further details on the specifications of the T-fan can be found in Tab. 6.4.

blade number $[-]$	8
blade height $[m]$	0.549
hub radius $[m]$	0.22
shroud radius $[m]$	0.771
hub to tip ratio $[-]$	0.29
aspect ratio $[-]$	2.23
chord (avg) $[m]$	0.17
tip gap $[m]$	0.002
pitch angle (hub) $[deg]$	31
rotational speed $[rpm]$	722.11
blade tip speed $[m/s]$	58
fan static pressure rise $[Pa]$	116.7

TABLE 6.4: T-fan design specifications

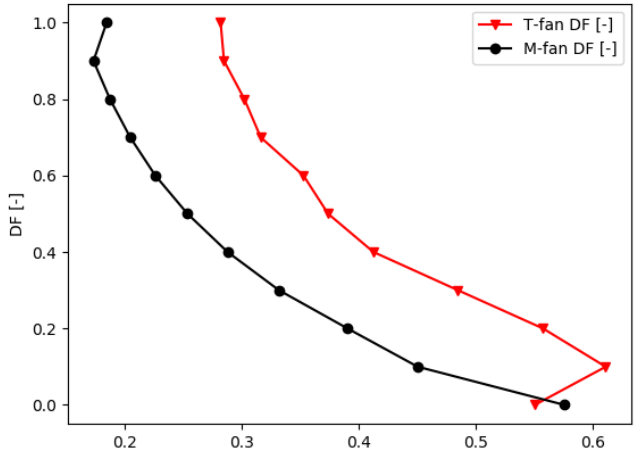

FIGURE 6.10: Diffusion Factor over the normalized radius for M-fan and T-fan.

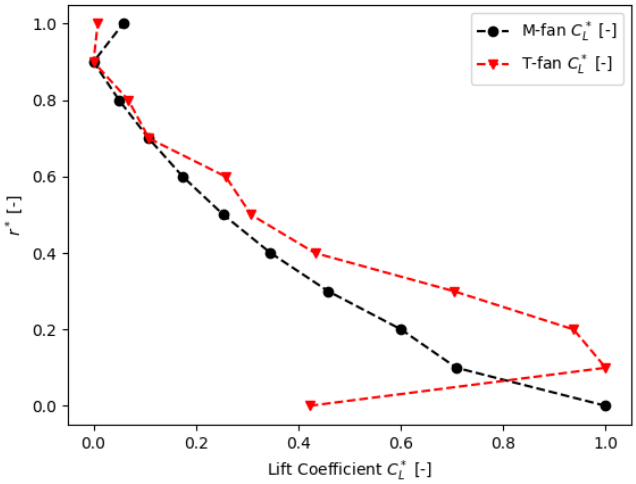

FIGURE 6.11: Normalized lift coefficient over the normalized radius for M-fan and T-fan.

Experimental Measurements

The experimental performance of the T-fan was measured on the ISO 5801, Type A (open-inlet and open-outlet) fan test facility at Stellenbosch University, shown in Fig. 6.14. The test facility makes use of a calibrated bell mouth at its inlet. The bell mouth is used to determine the flow rate of the air passing through the facility $\dot{V}$ by measuring the static pressure in the throat of the bell mouth, relative to the surrounding atmosphere. A set of louvres, situated downstream of the bell mouth, is used to control the flow rate through the facility. An axial flow auxiliary fan is used to overcome the losses of the test facility. Flow straighteners, guide vanes and mesh screens are used to

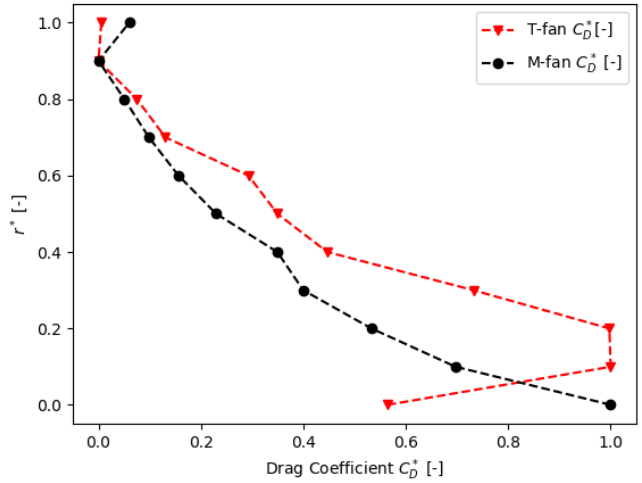

FIGURE 6.12: Normalized drag coefficient over the normalized radius for M-fan and T-fan.

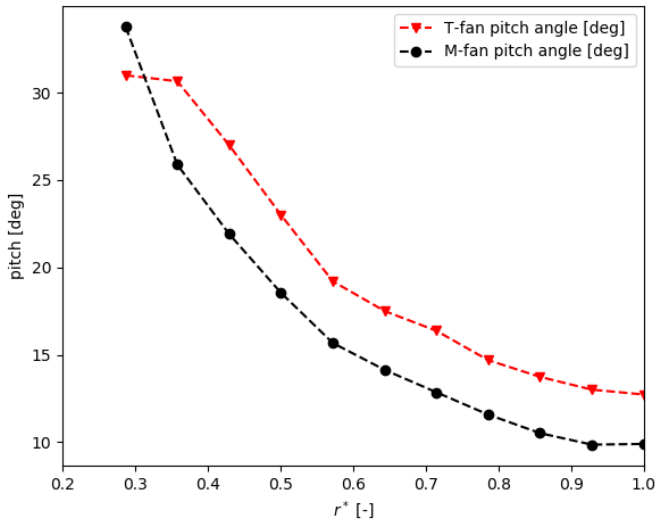

FIGURE 6.13: Distribution of pitch angle over the normalized radius for T-fan and M-fan.

condition the flow that enters the settling chamber upstream of the experimental fan. The settling chamber conditions the flow in order to ensure an even, low flow velocity profile upstream of the experimental fan. The fan static pressure measured at the side of the settling chamber, along with the average velocity upstream of the fan is used to calculate the fan static pressure rise:

$$\Delta p_{FS} = -(\Delta p_{plenum} + p_{dynplen}) \tag{6.2}$$

The experimental fan is located inside a shroud with an inside diameter of 1,542 m. The fan is driven from outside the facility. The fan shaft is equipped with a torque

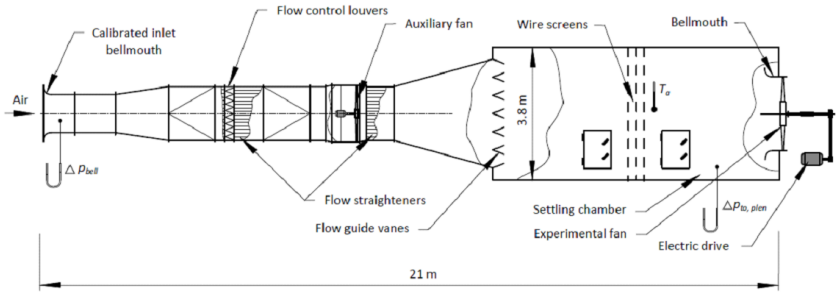

FIGURE 6.14: Layout of the ISO 5801, Type A fan test facility at Stellenbosch University.

transducer and proximity switch that are used to record the shaft speed and torque. These values are used to calculate the fan shaft power (PPF) consumption. The static pressure values in the inlet bell mouth and settling chamber are measured using two HBM PD1 pressure transducers. The shaft torque is measured using a HBM T22 torque transducer, while the rotational speed is measured by a proximity sensor mounted next to the shaft. All the measuring devices are connected to a HBM PMX data acquisitioning system. The PM is connect to a personal computer that uses CATMAN software for the data recording and processing. For the experimental evaluation of the T-fan, the blade stagger angles were measured using a digital protractor. Similarly, the fan tip gap was measured using a Vernier caliper. Zero readings were recorded at the start and end of each test set. The fan running speed was set at 722 rpm. The experimental data was recorded for 10 seconds at a sampling rate of 100 Hz. Each experiment was repeated three times. The results for the experimental evaluation of the T-fan are shown in Figures 6.15, 6.16 and 6.17. The results are shown for a tip clearance of 2 mm and for blade angles ranging from 29° to 32°, measured at the blade hub. As mentioned before the results display the average values for 3 tests per fan blade setting angle. The average standard deviation in fan static pressure and fan static efficiency for all the measured data points was calculated as 1.6 Pa and 1.4 % (absolute) respectively.

%subsubsectionFlow simulation

Computational domain and boundary conditions The computational fluid domain is designed as a vane of the rotor, a 45° wide cyclic section of the fan, corresponding to 1/8 of the entire machine, and delimited by the hub and shroud surfaces along the radial direction, by the inlet and outlet surfaces along the absolute velocity direction, and by the cyclic surfaces at the sides.

The resulting HEX8 mesh (Figg. 6.18, 6.19) counts about 6.7M nodes: further details about the computational domain can be found in Tab. 6.8.

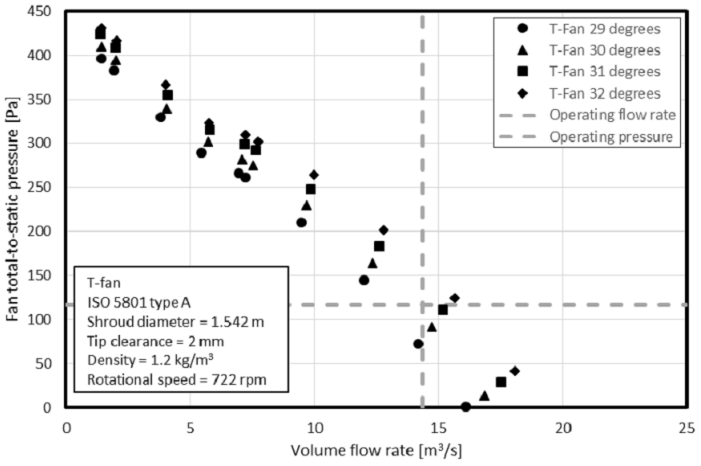

FIGURE 6.15: Fan static pressure vs. volume flow rate for T-fan.

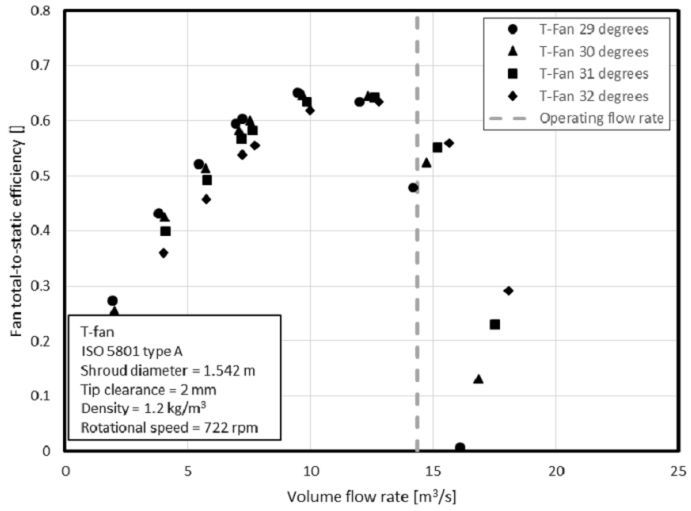

FIGURE 6.16: Fan static efficiency vs. volume flow rate for T-fan.

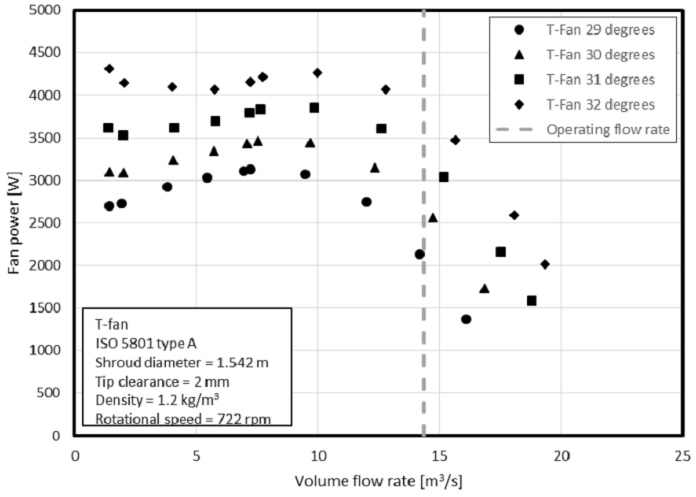

FIGURE 6.17: Fan shaft power vs. volume flow rate for T-fan.

Element type	HEX8
Total number of nodes	6749802
Total number of elements	6625884
Average y^+ at wall	1

TABLE 6.5: Computational domain characteristics.

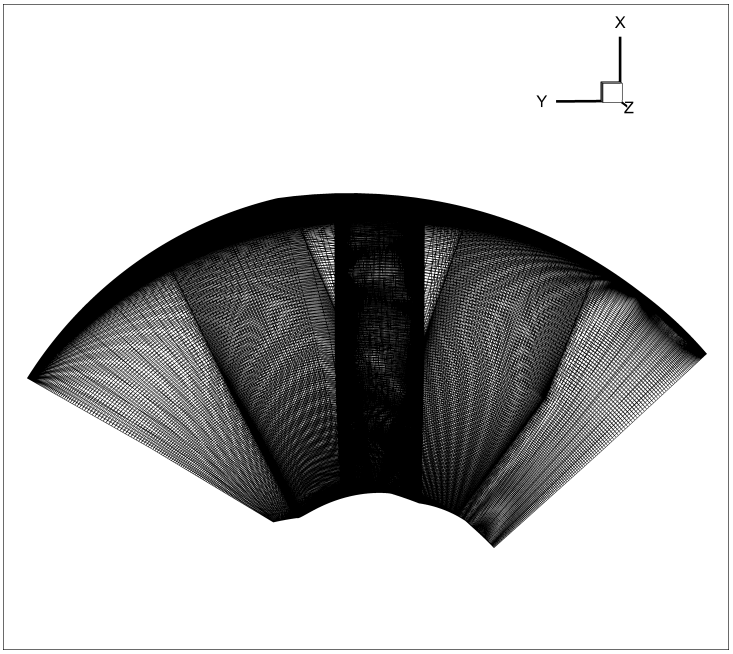

FIGURE 6.18: Computational domain for the CSP axial fan 3D flow simulation.

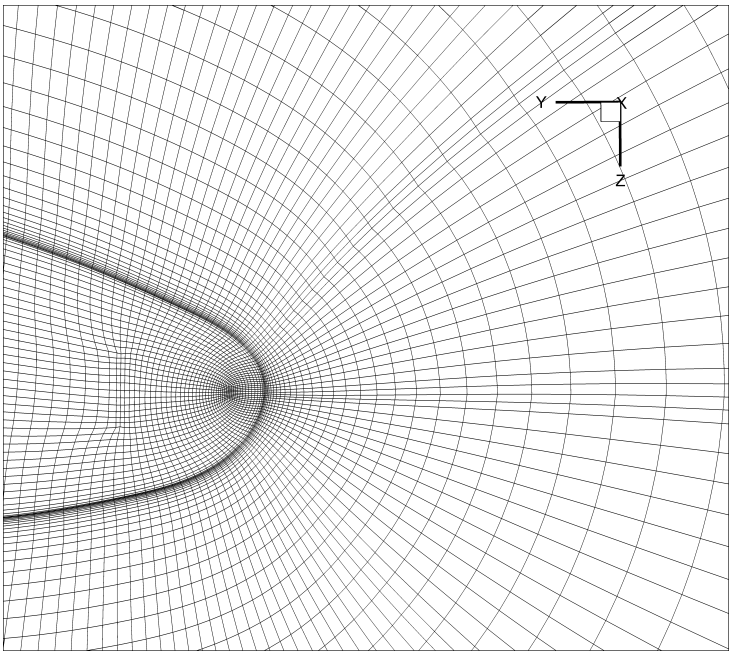

FIGURE 6.19: Detail of the grid at tip clearance, leading edge.

At the walls (i.e. the blade's surface, case and hub) the first cell presents $y^+ \simeq 1$, so it is possible to set a no-slip boundary condition, while the kinematic viscosity value is set as $\mu = 16 \times 10^{-6}\ Pa\ s$. The rotor is supposed to have a constant rotational velocity of $\omega = 722\ rpm$, while inlet boundary conditions is set with the relative velocity:

$$\boldsymbol{u}_{rel} = \boldsymbol{\omega} \times \boldsymbol{x}|_{inlet} + V_{inlet}\boldsymbol{y} \tag{6.3}$$

where $V_{inlet} = 10m/s$ is the absolute free-stream velocity associated to the selected flow rate. The simulation is carried on for 8×10^4 time steps, each one using $dt = 1 \times 10^{-6}s$, to observe a full revolution of the fan. The total simulation time is corresponding to a full rotor revolution to assure the complete blow out from the domain of the CFD start-up dynamics.

Frame of reference The frame of reference used to define the solution domain is an orthogonal right-handed set of unit vectors, defined as follows:

- O (origin): hub center, below airfoil center of gravity
- X (blade axis): vector parallel to the blade axis, directed from the hub to the tip.
- Z (rotation/thrust axis): vector lying on a plane normal to the rotor disc plane, parallel to the rotation axis of the fan and directed opposite to the fan angular velocity vector.
- Y (in-plane axis): vector lying on the rotor disc plane, defined to close the orthogonal right-hand system.

Aerodynamic performance analysis

In this section is presented a comparison between the simulation results with the experimental results in terms of fan efficiency, fan power and pressure rise. A good correlation between the simulation results and the experimental data is therefore reported, proving the suitability of both the RBVMS model for this specific case, even with the limitations of a validation using only one design point. Fig. 6.20 shows the total-to-static pressure rise obtained with the simulation as $\Delta p_{VMS} = 125\ Pa$, underestimating the experimental value of $\Delta p_{EXP} = 132\ Pa$ by 5%. In Fig. 6.21, the estimated absorbed power $P_{VMS} = 3176\ W$, obtained with the calculation of the resultant moment on the blade along the rotation axis, is compared with the experimental measurement of absorbed power at the shaft. From the previous values is evaluated a total-to-static efficiency $\eta_{VMS} = 0.56$, while the experimental results suggests an efficiency of $\eta_{EXP} = 0.58$ (Fig. 6.22).

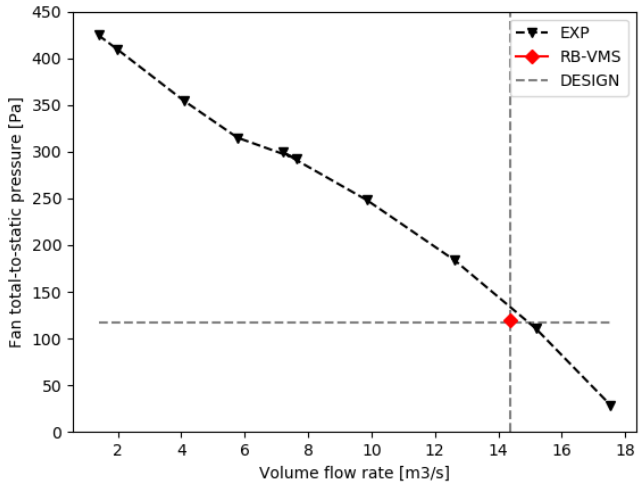

FIGURE 6.20: CFD evaluated pressure rise compared with experimental results.

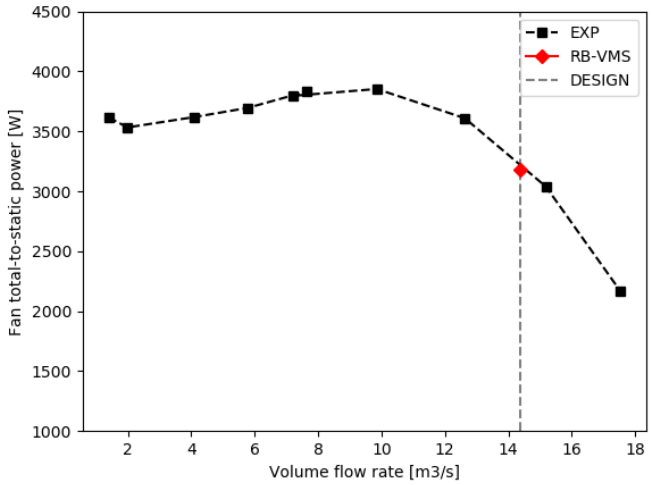

FIGURE 6.21: CFD evaluated absorbed power compared with experimental results.

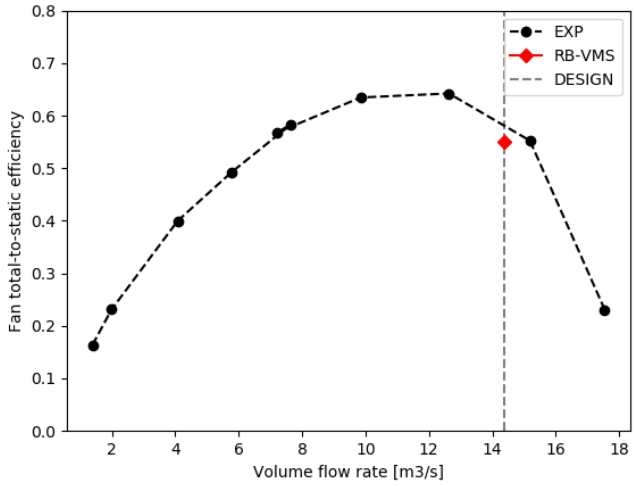

FIGURE 6.22: CFD evaluated efficiency compared with experimental results.

Flow survey

In this section are displayed the main characteristics of the simulated flow, focusing the attention on the behaviour of the flow at the tip clearance, where critical losses can occur in the form of tip leakage vortices. Indeed, one of the main issues addressed during the development of both the M-fan and the T-fan for the MinWaterCSP project was the analysis of the tip leakages. In Fig.6.23 it is possible to observe the presence of tip and hub turbulent structures behind the single blade, in terms of contours and iso-surfaces of the second invariant of the velocity tensor $Q = \frac{1}{2}(u_{i,i}u_{j,j} - u_{i,j}u_{j,i})$. In the figure are also displayed two planar sections of Q contours at a distance of $d_1 = 0.3 \cdot c$ and $d_2 = 1.5 \cdot c$ from the trailing edge, to enrich the view of the iso-surfaces of Q originating from the blade. When observing the flow at the tip clearance region, it is useful to initially analyze the pressure field, as reported in Fig. 6.24. By the observation of the pressure distribution, stands out a low pressure zone over the leading edge, on the pressure side, and next to the trailing edge at the suction side. This distribution suggests the suction of the flow in the proximity of the leading edge, and a flow separation at the trailing edge. According to the large number of analysis conducted on tip leakage flows in axial compressor and ducted fans [121], [122], it is observable in Fig. 6.25 and Fig. 6.26 the tip leakage vortex, drawn using an iso-surface of the pressure field and the streamlines of the relative velocity. A large vortex grows when the relative flow crosses the tip clearance from the pressure side to the suction side of the blade due to the large pressure difference, and it grows along the suction side. A small secondary vortex generates at the leading edge and covers the entire chord length traveling across the tip surface and mixing with the larger vortex downstream. Fig. 6.25 shows the presence of pressure fluctuations (intended as non continuous pressure field in a spatial sense)

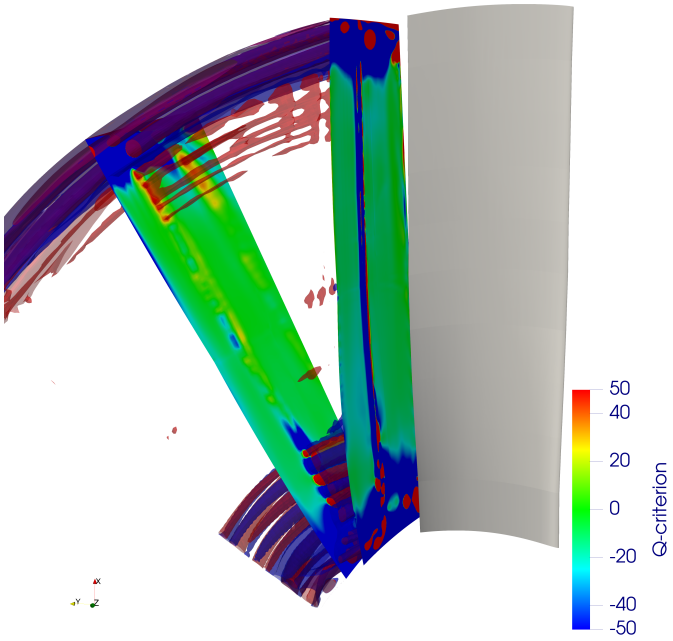

FIGURE 6.23: Iso-surfaces of Q-Criterion describing tip and hub vortices for the T-fan.

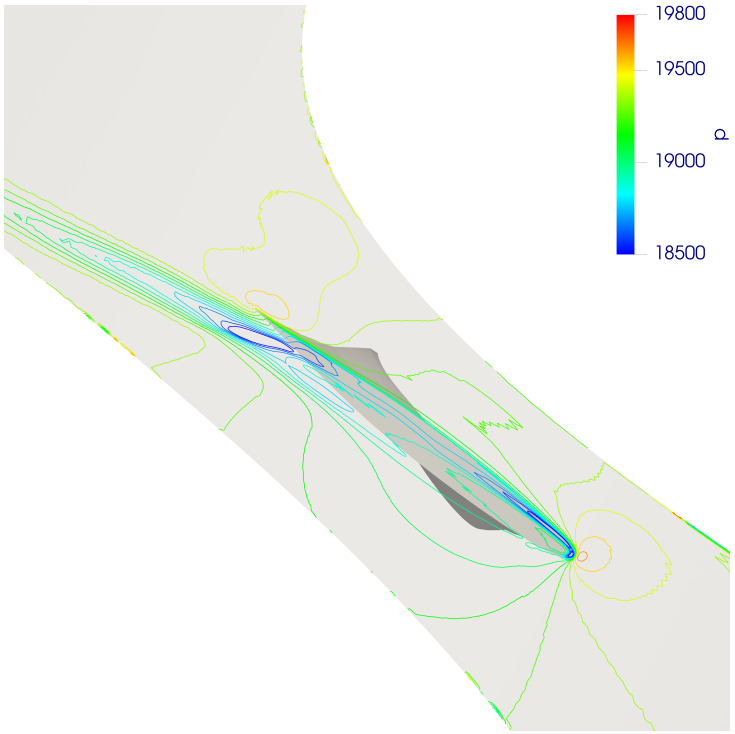

FIGURE 6.24: Iso-lines of pressure at a section in the tip clearance.

on the blade suction side surface, in the tip-span region of the fan blade, which can be related to the presence of turbulent structures.

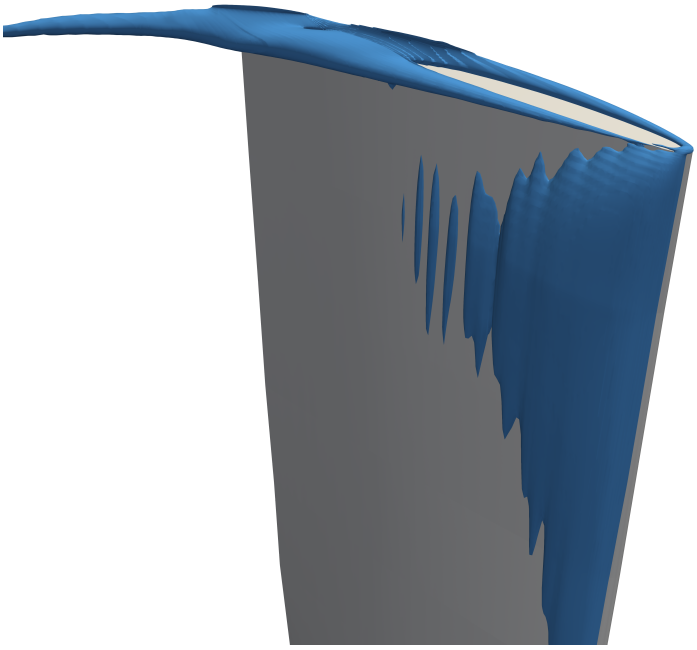

FIGURE 6.25: Iso-surface of pressure at the blade tip.

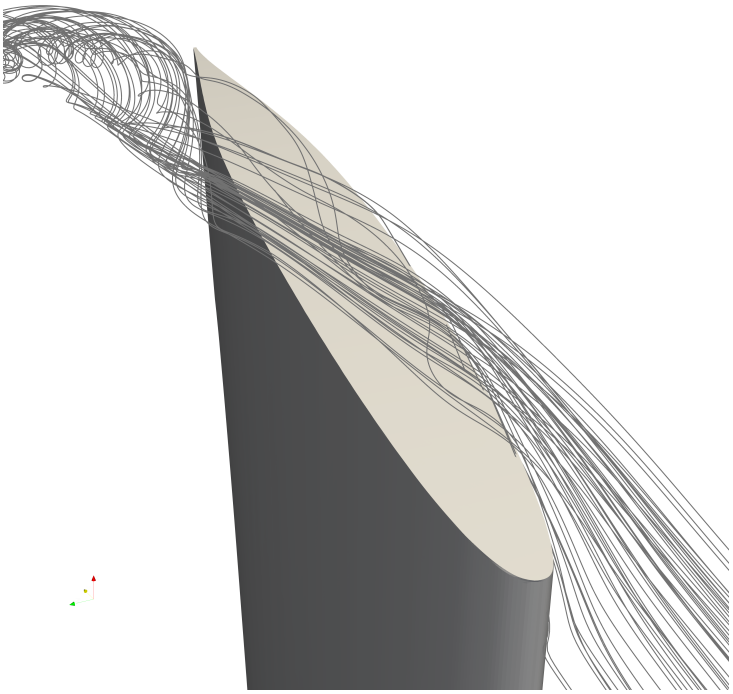

FIGURE 6.26: Streamlines of the relative velocity describing the tip leakage vortex.

Final remarks

In this section has been characterized an axial fan designed to serve at the ACC stage of a CSP plant, by performing both experimental measurements and a 3D unsteady

flow simulation. The fan geometry presented in this study, referred as T-fan, was designed using a hybrid method that relates the vortex distribution and the aerodynamic response of a blade profile. Experimental measurements have been performed for a scaled version of the original fan, at different blade's angle and different tip clearances, as well as an unsteady flow simulation of the most efficient set-up of the T-fan ($\eta = 0.58$ at the design point) when the blades are mounted 31°, with a 2 mm tip clearance. For this specific fan, the finite element formulation of the RBVMS model has been used to simulate the fan blade aerodynamics. The simulation results have been compared with the experimental measurements obtaining a good agreement in terms of integral quantities. A flow survey has been conducted to analyse in details the tip leakage flow and the pressure field. It is showed that pressure fluctuation are expected corresponding with the wake and tip vortex, but also on the blade suction side surface, in the tip-span region. The presence of the aforementioned pressure fluctuations are worth further investigations in terms of noise analysis and fluid structure interaction analysis, as they could suggest the insurgence of vibrations. The results of the unsteady flow simulation are promising with respect to the reliability and applicability of the RBVMS method in fans, and the agreement with the experimental results encourage further evaluation of the performances of the T-fan.

6.2 Morphing in turbomachinery

6.2.1 Morphing of reversible axial fan blade: a FSI-FEM study

Reversible axial fans are widely used in industrial and tunnel ventilation systems, and a lot of research effort is spent in the design process of the blades shape and blades profile. The target is to achieve reasonable performances in both flow directions, but those are still below the levels of the corresponding non-reversible geometries. In this work, an alternative design solution for reversible axial fan is presented by adopting flexible blades instead of the rigid ones. Such design, inspired by the boat sails, could allow the blade to change its shape by passively adapting to the flow field, from a symmetrical blade profile to a not symmetric one, and thus adapting the curvature to the flow condition. In the analysis, a series of alternative materials and material distributions are analysed and compared.

Reversible axial fans design

Reversible fans are a well established field of research, and are widely used for air supply and air extraction. In some applications, where fans can be large and expensive,

it is often convenient to use one reversible fan in place of two different standard fans[123, 124, 125].

Reversible fan blades are often designed starting from a non-cambered airfoil by inverting, at the pressure side, the leading edge with the trailing edge, and placing the inverted semi-profile in place of the suction side, effectively joining the leading edge at the pressure side with the former trailing edge at the suction side. Such design method leads to a reversible profile characterized by an unnecessary thickness at the trailing edge, due to the symmetry requirements to achieve the true reversibility. This choice results in a loss in performance, in terms of average aerodynamic efficiencies, of at least 5% with respect to the non reversible original airfoil [126]. An alternative design method for reversible fan blades was proposed in [46, 47], where the airfoil design relies on machine learning and artificial intelligence, by setting up surrogate models using Evolutionary Algorithms or Artificial Neural Networks. When looking at the expected characteristics of a reversible fan, it is possible to imagine that the desired geometry could be obtained not only through geometrical and topological optimization of the blade, but also through a blade morphing mechanism. The possibility to achieve this goal through a passive elastic morphing is the topic of the present study.

A symmetric, sufficiently thin and sufficiently flexible blade could possibly adapt itself to the flow and consequently be able to obtain adequate performances in both rotational directions. Similar solutions have already be proposed for Wells turbines, where the periodic change in the flow direction can deform the geometry of a flexible blade, both for a trailing edge flexible appendix [127] and for a full flexible blade section [30].

As observed in [30, 29], elastic morphing devices can be designed firstly considering an average static load, but then they should be checked in terms of fluid-structure dynamics feedback. This task can be aided by numerical simulation, implementing the proper ability to describe the Fluid-Structure Interaction (FSI) with a strong coupling algorithm [33].

Fan description

Reference fan The aim of this work is to study the behaviour of a morphing blade section. Therefore, the complete design procedure of a new axial flow fan is avoided, assuming to mount a reversible and flexible blade into an existing fan design. The main features of the machine are taken from an existing design from the literature. The fan developed by Van der Spuy in [128] has been taken as reference for the definition of the fan characteristics. The chosen fan has a high aspect ratio, and operates for low pressure rise and high flow rate condition. From the reference fan, the dimensions of the hub, the number of the blade, the length of the blades, the chord length and the angles were

	Value
Hub radius [m]	0.125
Shroud [m]	0.5
Chord length (avg) [m]	0.0743
Blade lenght [m]	0.375
Aspect Ratio	0.2
Angle of attack [deg]	4
Blade count	7
Solidity	3
Rotating speed [rpm]	1440

TABLE 6.6: Reference fan details from [128].

FIGURE 6.27: Profile sketch

taken, while the profile geometry has been substituted for the present analyses. A brief summary of the details of the original fan that were used in the flexible version can be found in Tab. 6.6. Further details on the blade of the reference fan and its design method can be found in [128]. It is important to underline that, having changed the aerodynamic profile of the blade, the results shown in this application are not related to the original one in terms of performance.

Reversible and flexible blade The reversible blade profile has been conceived as symmetrical flat plate rounded at the trailing and leading edges. The thickness of the plate is supposed constant and equal to 0.002 m. The blade section is constrained to two circular edges located inside the leading edge and at the trailing edge areas (Fig. 6.27). In an hypothetical 3D configuration, those circles corresponds to a 2D section of two cylindrical guides that pass all along the span of the blade, from the hub to the shroud, and that provides stiffness to keep the leading edge and trailing edge in place. Fig. 6.27 shows a qualitative sketch of the 2D section of the blade. In the figure, a red dot highlights the mid point taken as a probe for the displacements analysis, that will be discussed in the later sections.

Test cases A series of different simulations are performed to investigate the influence of different material choices and constraint configurations. In the following sections, different cases will be referred with capital letters from "A" to "D", with the special case of a reference, only aerodynamic simulation, in which the FSI coupling has been turned off, that will be referred as "FIX". Cases "A" and "B" represent configurations in which the leading edge constraint is kept fixed in its position, as a scleronomic constraint, while

	FIX	A	B	C	D
FSI	no	yes	yes	yes	yes
$E\ [MPa]$	-	50	200	50	200
ν	-	0.49	0.49	0.49	0.49
$\rho\ [Kg/m^3]$	-	2300	2300	2300	2300
constraint	-	fixed	fixed	moving	moving

TABLE 6.7: Test cases summary

cases "C" and "D" are referring to configurations in which the leading edge constraint is treated as a rheonomic constraint and it has a prescribed displacement in the direction of the chord, equal to 0.5% the length of the chord.

Such prescribed displacement is intended as an attempt to represent the deformation that would occur in a three dimensional blade constrained at the leading edge with a metallic cylindrical guide. It's possible to imagine that such material would be subjected to a small deformation in the central region of the blade due to the buckling displacement. The chosen value represent a trial assumption to include this mechanism in the 2D simulation. The new configuration could lead to a relaxation of the internal stress distribution of the blade, and to a more stable operating configuration, by relaxing the entire dynamic system.

In addition to the different constraint type, cases also differ from the material chosen to model the flexible blade: for cases "A" and "C" a Young's modulus of $E = 50\ MPa$ is considered, and cases "B" and "D" have $E = 200\ MPa$, while all material presents the same Poisson's ratio $\nu = 0.49$ and the same density $\rho = 2300\ Kg/m^3$. A summary of the properties for all the tested configurations is reported in Tab 6.7.

Numerical setup

In this work the 2D simulation of a section of the fan is performed. The computational domain is a rectangular and describes the blade vane of the fan cascade, extracted at radius $r = 0.421\ m$ (i.e. at 75% of the blade length). In the axial direction, the domain extends for one chord upstream and downstream with respect to the center of the blade. According to the operating conditions for the reference fan, described in [128], an inlet relative velocity of 65 m/s is imposed at the inlet boundary, resulting from a rotational speed of $\omega = 150\ rad/s$ (1440 rpm) and a flow rate $Q = 10.4\ m^3/s$, while at the outlet boundary a zero gradient condition is imposed. At the blade surface a no-slip boundary condition for the velocity is imposed weakly, according to the method described by Bazilevs in [33], to adaptively relax the no-slip boundary condition to a slip condition when the boundary layer is not sufficiently refined to resolve the flow. For such reason, it is still possible to carry on the simulation even with a coarse refinement

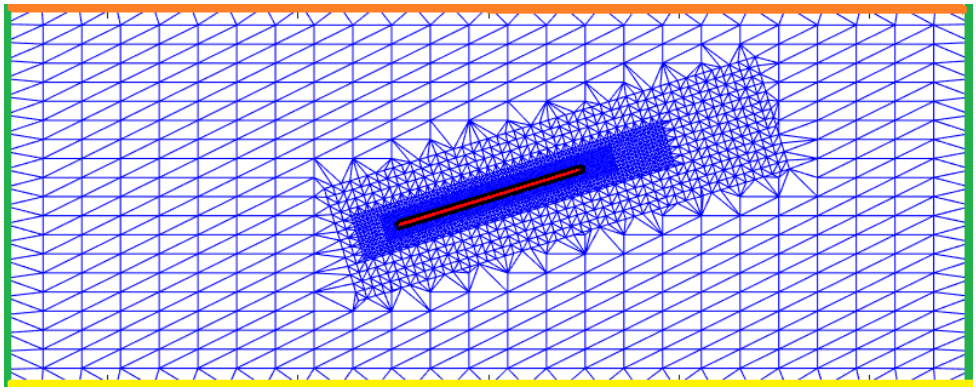

FIGURE 6.28: Domain mesh and BC.

	Value
Fluid nodes	3582
Solid nodes	1011
First layer thickness [m]	0.0003
Re	3.4×10^5
y+	60

TABLE 6.8: Mesh and BC details.

at the wall and a low number of nodes, with $y^+ = 60$ for $Re = 3.4 \times 10^5$. Fig. 6.28 shows the 2D resulting triangular mesh. In the figure are highlighted in blue the fluid domain, in red the solid domain, in yellow the inlet boundary, in black the wall boundary, in orange the outlet boundary, and in green the periodic boundaries. For the simulation a $dt = 1.16 \times 10^{-4}$ s (corresponding to an angular step of 1° for each time step) is set for 2880 time steps, effectively simulating 8 complete revolutions of the fan. The initial condition of the flow field for the FSI simulation was obtained by simulating the flow at the operating condition while keeping the structure fixed for 4 complete revolutions.

Results and discussion

All of the test cases were simulated using the same initial conditions. Results will be discussed both in terms of a dynamic analysis of the displacements of the point at the center of the pressure side, as reported in Fig. 6.27, and in terms of a flow survey, pointing out the most relevant features of the aerodynamic field.

Dynamic analysis It is useful to observe the dynamic behaviour of the different flexible blades in the frequency domain. Fig. 6.29 shows the Fast Fourier Transform (FFT) of the displacement of the central node on the pressure side, compared with the evolution of the pressure on a node on the suction side, for cases "A" and "B" in the upper subplot and for cases "C" and "D" in the lower subplot. The two subplots

compares the behaviour of the blade motion when it is coupled with the fluid dynamics. In this way, the effects of having different materials and different constraint are kept separated. The spectrum amplitude is normalized in all the plots. The normalization factor is taken as the difference between the minimum and the maximum amplitude in time of each variable, to compare phenomena that exists on different energy levels (i.e. pressure fluctuation and structure displacement).

The response in case "A" displacement (upper, black continuous line) is characterised by two main modes, the highest in amplitude directly related with the pressure dynamics (upper, black dashed line), which has a mode at the same frequency. It is possible then to consider such vibration as originated by the aeroelastic coupling. It is also possible to state that the secondary mode is only related to structure dynamics and not directly induced by the pressure field, possibly due to the low stiffness of the blade. A secondary pressure peak at higher frequency is not originating a vibration in the structure.

The response in case "B" displacement (upper, red continuous line) is presenting one principal mode at the same frequency of the main mode of the pressure dynamics (upper, red dashed line). The vibration is at higher frequency with respect to case "A", due to the higher stiffness of the structure.

Case "C" plot is showing that the blade displacement (lower, black continuous line) is still largely susceptible to the pressure field (lower, black dashed line), even when the tension of the blade is reduced due to the displacement of the leading edge constraint. This is largely related to the low stiffness of the blade, so that the blade itself is more prone to be affected by the variations of pressure load, leading to an unsteady behaviour. In case "D", the coupling is not highlighted by a specific mode, but both the displacement and pressure dynamics are closely related at all frequencies. It must be noted that, in this case, the normalization factor is much smaller than in the other cases (one order of magnitude). The blade section moves fast to a stable condition and the dynamics is only related to the small vibrations induced by local pressure fluctuations. A further explanation of the dynamic behaviour of the different proposed blade will be given with the aid of Fig. 6.30, where the average displacements of each blade profile are compared with the maximum displacements obtained during the simulation: according to Fig. 6.29, test cases "C" and "D" presents an average displacement dynamic far more stable than test cases "A" and "B".

Such a different behaviour can be further explained with the observation of the evolution in time of the stresses in the structural domain. Fig. 6.31 reports the evolution of the Von Mises measure of the Cauchy stress, in the point of maximum measured stress in the simulation. This point is the same for all the cases, and it corresponds to the constraint at the leading edge. The time axis for both plots is referring to the entire

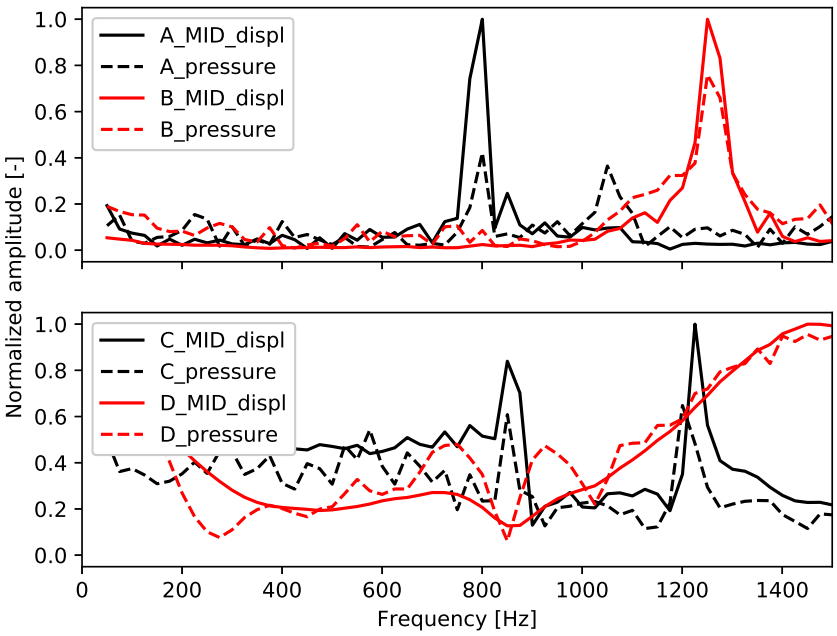

FIGURE 6.29: Comparison of pressure and displacement dynamics in frequency domain

simulation period, and the time starts when the coupling between the fluid system and the structure system is active (it has to be recalled that the initialization of the fluid field is conducted by keeping the blade fixed in place and not responding to the fluid, i.e. the FSI coupling is not active during the initialization).

From the plots in Fig. 6.31, the two different dynamic behaviours emerge, due to the relaxation of the leading edge constraint. The first and second subplots from the top report the peak stress for cases "A" and "B". Here, the blade is oscillating around its average position, and consequently the peak stress is constantly alternating between its maximum and minimum value presenting two different modes at different amplitude, according to the Fig. 6.29. The plot relative to case "A" shows a lower amplitude, lower frequency, less stable main mode of the stress dynamics, when compared to the main mode resulting from case "B". In addition to that, the secondary mode for test case "B" also appears more stable in its amplitude fluctuations. The third and fourth subplots from the top refer to the cases "B" and "D", where the coupled system is less stiff due to the imposed displacement of the trailing edge constraint, and the dynamics of the stress it's stable around its average value. In the end, we notice that the stress dynamics for case "C" (Fig. 6.31) keeps on fluctuating around its average value even after a rather large amount of time. On the other hand, test case "D" shows that the maximum average stress is reached after a shorter amount of time, holding a more stable deformed configuration.

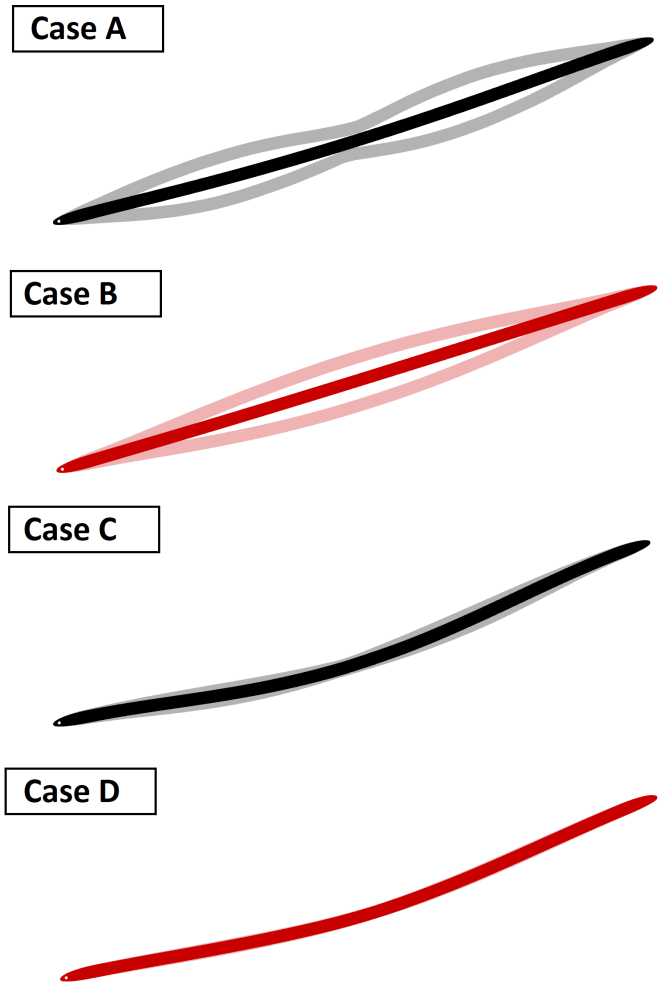

FIGURE 6.30: Average and deformed configurations. From top to bottom, cases "A", "B", "C", and "D".

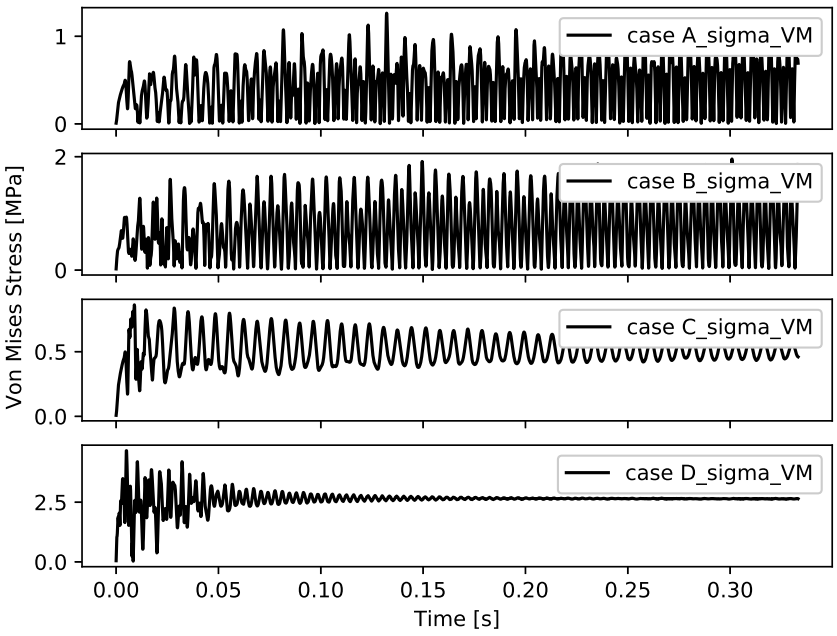

FIGURE 6.31: Time history of the Von Mises stress at the constraint.

To support the above considerations, Fig. 6.30 shows the maximum range of displacement plus the average configuration of the blade in each simulation. In particular, the average configuration is represented in solid color, while the two maximum deformations of a periodic cycle are drawn with a lighter opacity. As in Fig. 6.29, the same color represents the same material. For cases "A" and "B" the deformed configuration span over an high amplitude stable oscillation, while cases "C" and "D" present a more stable behaviour around the average configuration.

Flow survey Fig. 6.32 reports the contour of the average flow velocity magnitude in all the tested cases, from "FIX" to "D", where "FIX" is the first plot from top and "D" is the last plot.

When the blade is considered rigid (case "FIX"), it easy to observe that the averaged flow appears separated on the suction side. Cases "A" and "B" (Fig. 6.32, respectively 2^{nd} and 3^{rd} subfigures from the top) are representing the two cases when the constraint at leading edge is fixed. When the systems are coupled and the blade is able to respond to the flow field and to move, the averaged separation zone appears reduced in size with respect to the rigid body case, since the displacement of the blade is leading to an average configuration more similar to a curved, and thicker, blade profile. However, the displacement is not large enough and not stable enough to reduce the separation on the suction side. When comparing the two cases, we can infer that the blade of case "A", modelled with the low elasticity material leads to less separated flow field.

	FIX	A	B	C	D
C_L (avg)	0.482	0.461	0.370	0.387	0.402
C_D (avg)	0.041	0.037	0.037	0.032	0.030
C_L/C_D (avg)	11.75	12.46	10	12.09	13.40

TABLE 6.9: Results summary

Cases "C" and "D" (Fig. 6.32, respectively 4^{th} and 5^{th} subfigures from the top) are both presenting average deformed configurations that are stable and pronounced. In both cases, the separation at the suction side is only limited to the second half of the blade, and the inflow angle between the camber line and the relative velocity at the leading edge is smaller than in previous cases. However, the separation zone in the final portion of the blade is larger when compared to the cases "A" and "B".

Tab. 6.9 reports the average lift coefficient, average drag coefficient and average section aerodynamic efficiency (the ratio between lift and drag) computed in the simulations. As a general observation, it is possible to highlight that the lift coefficient is reduced for all the test cases with respect to the "FIX" case, as well as the drag coefficient. With the exception of case "B", a general improvement of the ratio C_L/C_D is observed due to the passive morphing of the blade. This specific behaviour has already been observed by Corsini et al. in wind turbines modified with a passive morphing structure at the trailing edge [129].

Final remarks

In this work, a series of 2D simulations have been performed to evaluate the aeroelastic dynamics of a flexible, reversible and symmetric fan blade. The analysis aims to evaluate the behaviour of different configurations of the morphing blade in terms of applied constraint and materials, observing the vibration modes induced by the fluid-structure interaction. Four test cases have been prepared varying the elastic modulus of the material of the blade, and relaxing the displacement constraint at the leading edge. The last choice is used to simulate a possible deformation that would occur in a 3D application of the proposed solution. The test cases are compared with the simulation of the same blade as rigid.

All the simulations have been performed with a finite element based software, implementing a fully coupled FSI algorithm, the Residual Based Variational Multiscale formulation of the Navier-Stokes equations, the Total Lagrangian formulation of the non-linear structure mechanics equations, and the Solid Extension Mesh Moving technique.

When comparing the structure dynamics for different flexible configurations, it's been observed in general an improvement of the stability of the motion thanks to

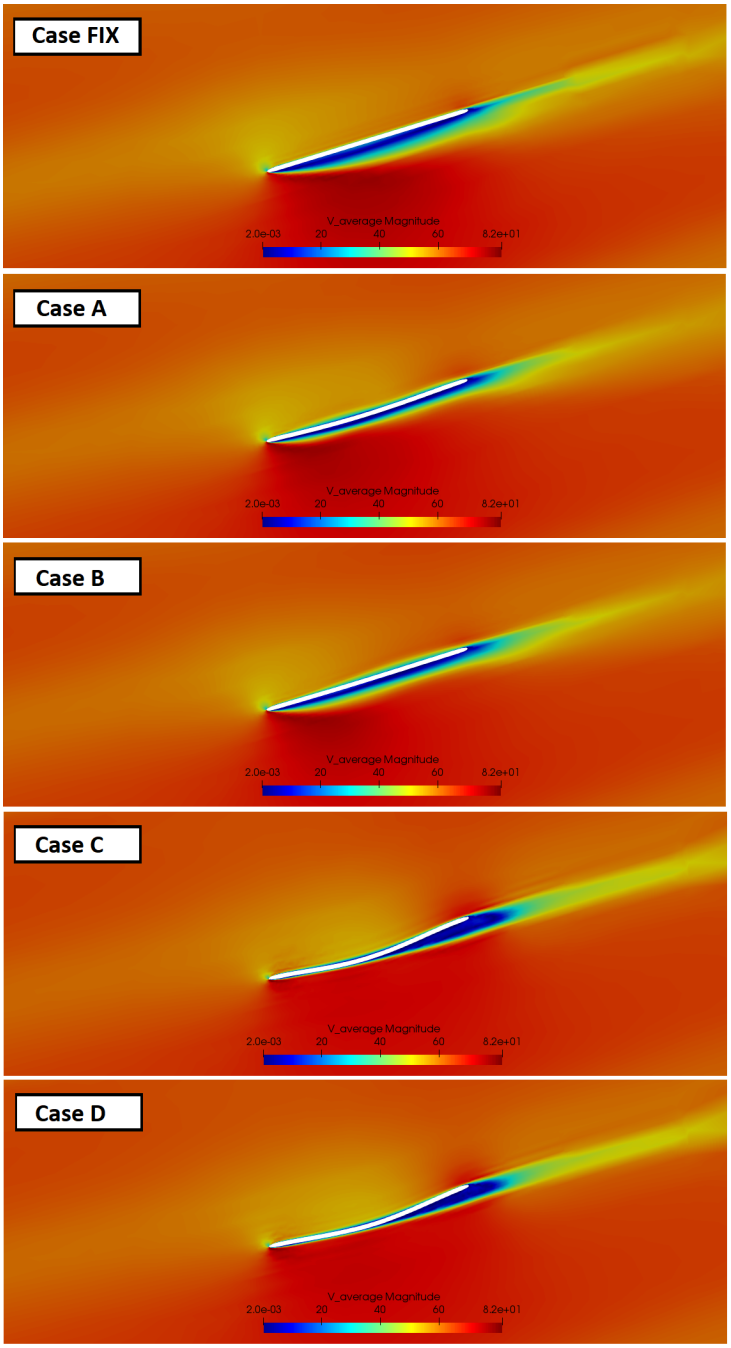

FIGURE 6.32: Average velocity magnitude field comparison.

the displacement of the constraint. From the flow survey, a general reduction of the separation zone in the average velocity field with respect to the fixed blade, is found for every flexible blade solution. The evaluation of the lift coefficient, drag coefficient, and their ratio, for all the test cases, show that the while the lift coefficient is reducing due to the passive morphing, the ratio C_L/C_D is in general improving with the exception of one test case.

Further studies are still required to improve this preliminary analysis and design, but the obtained results are strongly encouraging more effort in this direction: for example, a 3D FSI simulation would be mandatory to overcome the limits of the a-priori imposed displacement at the leading edge constraint; in addition to that, alternative designs of the blade geometry could be proposed, for example by adopting more sophisticated material models, i.e. layered structures.

6.2.2 FSI analysis and simulation of flexible blades in a Wells turbine for wave energy conversion

In the following section a preliminary design and a 2D and 3D computational fluid-structure interaction (FSI) simulation of a flexible blade for a Wells turbine is presented, by means of stabilized finite elements and a strongly coupled approaches for the multi-physics analysis. The main objective is to observe the behaviour of the flexible blades, and to evaluate the eventual occurrence of aeroelastic effects and unstable feedbacks in the coupled dynamics. A series of configurations for the same blade geometry, each one characterized by a different material and mechanical properties distribution will be compared. Results will be given in terms of total pressure difference, supported by a flow survey. The analysis is performed using FEMpar. The adopted models for the FSI simulation are the Residual Based Variational MultiScale method for the Navier-Stokes equations, the Total Lagrangian formulation for the non-linear elasticity problem, and the Solid Extension Mesh Moving technique for the moving mesh algorithm. The strong coupling technique between the physical subsystems adopted here is not often used for dynamic problems in turbomachinery, due to the fact that in this kind of applications, the stiffness of the blade is very high and the aerodynamic feedback is secondary with respect to the inertia forces. For these reasons a weak coupling approach [92] or a modal analysis [93] are often preferred. However, in this application the high flexibility of the blade required a strongly coupling approach.

Wells turbines with flexible blades

Renewable energy is consistently gaining more and more relevance in terms of shares of global electricity generation, growing worldwide from 14% in 2005 to 22% in 2013

[130], and its expected almost to double in 2040 [131]. In spite of the great potential of ocean waves energy as renewable energy source, the production of electricity from ocean waves is still limited if compared to more traditional clean and renewable energy sources such as wind, hydroelectric, solar and bioenergy, and the technology of the conversion system for waves energy recently reached a stagnation point. The reasons of the deceleration in the development of new technologies are multiple, but they all hinge around the predicted costs of wave power, evaluated as non-competitive with respect to other sources [132]. In analogy with others renewable -and non- energy sources and energy conversion systems, a reduction of the predicted costs of waves energy can be achieved by increasing the reliability of the equipment, increasing the efficiency of the conversion system, and reducing the capital investment costs [132]. In this framework, Wells turbines already represent a largely developed and consolidated technology for wave energy conversion in Oscillating Water Columns (OWC) devices. Traditional Wells turbines present limits in terms of average efficiency, because they mostly operate in off-design flow conditions. Furthermore, their peak efficiency is low, because of the symmetric, untwisted blades. It is trivial to observe that a non-symmetric, twisted blade profile could improve the performance of the machine. A Wells turbine with flexible, polymeric blades could solve at once multiple issues: a flexible blade can be designed in order to improve efficiency in off-design conditions (i.e. low mass flow rates) and to improve peak efficiency due to a passive-adaptive blade morphing, in both flow directions, without any need of guiding vanes. Such blade could also improve the reliability of the machine in a highly corrosive environment such as saline aerosol, and it can drastically reduce the capital investment and operative costs due to the adoption of a cheaper material with respect to steel. This preliminary study presents a numerical analysis of a Wells turbine with flexible blades by performing a coupled 2D computational Fluid Structure Interaction (FSI) dynamic simulation using stabilized Finite Element Method (FEM), for the solution of the fluid field and the structural dynamics.

The analysis can provide useful insights about the deformations and stresses of the blade and about the performance improvements due to the flexibility of the blade in different operative conditions. Complex FSI dynamics and large deformations are expected, indeed, the low stiffness structure is selected such to be able to deform under the aerodynamic loads. In addition to that, the deformation and the variation of the shape must be evaluated with respect to the OWC period and to the dynamic response of the blade, and consequently, the choice of the material and the blade thickness can be accordingly designed. In this section, a series of alternative solutions for the material of the blade of the Wells turbine, in terms of elasticity and density distribution, are proposed. As a preliminary study for this application, no optimization for the choice

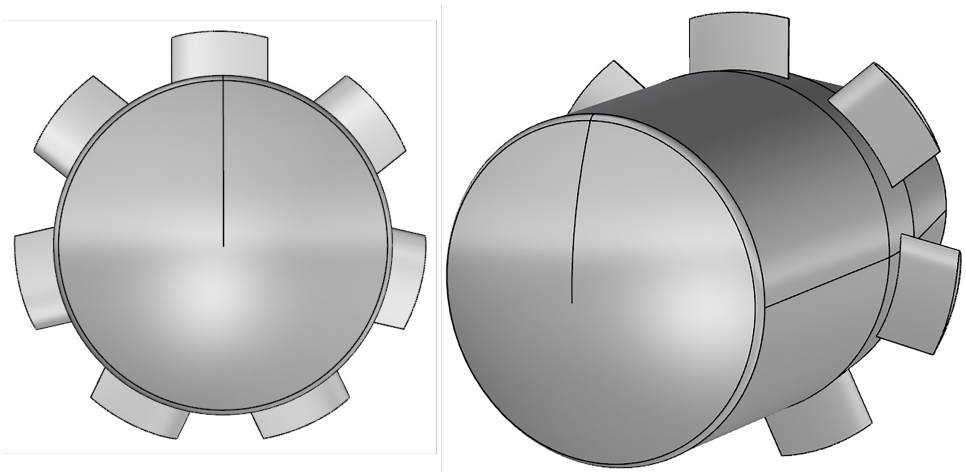

FIGURE 6.33: 3D CAD of the Wells turbine

of the material is performed, but rather there's a specific focus on the observation of the differences in the flow field and structural response of the blade, which is very thin, flexible, and constrained at the leading edge and trailing edge. An iterative, trial and error approach to define the material for each configuration is adopted, starting from a standard plastic material used for marine applications.

Blade geometry, preliminary analysis

In this work, a simulation of a peculiar design of a small Wells turbine will be presented. The design is directly descending from the DIMAFP-TW1.5, a prototype 1.5 Kw 3000 rpm Wells turbine for Mediterranean operations (i.e. designed to operate at an average wave height of 0.8m and an average period of the waves of 4s [133]), with a 9 bladed 0.5m diameter rotor, and two stators with 17 blades each. [134, 94]. Here, the object of the simulation is a 7 bladed, rotor only Wells turbine, with a NACA 0002 profile: such a low thickness profile is required to provide the blade with the desired flexional behaviour with realistic materials. The chord distribution is constant along the blade span. Further details on the blade can be found in Tab. 6.10, while a 3D sketch of the Wells Turbine can be seen in Fig. 6.33.

Starting from the NACA 0002 symmetric blade, the analysis begins from a CFD study where the performance of a series of blade profiles deformed a priori are evaluated, where the curvature degree varies by modifying the camber line, considering a curvature coefficient m as a percentage of the chord length, with m respectively equal to 2, 5, 7, 9, 11 and 15%, by assigning such value to the m coefficient in the expression

Casing diameter [mm]	500
Hub-to-casing diameter ratio	0.75
Aspect Ratio	0.5
Blade count	7
Blade chord [mm]	120
Blade height [mm]	60
Tip clearance [mm]	2.5
Solidity	0.6
Rotating speed [rpm]	300

TABLE 6.10: Wells turbine blade specifications

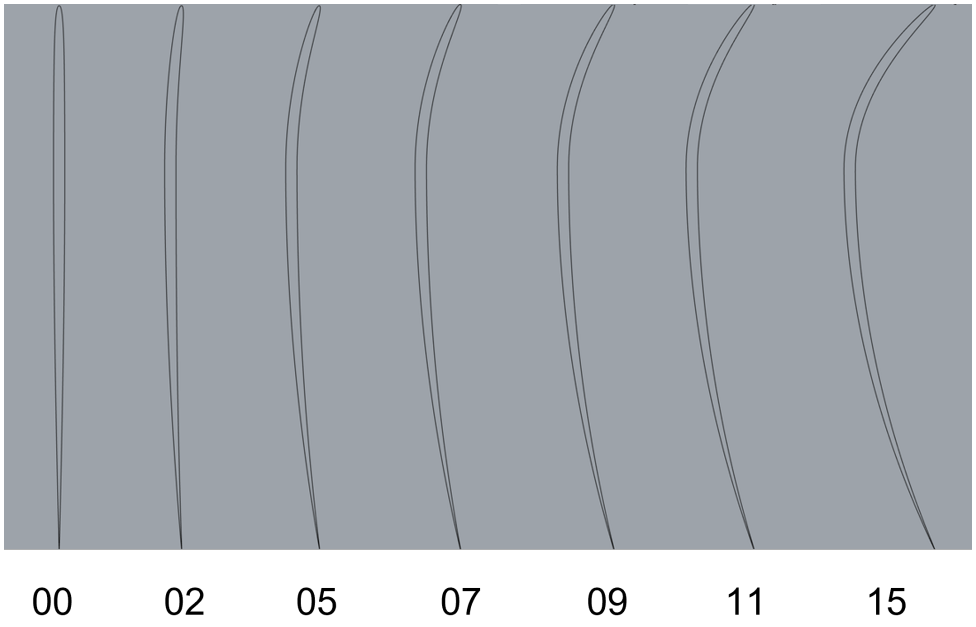

FIGURE 6.34: Resulting a priori deformed profiles.

of the camber line of Eq. (6.4):

$$y_c = \begin{cases} \frac{m}{p^2}\left(2p\left(\frac{x}{c}\right) - \left(\frac{x}{c}\right)^2\right) & \text{if } 0 \leq x \leq pc \\ \frac{m}{(1-p)^2}\left((1-2p) + 2p\left(\frac{x}{c}\right) - \left(\frac{x}{c}\right)^2\right) & \text{if } pc \leq x \leq c \end{cases} \tag{6.4}$$

Where y_c and x are the coordinate of the camber line, p is the position where the blades has maximum curvature and m is the maximum curvature express as percentage of the chord, obtaining the profiles showed in Fig. 6.34.

From the simulations, a series of characteristic curves are obtained as shown in Figg. 6.35, 6.36 and 6.37, respectively in terms of total pressure drop, power and efficiency. When looking at those lines, it is almost a natural consequence to consider the possibility to obtain an envelope curve, as shown in Figg. 6.38, 6.39 and 6.40, only by the passive morphing of the flexible blade, as if the material were able to deform in

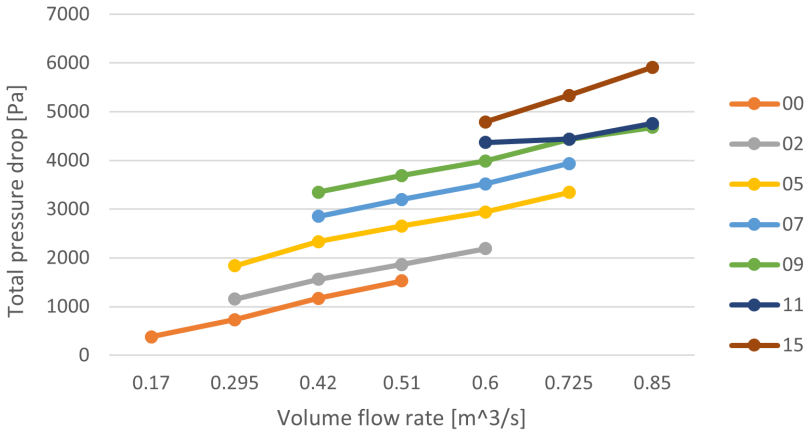

FIGURE 6.35: Characteristic curves for different, *a priori* imposed, blade geometry configurations for the Wells turbine in terms of total pressure drop.

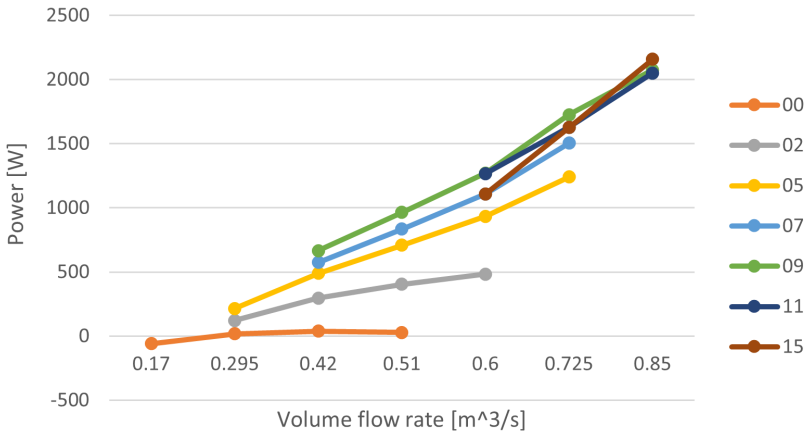

FIGURE 6.36: Characteristic curves for different, *a priori* imposed, blade geometry configurations for the Wells turbine in terms of power.

such a way to assume those positions as operative configurations.

With such background, the main objective is to evaluate the possibility of adopting an elastic material, characterized by low stiffness, to obtain a curved blade profile comparable to the one observed in Fig. 6.34, by exploiting the passive morphing capabilities of such a material. By doing so, a complex series of new vibrational dynamics -that must be considered when observing the total dynamic- are introduced, and for such reason both frequency analysis tools and FSI simulations are performed.

2D computations

Constraints, adopted materials In the 2D simulation of this flexible blade turbine, the blade is constrained to the hub as if two rigid clamps are holding the leading edge

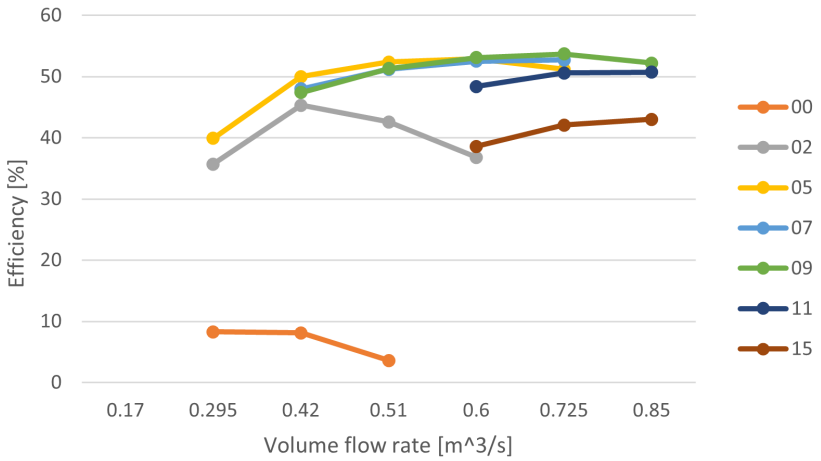

FIGURE 6.37: Characteristic curves for different, *a priori* imposed, blade geometry configurations for the Wells turbine in terms of efficiency.

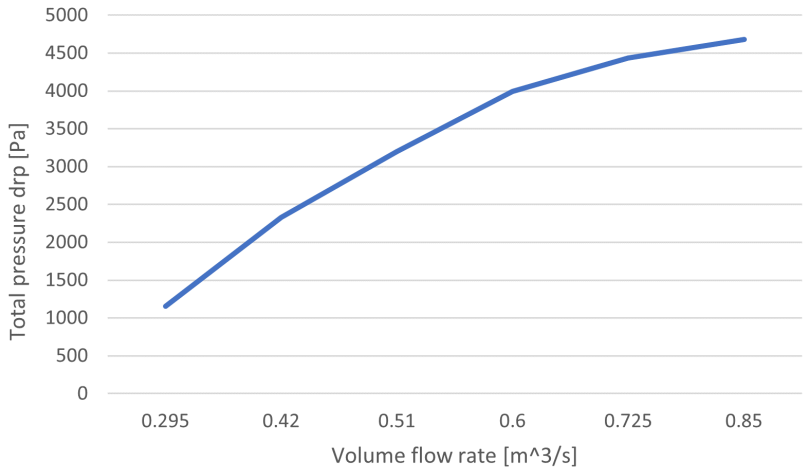

FIGURE 6.38: Envelope of the characteristic curves of Fig. 6.35.

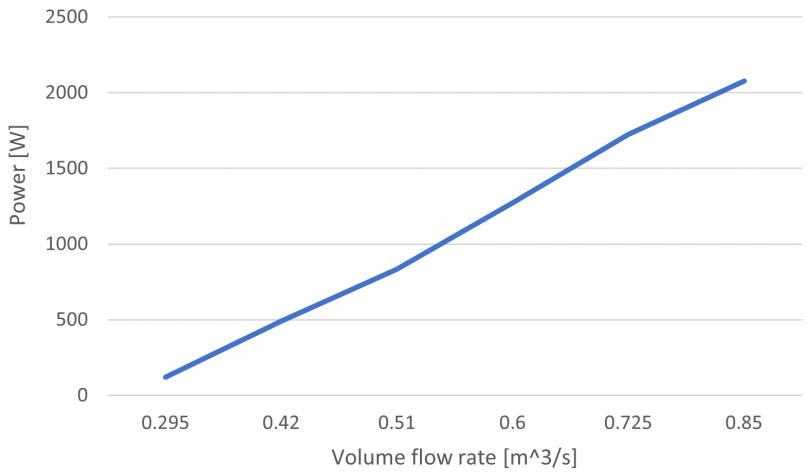

FIGURE 6.39: Envelope of the characteristic curves of Fig. 6.36.

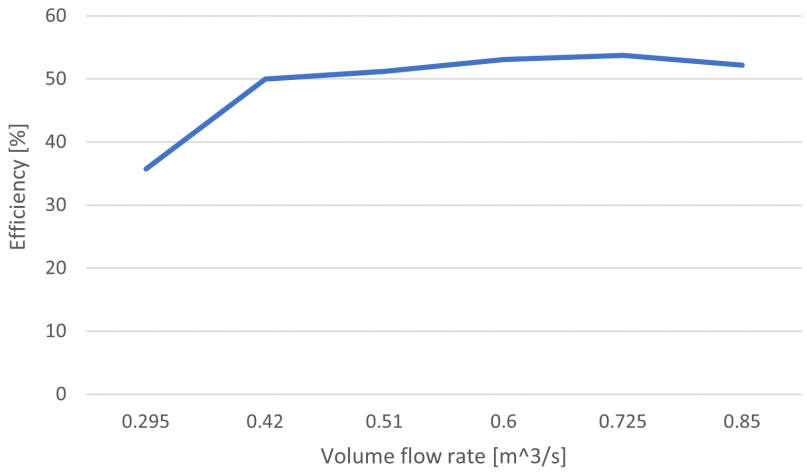

FIGURE 6.40: Envelope of the characteristic curves of Fig. 6.37.

and the trailing edge along the entire span of the blade, as illustrated in Fig. 6.41.

For this application, the objective is to simulate a flexible blade for a Wells turbine, without recurring to unrealistic materials, or unrealistic elasticity distributions. The reference material is Nylon 66/6 [135], a widely used polymer, reinforced with 30% glass fiber. Several variations of this material will be explored, with the objective to observe the dynamic response, deformations, and stable configurations for the blade. Additional details about the material will be provided for each case. In particular, there will be a comparison between the flow around an homogeneous isotropic Nylon 66/6 blade against other configurations with a distribution of elasticity and density differentiated in three zones: a zone next to the leading edge, a zone next to the trailing edge, and a central larger zone. This strategy allowed more control over the displacements of the blade, with the opportunity to compensate, for example, the extreme thinness at the trailing edge. A scheme of the subdivision in zones of the blade can be found in Fig. 6.42. The names of the zones, i.e. leading edge (LE), chord centre (MID), and trailing edge (TE), will be used in the upcoming sections to identify each zone.

In order to explore a larger amount of possible dynamic behaviours, it is useful to compare five different flexible blades characterized by a different material. All the blades where simulated under the same inflow conditions given in Sec. 6.2.2, so that the test cases are illustrated in Tab. 6.11 in terms of Young's Modulus E, density ρ and Poisson's ratio ν. Case A refers to the Nylon 66/6 blade, and case B, C, and D refer to the blades with modified elasticity and density distributions, according to the scheme of Fig. 6.42.

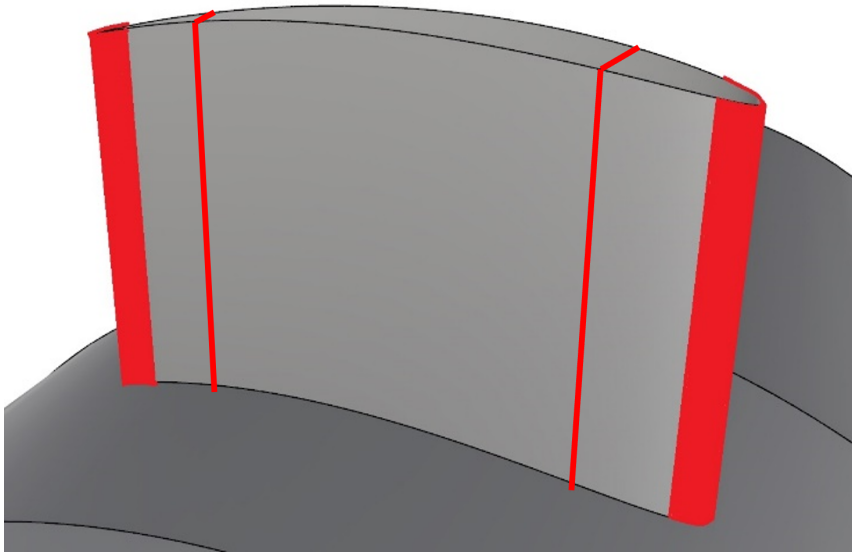

FIGURE 6.41: Clamps constraining the blade to the hub of the machine for the 2D simulation.

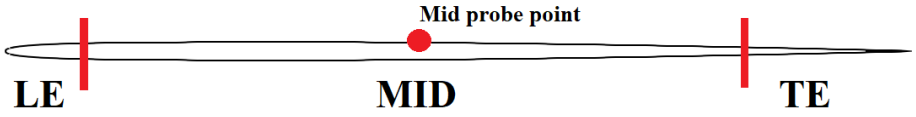

FIGURE 6.42: Subdivision of the blade in different zones, each one with a different material. The dot represents the point that will be taken as probe point.

	Case A	Case B	Case C	Case D	Case E
$E_{LE}[Pa]$	5.82e+9	5.0e+7	5.0e+7	5.0e+7	5.0e+7
$E_{MID}[Pa]$	5.82e+9	5.82e+9	2.92e+9	5.82e+9	2.92e+9
$E_{TE}[Pa]$	5.82e+9	30.82e+9	15.82e+9	30.82e+9	15.82e+9
$\rho_{LE}[kg/m^3]$	1270	3000	3000	3000	5000
$\rho_{MID}[kg/m^3]$	1270	1270	1270	2540	3000
$\rho_{TE}[kg/m^3]$	1270	2700	2700	2700	4000
ν_{LE}	0.381	0.3	0.3	0.3	0.3
ν_{MID}	0.381	0.381	0.381	0.381	0.381
ν_{TE}	0.381	0.3	0.3	0.3	0.3

TABLE 6.11: Test cases

Numerical setup The computational domain is a rectangle obtained extracting the cascade section of the turbine at radius $r = 246mm$, corresponding to the 95% of the blade span. The very small ratio between the axial and tangential velocity components at the inflow and the high solidity of the turbine, allowed to use a relatively small domain. The rectangle extends, in the axial direction, for one chord upstream and one chord downstream. The resulting mesh is shown in Fig. 6.43, it contains about 5000 fluid nodes and 1500 solid nodes. An inlet relative velocity condition of $v = 78.3\ m/s$ is imposed at the inlet boundary. The relative velocity is obtained by considering the nominal flow rate of $Q = 0.85\ m^3/s$ and the rotational speed of $\omega = 3000rpm$. The simulated wells turbine is operating at max load conditions, to obtain the maximum deformation of the blade. At the outflow boundary, a zero gradient condition is imposed. The remaining boundaries are considered as periodic boundaries, so that the domain can be considered as a periodic cascade. At the wall, the no-slip condition is imposed in a weak form, according to Bazilevs [80]. This technique will result in an effective relaxation allowing a slip velocity when a not sufficient minimum wall distance is locally encountered to resolve the flow at the boundary layer. The first cell is at a distance of $3.3 \times 10^{-4)}\ m$ and $Re \approx 3 \times 10^6$. The simulation is carried on for 4000 time steps, with a $dt = 1 \times 10^{-5}\ s$, corresponding to two revolutions of the turbine. The FSI simulation starts from the full resolved dynamic flow field solution, obtained by keeping the blade rigid.

Results and discussion The main objective of the 2D simulation was to evaluate the possibility to realize flexible blades for a Wells turbine. The dynamic behaviour of the five blades defined above, plus a fixed blade considered as a reference, in a cascade configuration is obtained by the observation of the CFD results. Each blade deformed in a different way, some of them barely moved around their original position, others presented an unstable dynamic behaviour because of a positive feedback between solid mechanics and fluid dynamics of the field. In order to evaluate the difference in performance for the different blades, it is possible to see in Tab. 6.12 the total-to-total average pressure drop, computed in two different sections, upstream and downstream with respect to the blade, both at the same distance of half of the chord length. From the table stands out that the larger pressure drop is obtained with the configuration D. In addition to that, it is possible to observe that all flexible blades present a larger pressure total difference with respect to the fixed blade, encouraging further optimization of the design and of the choice of the material.

A primary remark of this work is that the choice of the right material is crucial, because the final shape of the blade, the stability of its dynamic response, and consequently the performance of the turbine, are all affected by the choice of the material

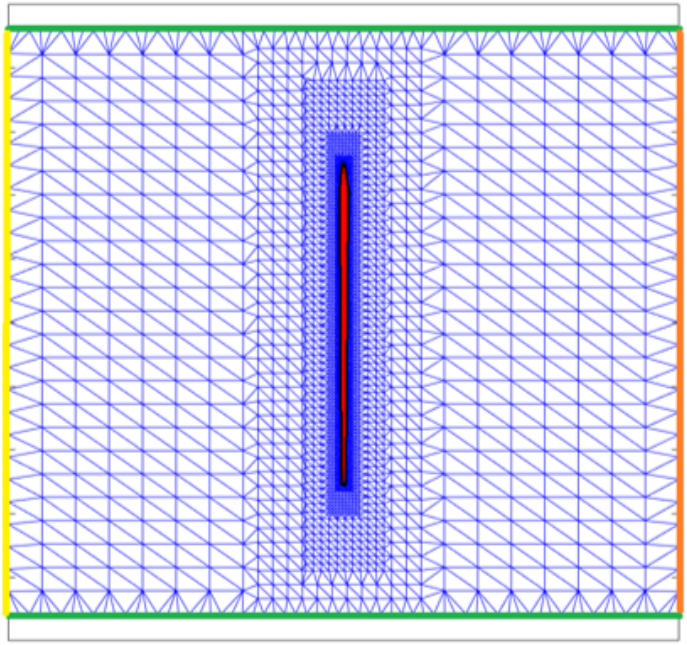

FIGURE 6.43: Triangular mesh of the computational domain (blue for fluid domain, red for structural domain). The boundary conditions are highlighted in different colours (yellow for inlet boundary, black for wall boundary/interface, green for periodic boundaries).

	Case Zero	Case A	Case B	Case C	Case D	Case E
Up $P_{TOT,avg}[Pa]$	16921.91	17074.44	17047.97	17254.69	17224.36	16970.93
Down $P_{TOT,avg}[Pa]$	15900.64	15952.14	14984.01	15761.88	14459.03	14988.23
Drop $P_{TOT,avg}[Pa]$	1021.27	1122.30	2063.95	1492.81	2765.33	1982.69

TABLE 6.12: Upstream-Downstream Total pressure differences

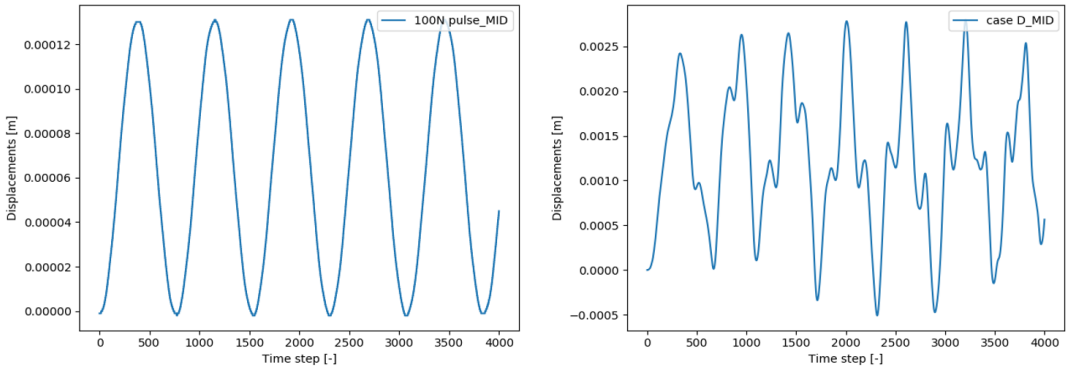

FIGURE 6.44: Time variation of the displacement of the center of the blade D: response to uniformly distributed impulse (left), FSI simulation (right).

more than any other design variable. In order to highlight the occurrence of aeroelastic coupling, it's useful to see in Fig. 6.44 the comparison of the displacement of the center of the blade (e.g. in configuration D) when it is subjected to a uniformly distributed $100N$ impulse in the y- direction, and during the FSI simulation. In Fig. 6.45 the main frequencies of the dynamics are compared by plotting the normalized Fast Fourier Transformations (FFT) of the computed displacement and the static pressure evolution. It's easy to notice from Fig. 6.45 both the variation in frequency of the main dynamics and the occurrence of new modes (due to the aeroelastic coupling). In addition to that, in the same Fig. 6.45 stands out that the main frequency of the pressure is perfectly overlapping one of the secondary modes of the emerged from the FSI simulation, and therefore it's possible to conclude that such secondary mode, not present in the structure-only dynamic, it's entirely originated by the aeroelastic coupling.

A flow survey obtained with the 2D FSI simulation is reported. For the sake of brevity, only images from the configuration D, which resulted the most promising configuration according to Tab. 6.12, are reported for the flow survey. Fig. 6.46 presents the total pressure and the velocity magnitude, both averaged in time. Fig. 6.47 shows two instantaneous fields of static pressure and velocity magnitude, both describing the same time step, at which the displacement of the central probe point was maximum.

For the sake of completeness, in Fig. 6.48 it is possible to see the Von Mises stress distribution for the test Case A (i.e. characterized with homogeneous Nylon 66/6). In the picture, only the stress distribution for Case A is shown, because it is only possible to compare the simulation results with real yield stress values for the Nylon 66/6. The yield stress value considered for the material is $Y = 162MPa$, and as shown in the picture the entire blade is below the threshold value set as $0.2Y$ for safety reasons, with the exception of the zones near the constraints at the leading edge and trailing edge. Those peaks in the stress distribution suggest to modify the fixed constraints

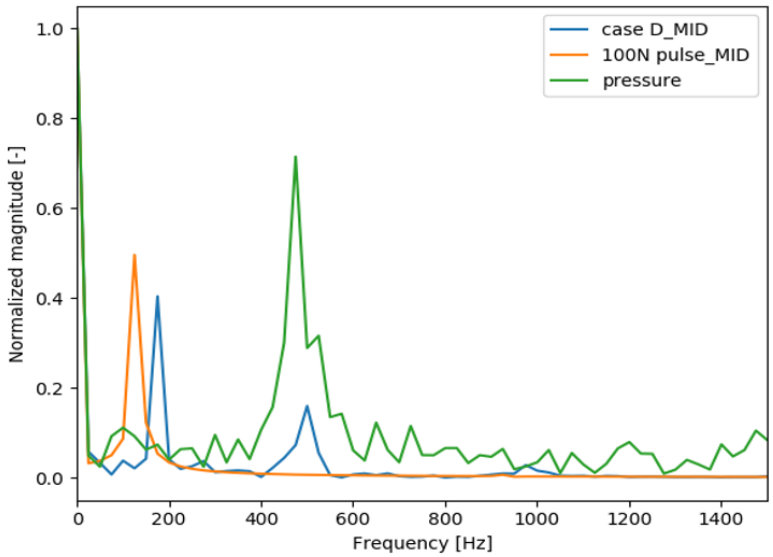

FIGURE 6.45: Normalized FFT of the displacement of the of the center of the blade D: frequency response to uniformly distributed impulse (orange), FSI simulation (blue), static pressure for the fixed blade configuration (green).

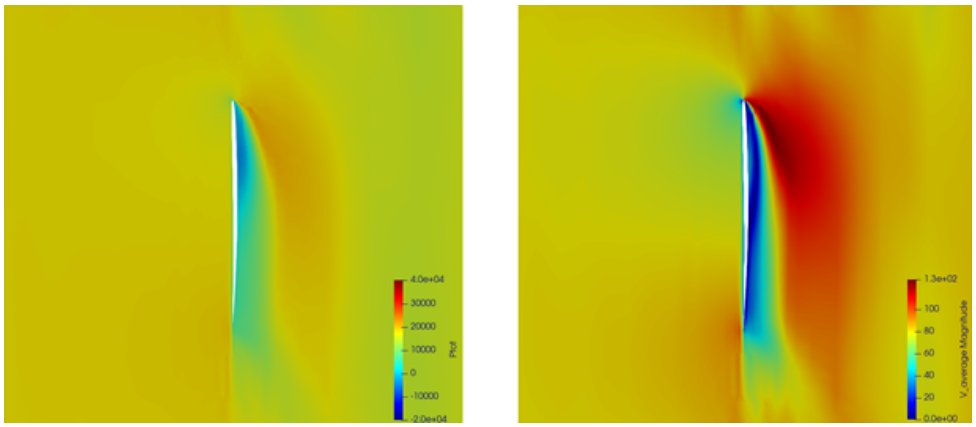

FIGURE 6.46: Average total pressure field for case D (left), average velocity magnitude field for case D (right).

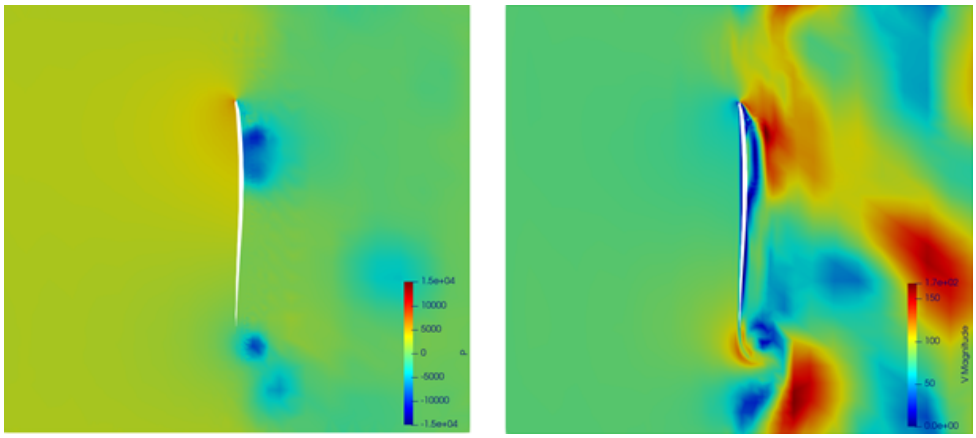

FIGURE 6.47: Instantaneous static pressure field for case D (left), instantaneous velocity magnitude field for case D (right).

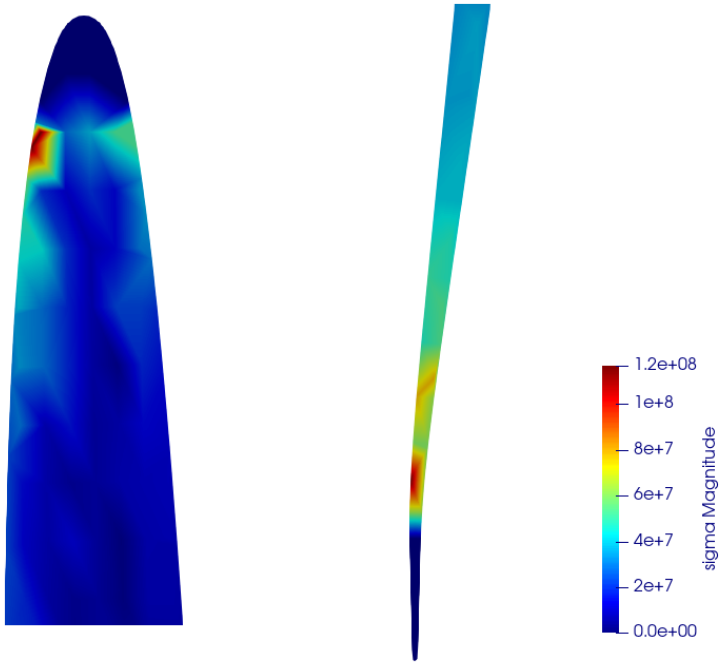

FIGURE 6.48: Von Mises stress distribution details for case A: leading edge (left) and trailing edge (right), entire blade (up).

with a different constraint, i.e. a hinge. In any case, the distribution still proves the correctness of the initial choice of the material. Fig. 6.49 presents the comparison, for all test cases, of the deformed blade profile with respect to the undeformed configuration. The deformed profile is obtained as an average of the positions occupied by the blade during the simulation time. It is easy to observe that the average displacement varies by considerable amount from one material to the other, and that the obtained curvature of the blade by the adoption of those materials configurations, are not large enough to be comparable to the a priori imposed curvatures of the blades tested to obtain the curves of Fig. 6.34, but the effect on the flow of the passive morphing blade is still considerable.

3D computations

Constraints, adopted materials For the 3D simulation, the structure and the constraints are slightly modified with respect to the former 2D simulation. In particular, the trailing edge is considered fixed, while the leading edge is kept fixed in place by a thin steel bar, to allow torsion and the displacement in the tangential direction of the leading edge, as illustrated in Fig. 6.50. For this 3D simulation, only one specific configuration of the materials is observed, mainly due to the larger computational effort with respect to the 2D counterpart: an optimization process is however advised to reduce

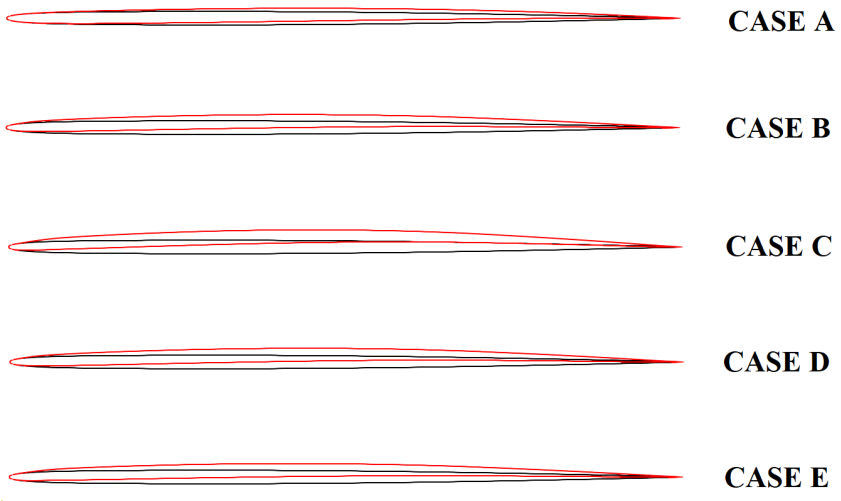

FIGURE 6.49: Comparison of the deformed blade profile (in red, average) and the undeformed blade profile (in black) for all test cases.

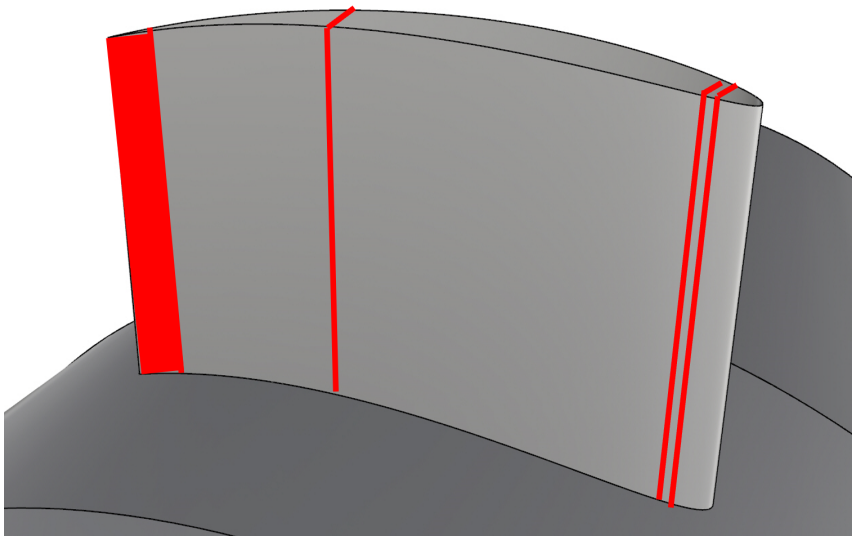

FIGURE 6.50: A thin torsion bar keeps the leading edge in position for the 3D simulation.

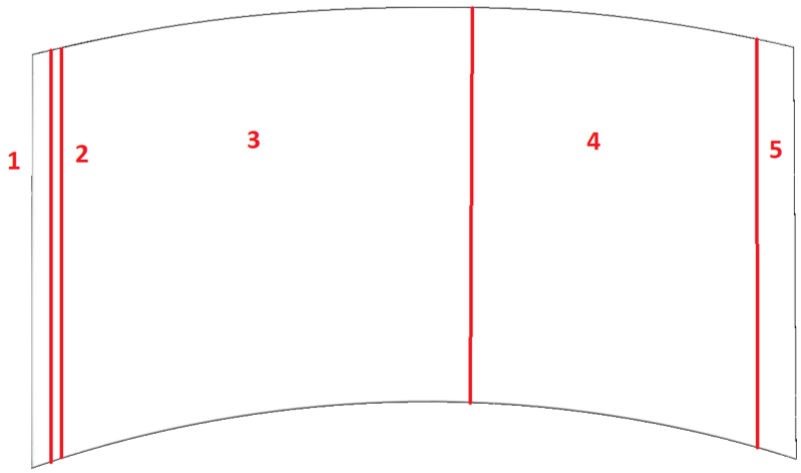

FIGURE 6.51: Subdivision of the 3D blade in five different zones, each one with a different material.

	Values
$E_1[Pa]$	2e+8
$E_2[Pa]$	2.2e+11
$E_3[Pa]$	2e+8
$E_4[Pa]$	2e+9
$E_5[Pa]$	-
$\rho[kg/m^3]$	3000
ν	0.3

TABLE 6.13: 3D simulation material setup

the amount of sub-optimal configurations. The specific configuration of materials is represented by both Fig. 6.51 and by Tab. 6.13: specifically, the blade is subdivided in 5 different zones, of solid subdomains, each one differing from the other for the Young's modulus according to the table. The motivation of such subdivision is based on the previous experience of the 2D simulation, as well as the experience of the application of Sec. 6.2.1.In fact the five zones are organized in such a way that they can behave both as a relaxed constraint as well as a low stiffness material: the high stiffness of the thin steel bar can be considered as an attempt to properly model the a priori imposed displacement presented of Sec. 6.2.1, while the two different zones at the center of the blade are set up in order to obtain a deformed configuration that could possibly be more similar to the one considered for the preliminary analysis of Sec. 6.2.2.

Numerical setup The 3D computational domain is a $51.428\,\text{deg} = 0.8975rad$ cyclic section of the rotor vane (see Fig. 6.52), corresponding to a seventh of the entire rotor. The domain extends, in the axial direction, for one and a half chord upstream and one and a half chord downstream. The resulting mesh is shown in Fig. 6.53, it contains

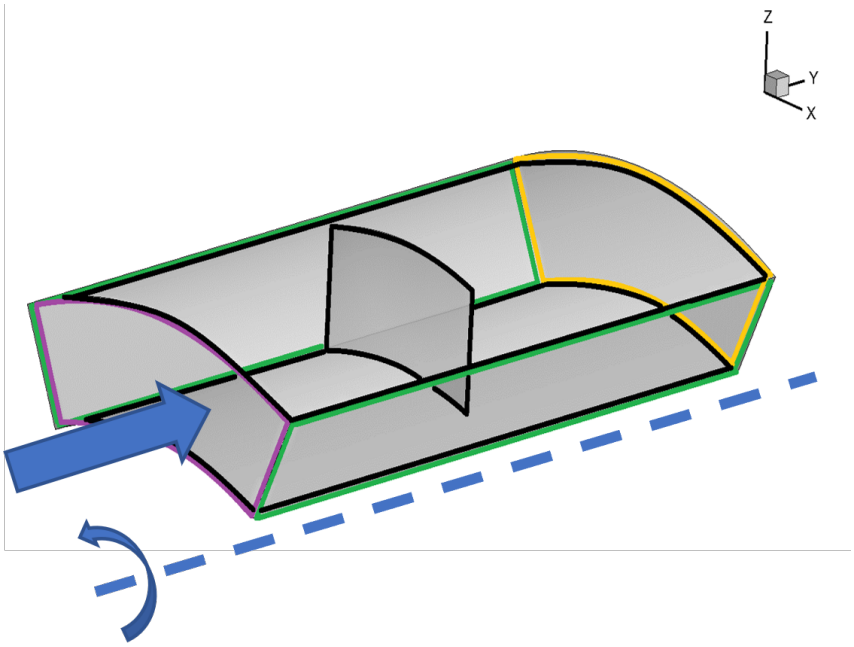

FIGURE 6.52: The fluid computational domain for the 3D simulation. The boundary conditions are highlighted in different colours (yellow for inlet boundary, black for wall boundary/interface, green for periodic boundaries).

about 2×10^5 fluid nodes and 1.1×10^6 elements , while the solid mesh is composed by $3.5{\times}10^4$ nodes and $1.7{\times}10^4$ elements. The operating conditions, as well as the boundary conditions, are set identically to the 2D simulation of Sec. 6.2.2: at the inlet boundary, a relative velocity condition of $v = 78.3\ m/s$ is imposed for the same nominal flow rate of $Q = 0.85\ m^3/s$ and a rotational speed of $\omega = 3000rpm$, at max load condition, while a zero gradient condition is imposed at the outlet boundary. The remaining boundaries are considered as periodic boundaries. Similarly to the 2D computation, also in the 3D simulation the no-slip condition at the wall is imposed weakly, allowing a first cell distance of $2.3 \times 10^{-4)}\ m$ with $Re \approx 3 \times 10^6$. The simulation is carried on for approximately 40000 time steps, with a $dt = 5.5 \times 10^{-5}\ s$, corresponding to 40 revolutions of the turbine, after an initialization period with the blade considered fixed for approximately 55 revolutions. In Fig. 6.52, the boundary conditions are highlighted with the same color scheme adopted for the 2D simulation:yellow for inlet boundary, black for wall boundaries and fluid-solid interface, green for periodic boundaries.

Results and discussion The main objective of the 3D FSI simulation was to enrich the information obtained with the 2D simulation by including three dimensional effects, increasing the resolution of the numerical discretization in order to observe more complex turbulent structures, and testing an alternative design of the constraints and of the distribution of the materials on the blade, considering the results obtained with

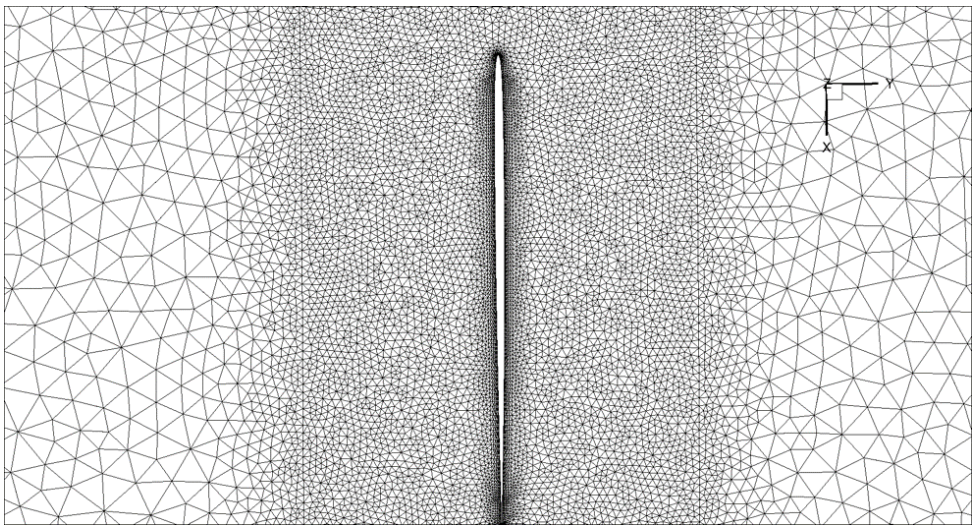

FIGURE 6.53: A cylindrical section of the 3D fluid mesh.

	Fixed	Adaptive
Power $[W]$	1004	1044
Drop $\Delta P_{TOT,avg}[Pa]$	2762	3118
Flow rate $Q[-]$	0.85	0.85
Efficiency $\eta[-]$	0.42	0.39

TABLE 6.14: Upstream-Downstream Total pressure differences

the applications of Sec. 6.2.1 and Sec. 6.2.2.

In order to evaluate the difference in performance for the different blades, it is possible to see in Tab. 6.14 the total-to-total average pressure drop, the power obtained by the adaptive Wells, as wells as the efficiency of the devices, compared with its fixed counterpart. From the table stands out that there is not a straight improvement in the performance of the fan in terms of efficiency, however, the adaptive blade is expressing an increasing total-to-total average pressure drop.

It has to be noted, however, that no structural optimization has been performed at this stage of the presented work. In analogy with the considerations of Sec. 6.2.2, a proper optimization of the choice of the material is strongly advised, in order to obtain the desired deformed and stable configuration, and consequently, an effective improvement of the performance.

The relevance of the effects related to the aeroelastic coupling for the 3D blade is observed in Fig. 6.55, where the normalized FFT of displacement in the axial direction (red, continuous line in Fig. 6.55) of a structural node (positioned at the center of the chord, at the tip section of the blade, red dot in Fig. 6.56, characterized by the displacement reported in Fig. 6.54) is compared with the normalized FFT of the static pressure of a fluid node (near the leading edge, outside the boundary layer, blue dot in Fig. 6.56). In the figure, two main modes are emerging from the coupling in both

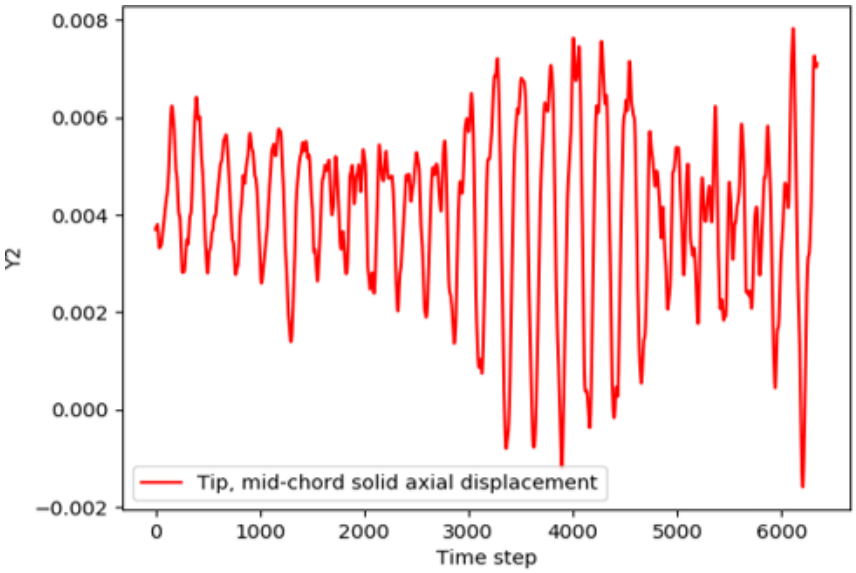

FIGURE 6.54: Time variation of the displacement in the axial direction of the center of the chord, at the tip

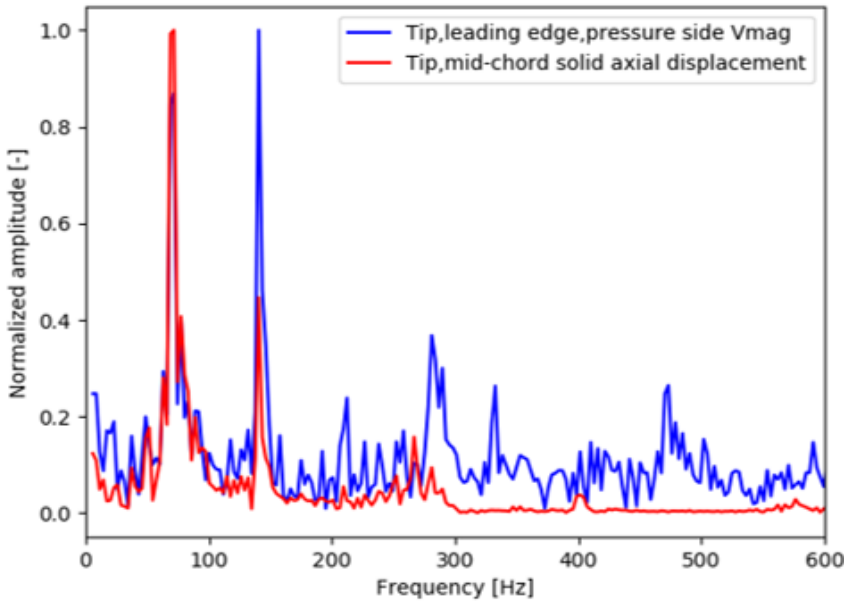

FIGURE 6.55: Normalized FFT of the displacement in the axial direction of the center of the blade (red) at the tip, compared with the normalized FFT of the static pressure (blue) at the leading edge.

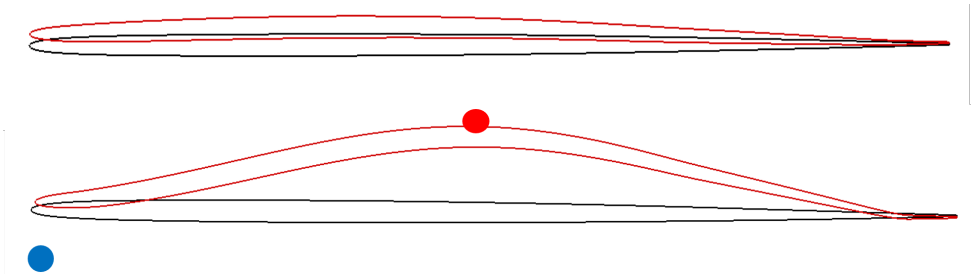

FIGURE 6.56: Comparison of the structural deformed configuration (red line) with respect to the fixed blade (black line), averaged in time (up) and for a given time.

fluid and structural domain, at the same frequencies, showing an incipient resonating dynamic. The higher frequency modes of the fluid dynamics seems not to affect the structural dynamics. The first mode of the FFT in Fig. 6.55 can be related to the flapping dynamics that originates from the thin steel bar close to the leading edge: the attempt of relaxing the constraint by allowing a displacement at the tip in the proximity of the leading looks interesting from a design perspective (and from the perspective of the design of a numerical experiment), however the elastic behaviour of the bar must be taken into account in the optimization process.

In Fig. 6.56 are compare the average deformed configuration (in red, above) and the instantaneous deformed configuration (red, below) at the maximum displacement of the mid-chord structural node at the tip section, with the fixed configuration (in black, above and below): in spite of the pronounced maximum displacement, the average deformed configuration is not sufficiently different from the fixed one, and due the relatively high frequency of the displacement dynamics, it looks not stable enough to remain in a rest position. In Fig. 6.57 is visualized the structural displacement in the axial direction in terms of isocontours on the blade surface, on an instantaneous deformed configuration. The blade assumes a concave shape when invested by the flow, and the camber curves accordingly. The maximum of the axial displacement is concentrated at the center of the blade, in the region closest to the tip. The thin bar at the leading edge allows a non negligible displacement, that originates a flapping motion in the axial direction and a small variation in the incidence angle.

Fig. 6.58 shows an instantaneous Von Mises measure of the stresses on the blade: the Von Mises stress is concentrated on the support bar, while the surface of the blade almost not sollecitated. In spite of the stress at the constraint being well above the steel ultimate yield stress, its oscillating nature (as it can be deducted by the displacement time history of Fig. 6.54 and by the FFTs in Fig. 6.55) could easily lead to a disruptive failure by fatigue. A possible solution, yet to be optimized and tested, could be to increase the thickness of the bar and evaluate the stress cycle on a longer observation time period.

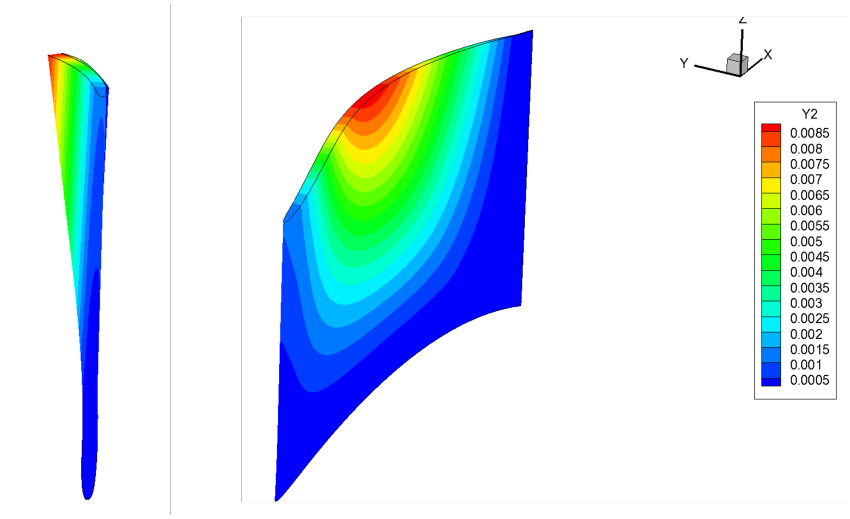

FIGURE 6.57: Structural displacements in the axial direction for the adaptive blade: front view (left), suction side (right)

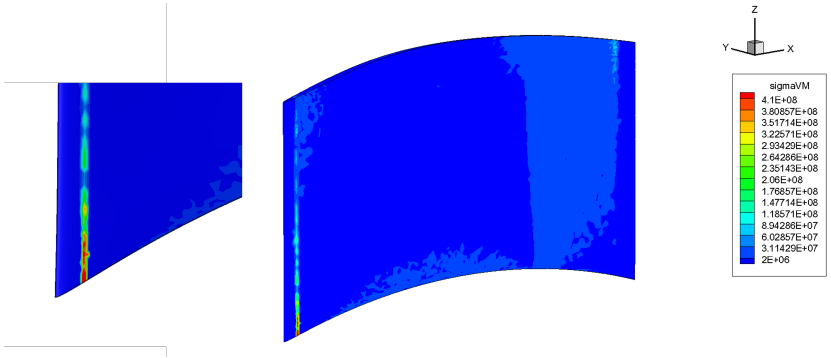

FIGURE 6.58: Von Mises stress distribution on the adaptive blade: detail at the leading edge constraint (left), entire blade (right)

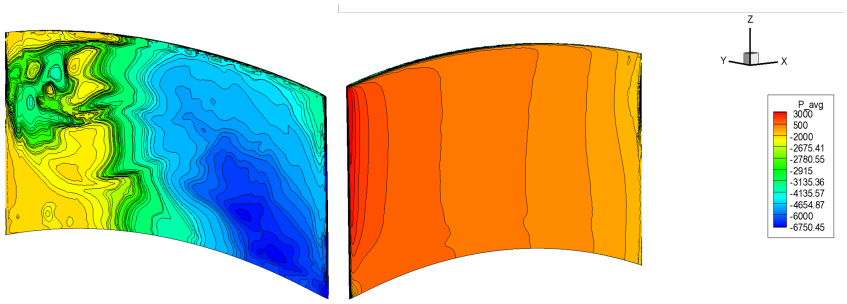

FIGURE 6.59: Average static pressure distribution on the suction side (left), and pressure side (right) of the adaptive blade

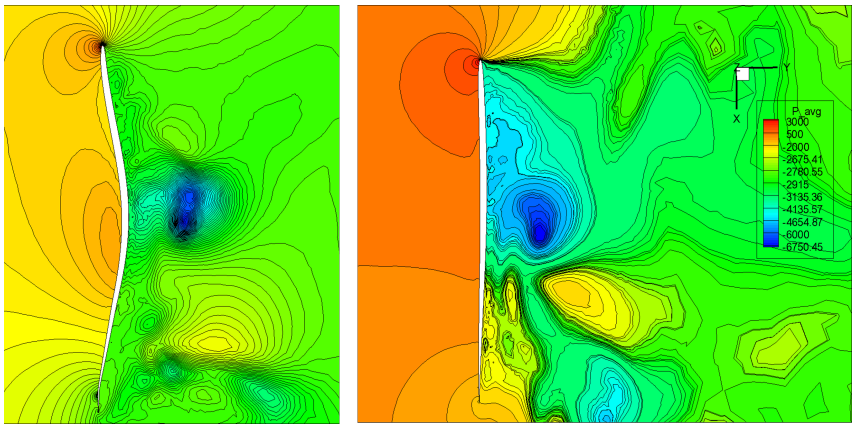

FIGURE 6.60: Average static pressure distribution (left), instantaneous static pressure distribution (right) at a section close to the tip radius of the adaptive blade.

A flow survey is reported as well for the 3D unsteady simulation. In Fig. 6.59 are presented the isocontours of the average static pressure distribution on the suction side and pressure side of the flexible blade. The average pressure distribution on the blades shows a large depression zone on the suction side, rising from the blade root. On the suction side, the average pressure is regularly distributed. However, a small depression area can be seen near the root, at the leading edge. The same static pressure can be observed from a cylindrical slice at the tip of the adaptive blade in Fig. 6.60 both for the average field, and for the instantaneous field.

In Figg. 6.61 and 6.62 is possible to see the isocontours of the velocity magnitude and tangential velocity for the average flow field and for the most deformed configuration at the tip radius. A large recirculation zone is observed along the entire chord of the blade, with its maximum absolute value at the center of the blade. As a consequence, the suction side is experiencing an adverse velocity gradient at the wall for it's entire length, and the flow is completely separated.

In Fig. 6.63 is presented a set of three overlapped and transparent isosurfaces of vorticity magnitude, respectively at 5000, 10000 and 150000: in spite of the coarse mesh,

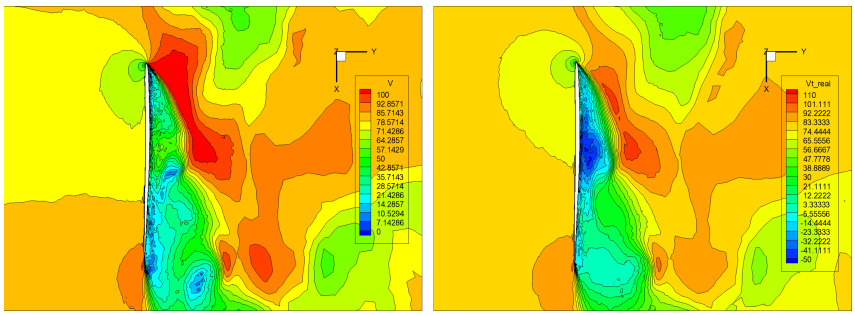

FIGURE 6.61: Average velocity magnitude isocontours (left), average tangential velocity isocontours (right) at a section close to the tip radius of the adaptive blade.

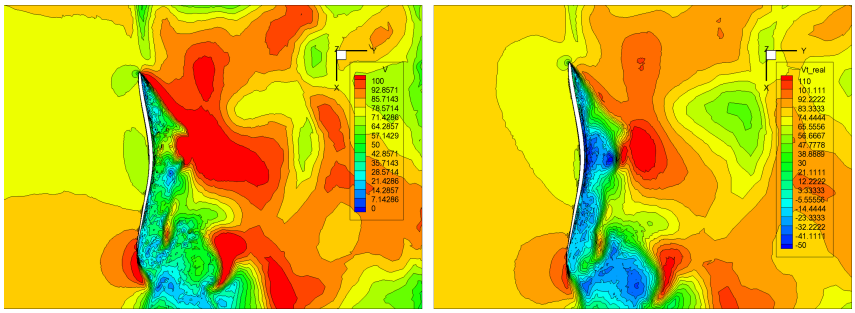

FIGURE 6.62: Instantaneous velocity magnitude isocontours (left), instantaneous tangential velocity isocontours (right) at a section close to the tip radius of the adaptive blade.

complex turbulent structures can be observed in the fluid domain. Multiple vortices are arising from the blade root, from the tip leakage, and from the blade surface at the suction side.

Final remarks

In this section is presented a design of a small Wells turbine with flexible blades, as well as 2D and 3D CFD FSI analysis using an in-house built software. The study begins from the consideration that an a priori imposed curvature on the blade produces different total pressure drop across the blade with respect to the straight blade. In the 2D simulation, in order to observe the dynamic behaviour of a low-stiffness blade when the flow is at nominal condition, five different blade sections are set up, each one with its peculiar material, all constrained with two rigid clamps on the leading edge and trailing edge. The materials chosen for the test cases are different in terms of density and Young's Modulus, and the reinforced polymer Nylon 66/6 is considered as the reference material. A comparison of the CFD results for each blade in terms of total pressure drop is given, and the most promising configuration is selected, as the obtained value of the total pressure drop is almost three times the value obtained with a fixed, straight blade. An alternative design of the adaptive rotor is studied with a 3D simulation: the

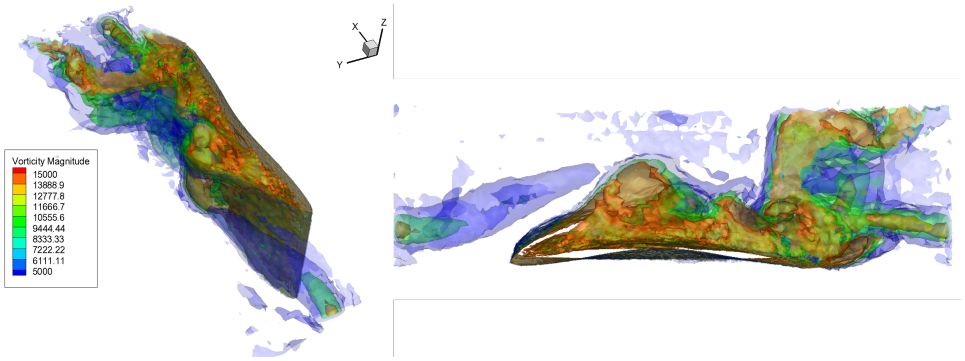

FIGURE 6.63: Isosurfaces of vorticity magnitude for the flow around the adaptive blade.

same blade profile geometry is constrained at the hub by considering fixed the trailing edge, while allowing torsional and bending modes at the leading edge by means of a thin steel bar; different materials were adopted over the chord of the blade to enhance the curvature of the camber. The 3D simulation pointed out the presence of vortices arising from the tip leakage and the intersection between the trailing edge and the hub section, as well as a large recirculation zone on the suction side. The unsteady structural dynamic observed due to the strong coupling suggests that a stable configuration for the adaptive blade can be obtained by further optimizing the adopted materials and by further relaxing the leading edge constraint. With the aid of FSI-CFD simulations, the insurgence of new dynamics effect is highlighted, in terms of vibrations modes rising from the coupling of the fluid dynamics and structural dynamics, thanks to the strong coupling technique of the physical sub-systems implemented in the software. These modes of vibration are entirely excited by the aeroelastic effects. The results are encouraging further studies, and possibly a new design iteration with new constraints and new materials, supported by an optimization process, to select an optimal material and possibly to reach stable configurations with high curvature of the blade.

7. Conclusions

In this work is presented a framework to implement multiphysics fluid structure interaction simulations in Digital Twins and Cyber-Physical systems for turbomachinery, using advanced multiscale methods with finite elements. In general, it is safe to assume that a direct adoption of high fidelity multiphysics simulations inside real time systems such as CPS and DT is not feasible. However, the large amount of high quality, clean and partially preprocessed data available from CFD and FSI software can be used to set up, train and test a set of machine learning algorithm to design a reduced ored model, in order to unlock the potential adoption of the CFD-FSI technology even for the DT and CPS paradigms, in an indirect way. FSI in turbomachinery may appear at a first glance as a small research niche, due to the fact that in general, rotating fluid machinery are usually built with quite rather stiff materials, and consequently the structural displacements of the moving parts are often negligible. Nonetheless there are several possible (canonical and innovative) applications for FSI in turbomachinery that could be modelled as FSI problems, and solved with numerical FSI simulations, such as the analysis of large wind turbines, high aspect ratio fans, passive adaptive rotors (i.e. made with flexible materials), turbomachinery for high density fluids (i.e. tidal turbines). For some of those applications, a strongly coupled FSI simulation would be of great importance to acquire detailed informations about complex aeroelastic effects, generated from the positive feedback between fluid domain and structural domain, and viceversa. In this context, it is of great importance to possess a tool that is able not only to provide reliable results for complex multiphysics problems such as fluid structure interaction in turbomachinery, but that is also completely controllable in every aspect, from the implemented model, to the coupling strategy, and even the memory management. Commercial tools are not often suitable, as they are, to be inserted in a complex research environment where most of the software tools are developed in-house, from different research groups at different times, vastly increasing the effort for interoperability and requiring, each time, time consuming adjustments at the I/O stages. Additionally, it is hard to find software that can be adjusted and tweaked in their core functional aspects, such as the memory management, the portability, the scalability, while still retaining the flexibility to allow the user to implement specific algorithms or solution

strategy, as a strongly coupled FSI solver. For this reason a software for 3D and 2D CFD-FSI unsteady simulations with Finite Elements has been developed from scratch, using a combination of Python and C++, using advanced models such as RBVMS for the Navier-Stokes equations and SEMMT to account for the large mesh displacements. The adopted models have been briefly presented in a concise manner, followed by a description of the most relevant features and characteristics of the software. The developed software, FEMpar, was used to perform several CFD and FSI simulations on different rotating devices, and each application highlighted different peculiar aspect of the adoption of a coupled FSI software at the design stage as well as at the verification stage. The following simulations have been presented: two high aspect ratio axial fans (one 3D printed, the other one with a 10 m diameter), a reversible axial fan with flexible blades, and a Wells turbine with flexible blades, designed to operate in the Mediterranean Sea. From both the simulations of the axial fans were obtained the verification of the performances of the devices, as well as a large and reliable data set for further research. On the other hand, from the simulations of the passive adaptive devices (i.e. the Wells turbine and the reversible axial fan with flexible blades), the software provided precious insights about the design of the constraints, the choice of the materials to build the flexible rotor, as wells as a quantification of the performances for such innovative and still untested designs.

Bibliography

[1] Jianhua Shi et al. "A survey of cyber-physical systems". In: *2011 international conference on wireless communications and signal processing (WCSP)*. IEEE. 2011, pp. 1–6.

[2] Stephan Weyer et al. "Future modeling and simulation of CPS-based factories: an example from the automotive industry". In: *Ifac-Papersonline* 49.31 (2016), pp. 97–102.

[3] Fan Zhang et al. "A novel CPS system for evaluating a neural-machine interface for artificial legs". In: *2011 IEEE/ACM Second International Conference on Cyber-Physical Systems*. IEEE. 2011, pp. 67–76.

[4] Yin Zhang et al. "Health-CPS: Healthcare cyber-physical system assisted by cloud and big data". In: *IEEE Systems Journal* 11.1 (2015), pp. 88–95.

[5] Gang Xiong et al. "Cyber-physical-social system in intelligent transportation". In: *IEEE/CAA Journal of Automatica Sinica* 2.3 (2015), pp. 320–333.

[6] Jay Taneja, Randy Katz, and David Culler. "Defining cps challenges in a sustainable electricity grid". In: *2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*. IEEE. 2012, pp. 119–128.

[7] Edward A Lee. "Cyber-physical systems-are computing foundations adequate". In: *Position paper for NSF workshop on cyber-physical systems: research motivation, techniques and roadmap*. Vol. 2. Citeseer. 2006, pp. 1–9.

[8] Ragunathan Rajkumar et al. "Cyber-physical systems: the next computing revolution". In: *Design automation conference*. IEEE. 2010, pp. 731–736.

[9] RADHA Poovendran. "Cyber–physical systems: Close encounters between two parallel worlds [point of view]". In: *Proceedings of the IEEE* 98.8 (2010), pp. 1363–1366.

[10] Volkan Gunes et al. "A survey on concepts, applications, and challenges in cyber-physical systems." In: *KSII Transactions on Internet & Information Systems* 8.12 (2014).

[11] Paul Kunzemann, Georg Jacobs, and Ralf Schelenz. "Application of CPS within wind energy—current implementation and future potential". In: *Industrial Internet of Things*. Springer, 2017, pp. 647–670.

[12] Michael W Grieves. "Product lifecycle management: the new paradigm for enterprises". In: *International Journal of Product Development* 2.1-2 (2005), pp. 71–84.

[13] Michael Grieves and John Vickers. "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems". In: *Transdisciplinary perspectives on complex systems*. Springer, 2017, pp. 85–113.

[14] Eric J Tuegel et al. "Reengineering aircraft structural life prediction using a digital twin". In: *International Journal of Aerospace Engineering* 2011 (2011).

[15] Edward Glaessgen and David Stargel. "The digital twin paradigm for future NASA and US Air Force vehicles". In: *53rd AIAA/ASME/ASCE/AHS/ASC structures, structural dynamics and materials conference 20th AIAA/ASME/AHS adaptive structures conference 14th AIAA*. 2012, p. 1818.

[16] David Jones et al. "Characterising the Digital Twin: A systematic literature review". In: *CIRP Journal of Manufacturing Science and Technology* 29 (2020), pp. 36–52.

[17] Rong Xie et al. "Digital Twin Technologies for Turbomachinery in a Life Cycle Perspective: A Review". In: *Sustainability* 13.5 (2021), p. 2495.

[18] Alessandro Baldassarre et al. "Towards a digital twin realization of the blade system design study wind turbine blade". In: *Wind and Structures* 28.5 (2019), pp. 271–284.

[19] Iris Graessler and Alexander Poehler. "Intelligent control of an assembly station by integration of a digital twin for employees into the decentralized control system". In: *Procedia Manufacturing* 24 (2018), pp. 185–189.

[20] Banavara R Seshadri and Thiagarajan Krishnamurthy. "Structural health management of damaged aircraft structures using digital twin concept". In: *25th aiaa/ahs adaptive structures conference*. 2017, p. 1675.

[21] Philippe Spalart and Steven Allmaras. "A one-equation turbulence model for aerodynamic flows". In: *30th aerospace sciences meeting and exhibit*. 1992, p. 439.

[22] Brian Edward Launder and Bahrat I Sharma. "Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc". In: *Letters in heat and mass transfer* 1.2 (1974), pp. 131–137.

[23] Florian R Menter. "Improved two-equation k-omega turbulence models for aerodynamic flows". In: *Nasa Sti/recon Technical Report N* 93 (1992), p. 22809.

[24] Georgi Kalitzin et al. "Near-wall behavior of RANS turbulence models and implications for wall functions". In: *Journal of Computational Physics* 204.1 (2005), pp. 265–291.

[25] Kenji Takizawa et al. "Space–time techniques for computational aerodynamics modeling of flapping wings of an actual locust". In: *Computational Mechanics* 50.6 (2012), pp. 743–760.

[26] Kenji Takizawa, Timothy Spielman, and Tayfun E Tezduyar. "Space–time FSI modeling and dynamical analysis of spacecraft parachutes and parachute clusters". In: *Computational Mechanics* 48.3 (2011), pp. 345–364.

[27] Tayfun E Tezduyar et al. "Arterial fluid mechanics modeling with the stabilized space–time fluid–structure interaction technique". In: *International Journal for Numerical Methods in Fluids* 57.5 (2008), pp. 601–629.

[28] Y Bazilevs et al. "3D simulation of wind turbine rotors at full scale. Part II: Fluid–structure interaction modeling with composite blades". In: *International Journal for numerical methods in fluids* 65.1-3 (2011), pp. 236–253.

[29] Alessio Castorrini et al. "A stabilized ALE method for computational fluid–structure interaction analysis of passive morphing in turbomachinery". In: *Mathematical Models and Methods in Applied Sciences* 29.05 (2019), pp. 967–994.

[30] Valerio Francesco Barnabei et al. "FSI analysis and simulation of flexible blades in a Wells turbine for wave energy conversion". In: *E3S Web of Conferences*. Vol. 197. EDP Sciences. 2020, p. 11008.

[31] A Castorrini et al. "Strongly Coupled Fluid-Structure Interaction Simulation of a 3D Printed Fan Rotor". In: *Turbo Expo: Power for Land, Sea, and Air*. Vol. 58547. American Society of Mechanical Engineers. 2019, V001T09A006.

[32] Alessio Castorrini et al. "Computational analysis of wind-turbine blade rain erosion". In: *Computers & Fluids* 141 (2016), pp. 175–183.

[33] Yuri Bazilevs, Kenji Takizawa, and Tayfun E Tezduyar. *Computational fluid-structure interaction: methods and applications*. John Wiley & Sons, 2013.

[34] Vladimir Cherkassky and Filip M Mulier. *Learning from data: concepts, theory, and methods*. John Wiley & Sons, 2007.

[35] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. "Machine learning for fluid mechanics". In: *Annual Review of Fluid Mechanics* 52 (2020), pp. 477–508.

[36] Clarence W Rowley and Scott TM Dawson. "Model reduction for flow analysis and control". In: *Annual Review of Fluid Mechanics* 49 (2017), pp. 387–417.

[37] Isaac E Lagaris, Aristidis Likas, and Dimitrios I Fotiadis. "Artificial neural networks for solving ordinary and partial differential equations". In: *IEEE transactions on neural networks* 9.5 (1998), pp. 987–1000.

[38] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations". In: *Journal of Computational Physics* 378 (2019), pp. 686–707.

[39] Jonathan Tompson et al. "Accelerating eulerian fluid simulation with convolutional networks". In: *International Conference on Machine Learning*. PMLR. 2017, pp. 3424–3433.

[40] Siddhartha Mishra. "A machine learning framework for data driven acceleration of computations of differential equations". In: *arXiv preprint arXiv:1807.09519* (2018).

[41] Hatem Ghorbel et al. "Smart adaptive run parameterization (SArP): enhancement of user manual selection of running parameters in fluid dynamic simulations using bio-inspired and machine-learning techniques". In: *Soft Computing* 23.22 (2019), pp. 12031–12047.

[42] Julia Ling. "Using machine learning to understand and mitigate model form uncertainty in turbulence models". In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2015, pp. 813–818.

[43] Yang Liu et al. "Data-driven modeling for boiling heat transfer: using deep neural networks and high-fidelity simulation results". In: *Applied Thermal Engineering* 144 (2018), pp. 305–320.

[44] Thomas Duriez, Steven L Brunton, and Bernd R Noack. *Machine learning control-taming nonlinear dynamics and turbulence*. Springer, 2017.

[45] Lorenzo Tieghi et al. "Assessment of a Machine-Learnt Adaptive Wall-Function in a Compressor Cascade With Sinusoidal Leading Edge". In: *ASME Turbo Expo 2019: Turbomachinery Technical Conference and Exposition*. American Society of Mechanical Engineers Digital Collection. 2019.

[46] Gino Angelini et al. "A meta-model for aerodynamic properties of a reversible profile in cascade with variable stagger and solidity". In: *ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition*. American Society of Mechanical Engineers Digital Collection. 2018.

[47] Gino Angelini et al. "On surrogate-based optimization of truly reversible blade profiles for axial fans". In: *Designs* 2.2 (2018), p. 19.

[48] Gino Angelini, Alessandro Corsini, and Sergio Lavagnoli. "Machine-learnt topology of complex tip geometries in gas turbine rotors". In: *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy* (2020), p. 0957650920948413.

[49] AA Javadi, TP Tan, and M Zhang. "Neural network for constitutive modelling in finite element analysis". In: *Computer Assisted Mechanics and Engineering Sciences* 10.4 (2003), pp. 523–530.

[50] YMA Hashash, S Jung, and J Ghaboussi. "Numerical implementation of a neural network based material model in finite element analysis". In: *International Journal for numerical methods in engineering* 59.7 (2004), pp. 989–1005.

[51] Lu Trong Khiem Nguyen and Marc-André Keip. "A data-driven approach to nonlinear elasticity". In: *Computers & Structures* 194 (2018), pp. 97–115.

[52] Chiara Bisagni and Luca Lanzi. "Post-buckling optimisation of composite stiffened panels using neural networks". In: *Composite Structures* 58.2 (2002), pp. 237–247.

[53] W Ruijter et al. "Optimization of composite panels using neural networks and genetic algorithms". In: *Computational Fluid and Solid Mechanics 2003*. Elsevier, 2003, pp. 2359–2363.

[54] Luca Lanzi, Chiara Bisagni, and Sergio Ricci. "Neural network systems to reproduce crash behavior of structural components". In: *Computers & Structures* 82.1 (2004), pp. 93–108.

[55] MA Bessa and S Pellegrino. "Design of ultra-thin shell structures in the stochastic post-buckling range using Bayesian machine learning and optimization". In: *International Journal of Solids and Structures* 139 (2018), pp. 174–188.

[56] Robert Ian Levin and NAJ Lieven. "Dynamic finite element model updating using neural networks". In: *Journal of Sound and Vibration* 210.5 (1998), pp. 593–607.

[57] Liang Liang et al. "A deep learning approach to estimate stress distribution: a fast and accurate surrogate of finite-element analysis". In: *Journal of The Royal Society Interface* 15.138 (2018), p. 20170844.

[58] Roberto Molinaro et al. "Embedding data analytics and CFD into the digital twin concept". In: *Computers & Fluids* 214 (2021), p. 104759.

[59] Alessio Castorrini et al. "Numerical Testing of a Trailing Edge Passive Morphing Control for Large Axial Fan Blades". In: *Journal of Engineering for Gas Turbines and Power* 140.3 (2018), p. 032606.

[60] Raymond W Ogden. *Non-linear elastic deformations*. Courier Corporation, 1997.

[61] Thomas JR Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.

[62] Bruce M Irons. "Engineering applications of numerical integration in stiffness methods." In: *AIAA journal* 4.11 (1966), pp. 2035–2037.

[63] J Donea, J-Ph Ponthot, and Antonio Rodriguez Ferran. *Encyclopedia of computational mechanics*. 2004.

[64] Lawrence E Malvern. *Introduction to the Mechanics of a Continuous Medium*. Monograph. 1969.

[65] Thomas JR Hughes. "Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods". In: *Computer methods in applied mechanics and engineering* 127.1-4 (1995), pp. 387–401.

[66] Thomas JR Hughes et al. "The variational multiscale method—a paradigm for computational mechanics". In: *Computer methods in applied mechanics and engineering* 166.1-2 (1998), pp. 3–24.

[67] Thomas JR Hughes, Luca Mazzei, and Kenneth E Jansen. "Large eddy simulation and the variational multiscale method". In: *Computing and visualization in science* 3.1 (2000), pp. 47–59.

[68] Thomas JR Hughes, Assad A Oberai, and Luca Mazzei. "Large eddy simulation of turbulent channel flows by the variational multiscale method". In: *Physics of fluids* 13.6 (2001), pp. 1784–1799.

[69] Thomas JR Hughes et al. "The multiscale formulation of large eddy simulation: decay of homogeneous isotropic turbulence". In: *Physics of fluids* 13.2 (2001), pp. 505–512.

[70] Thomas JR Hughes, Garth N Wells, and Alan A Wray. "Energy transfers and spectral eddy viscosity in large-eddy simulations of homogeneous isotropic turbulence: Comparison of dynamic Smagorinsky and multiscale models over a range of discretizations". In: *Physics of Fluids* 16.11 (2004), pp. 4044–4052.

[71] Y Bazilevs et al. "Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows". In: *Computer methods in applied mechanics and engineering* 197.1-4 (2007), pp. 173–201.

[72] Thomas JR Hughes and Giancarlo Sangalli. "Variational multiscale analysis: the fine-scale Green's function, projection, optimization, localization, and stabilized methods". In: *SIAM Journal on Numerical Analysis* 45.2 (2007), pp. 539–557.

[73] Alexander N Brooks and Thomas JR Hughes. "Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations". In: *Computer methods in applied mechanics and engineering* 32.1-3 (1982), pp. 199–259.

[74] Thomas JR Hughes and TE Tezduyar. "Finite element methods for first-order hyperbolic systems with particular emphasis on the compressible Euler equations". In: *Computer methods in applied mechanics and engineering* 45.1-3 (1984), pp. 217–284.

[75] Tayfun E Tezduyar and YJ Park. "Discontinuity-capturing finite element formulations for nonlinear convection-diffusion-reaction equations". In: *Computer methods in applied mechanics and engineering* 59.3 (1986), pp. 307–325.

[76] Thomas JR Hughes, Leopoldo P Franca, and Marc Balestra. "A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: A stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations". In: *Computer Methods in Applied Mechanics and Engineering* 59.1 (1986), pp. 85–99.

[77] Tayfun E Tezduyar and Yasuo Osawa. "Finite element stabilization parameters computed from element matrices and vectors". In: *Computer Methods in Applied Mechanics and Engineering* 190.3-4 (2000), pp. 411–430.

[78] Tayfun E Tezduyar. "Computation of moving boundaries and interfaces and stabilization parameters". In: *International Journal for Numerical Methods in Fluids* 43.5 (2003), pp. 555–575.

[79] Claes Johnson. *Numerical solution of partial differential equations by the finite element method*. Courier Corporation, 2012.

[80] Yuri Bazilevs and Thomas JR Hughes. "Weak imposition of Dirichlet boundary conditions in fluid mechanics". In: *Computers & Fluids* 36.1 (2007), pp. 12–26.

[81] Klaus-Jürgen Bathe. *Finite element procedures*. Klaus-Jurgen Bathe, 2006.

[82] Andrew A Johnson and Tayfun E Tezduyar. "Mesh update strategies in parallel finite element computations of flow problems with moving boundaries and interfaces". In: *Computer methods in applied mechanics and engineering* 119.1-2 (1994), pp. 73–94.

[83] Tayfun E Tezduyar et al. "Computation of unsteady incompressible flows with the stabilized finite element methods: space-time formulations, iterative strategies and massively parallel implementations". In: *ASME PRESSURE VESSELS PIPING DIV PUBL PVP., ASME, NEW YORK, NY(USA), 1992*, 246 (1992), pp. 7–24.

[84] Tayfun Tezduyar et al. "Parallel finite-element computation of 3D flows". In: *Computer* 26.10 (1993), pp. 27–36.

[85] John R Dormand and Peter J Prince. "A family of embedded Runge-Kutta formulae". In: *Journal of computational and applied mathematics* 6.1 (1980), pp. 19–26.

[86] Nathan M Newmark. "A method of computation for structural dynamics". In: *Journal of the engineering mechanics division* 85.3 (1959), pp. 67–94.

[87] Jintai Chung and GM Hulbert. "A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method". In: (1993).

[88] Kenneth E Jansen, Christian H Whiting, and Gregory M Hulbert. "A generalized-α method for integrating the filtered Navier–Stokes equations with a stabilized finite element method". In: *Computer methods in applied mechanics and engineering* 190.3-4 (2000), pp. 305–319.

[89] Yuri Bazilevs et al. "Isogeometric fluid-structure interaction: theory, algorithms, and computations". In: *Computational mechanics* 43.1 (2008), pp. 3–37.

[90] Yuri Bazilevs et al. "ALE-VMS and ST-VMS methods for computer modeling of wind-turbine rotor aerodynamics and fluid–structure interaction". In: *Mathematical Models and Methods in Applied Sciences* 22.supp02 (2012), p. 1230002.

[91] Kenji Takizawa et al. "Space–time fluid mechanics computation of heart valve models". In: *Computational Mechanics* 54.4 (2014), pp. 973–986.

[92] Ruofu XIAO, Zhengwei WANG, and Yongyao LUO. "Stress analysis of Francis turbine runners based on FSI [J]". In: *Journal of Hydroelectric Engineering* 3 (2007), p. 75.

[93] Atsushi Tateishi et al. "Multimode flutter analysis of transonic fan using FSI simulation". In: *ASME Turbo Expo 2014: Turbine Technical Conference and Exposition*. American Society of Mechanical Engineers Digital Collection. 2014.

[94] Lucio Cardillo et al. "Flow analysis of a wave-energy air turbine with the SUPG/PSPG stabilization and discontinuity-capturing directional dissipation". In: *Computers & Fluids* 141 (2016), pp. 184–190.

[95] P. Venturini et al. "Modeling of multiphase combustion and deposit formation in a biomass-fed furnace". In: *Energy* 35.7 (2010), pp. 3008–3021. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2010.03.038. URL: https://www.sciencedirect.com/science/article/pii/S0360544210001672.

[96] Alessio Castorrini et al. “Computational analysis of particle-laden-airflow erosion and experimental verification”. In: *Computational Mechanics* (2020), pp. 1–17.

[97] Alessio Castorrini et al. “Simulation of the deposit evolution on a fan blade for tunnel ventilation”. In: *Journal of Engineering for Gas Turbines and Power* 142.4 (2020).

[98] Benjamin S Kirk et al. “libMesh: a C++ library for parallel adaptive mesh refinement/coarsening simulations”. In: *Engineering with Computers* 22.3-4 (2006), pp. 237–254.

[99] Valerio Francesco Barnabei et al. “Unsteady Flow Simulation of an Axial Fan for Dry Cooling in a CSP Plant Using the Variational Multiscale Method”. In: *ASME Turbo Expo 2020: Turbomachinery Technical Conference and Exposition*. American Society of Mechanical Engineers Digital Collection. 2020.

[100] Satish Kumar Chimakurthi et al. “ANSYS Workbench System Coupling: a state-of-the-art computational framework for analyzing multiphysics problems”. In: *engineering with Computers* 34.2 (2018), pp. 385–411.

[101] S CD-adapco. “STAR CCM+ user guide version 12.04”. In: *CD-Adapco: New York, NY, USA* (2017).

[102] Int NUMECA. “NUMECA’s flow integrated environment for turbomachinery and internal flows”. In: *User Manual, Numeca Int., Brussels, Belgium* (2000).

[103] Hrvoje Jasak, Aleksandar Jemcov, Zeljko Tukovic, et al. “OpenFOAM: A C++ library for complex physics simulations”. In: *International workshop on coupled methods in numerical dynamics*. Vol. 1000. IUC Dubrovnik Croatia. 2007, pp. 1–20.

[104] Wes McKinney. *Python for data analysis: Data wrangling with Pandas, NumPy, and IPython*. " O’Reilly Media, Inc.", 2012.

[105] George Karypis. “METIS and ParMETIS”. In: *Encyclopedia of parallel computing* (2011), pp. 1117–1124.

[106] Satish Balay et al. “PETSc users manual”. In: (2019).

[107] Matteo Bernardini, Sergio Pirozzoli, and Paolo Orlandi. “Velocity statistics in turbulent channel flow up to”. In: *Journal of Fluid Mechanics* 742 (2014), pp. 171–191.

[108] A Castorrini et al. “Morphing of reversible axial fan blade: a FSI-FEM study”. In: *ASME Turbo Expo 2021*. American Society of Mechanical Engineers. 2021, GT2021–59227.

[109] Shady Farah, Daniel G Anderson, and Robert Langer. "Physical and mechanical properties of PLA, and their functions in widespread applications - A comprehensive review". In: *Advanced drug delivery reviews* 107 (2016), pp. 367–392.

[110] A Castorrini et al. "Numerical study on the passive control of the aeroelastic response in large axial fans". In: *ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition*. American Society of Mechanical Engineers. 2016, V001T09A010–V001T09A010.

[111] Stefan Turek and Jaroslav Hron. "Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow". In: *Fluid-structure interaction*. Springer, 2006, pp. 371–385.

[112] Detlev G Kröger. *Air-cooled heat exchangers and cooling towers*. Vol. 1. PennWell Books, 2004.

[113] http://www.minwatercsp.eu/. *MinWaterCSP EU Project*. 2016. URL: `http://www.minwatercsp.eu/` (visited on 11/25/2019).

[114] Michael B Wilkinson, Johan Van Der Spuy, and Theodor W von Backström. "The design of a large diameter axial flow fan for air-cooled heat exchanger applications". In: *ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition*. American Society of Mechanical Engineers Digital Collection. 2017.

[115] Michael B Wilkinson, Sybrand J van der Spuy, and Theodor W von Backström. "Performance Testing of an Axial Flow Fan Designed for Air-Cooled Heat Exchanger Applications". In: *Journal of Engineering for Gas Turbines and Power* 141.5 (2019).

[116] C Van Niekerk. "Ducted fan design theory". In: *ASME J. Appl. Mech* 25.3 (1958), pp. 325–331.

[117] Mark Drela. "XFOIL: An analysis and design system for low Reynolds number airfoils". In: *Low Reynolds number aerodynamics*. Springer, 1989, pp. 1–12.

[118] Tommaso Bonanni et al. "Design of a single stage variable pitch axial fan". In: *ASME Turbo Expo 2017: Turbomachinery Technical Conference and Exposition*. American Society of Mechanical Engineers Digital Collection. 2017.

[119] David Volponi et al. "CFD SIMULATION RESULTS FOR THE MINWATERCSP COOLING FAN". In: *Fan 2018 – International conference on fan noise, aerodynamics, applications and systems*. May 2018.

[120] Seymour Lieblein, Francis C Schwenk, and Robert L Broderick. *Diffusion factor for estimating losses and limiting blade loadings in axial-flow-compressor blade elements*. Tech. rep. NATIONAL ADVISORY COMMITTEE FOR AERONAUTICS CLEVELAND OH LEWIS FLIGHT . . ., 1953.

[121] B. Lakshminarayana. "Methods of Predicting the Tip Clearance Effects in Axial Flow Turbomachinery". In: *Journal of Basic Engineering* 92.3 (Sept. 1970), pp. 467–480. ISSN: 0021-9223. DOI: 10.1115/1.3425036. eprint: https://asmedigitalcollection.asme.org/fluidsengineering/article-pdf/92/3/467/5657696/467_1.pdf. URL: https://doi.org/10.1115/1.3425036.

[122] Nicholas A Cumpsty. *Compressor aerodynamics*. Longman Scientific & Technical, 1989.

[123] AG Sheard and K Daneshkhah. "The conceptual design of high pressure reversible axial tunnel ventilation fans". In: *Advances in Acoustics and Vibration* 2012 (2012).

[124] AN Bolton. "Installation effects in fan systems". In: *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy* 204.3 (1990), pp. 201–215.

[125] Lucio Cardillo et al. "A numerical investigation into the aerodynamic effect of pressure pulses on a tunnel ventilation fan". In: *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy* 228.3 (2014), pp. 285–299.

[126] William Cory. *Fans and ventilation: a practical guide*. Elsevier, 2010.

[127] Kellis C Kincaid and David W MacPhee. "Numerical Fluid–Structure Interaction Analysis of a Wells Turbine With Flexible Blades". In: *Journal of Energy Resources Technology* 142.8 (2020).

[128] Sybrand Johannes Van der Spuy. "The design of a low-noise rotor-only axial flow fan series". PhD thesis. Stellenbosch: University of Stellenbosch, 1997.

[129] A Corsini et al. "Numerical study on active and passive trailing edge morphing applied to a multi-MW wind turbine section". In: *Sixth International Conference on Computational Methods in Marine Engineering (MARINE), Rome, Italy, June*. 2015, pp. 15–17.

[130] International Energy Agency (IEA). "Medium Term Renewable Energy Market Report 2014—Market Analysis and Forecasts to 2020". In: IEA Paris. 2014.

[131] Anna Galkina, Vyacheslav Kulagin, and Irina Mironova. "Renewable Energy Sources: Global and Russian Outlook Up to 2040". In: *Journal of Technology Innovations in Renewable Energy* 3.4 (2014), pp. 185–194.

[132] Ahmed S Shehata et al. "Wells turbine for wave energy conversion: a review". In: *International journal of energy research* 41.1 (2017), pp. 6–38.

[133] M Bassetti et al. "Modelling and transient simulation of an OWC wave energy converters tailored to the Mediterranean shore-line". In: *Fourth International Symposium "Monitoring of Mediterranean coastal areas problems and measurements techniques*. 2012.

[134] M Bassetti et al. "Design and verification of a micro Wells turbine for Mediterranean operations". In: *11 th European Conference on Turbomachinery Fluid dynamics & Thermodynamics*. EUROPEAN TURBOMACHINERY SOCIETY. 2015.

[135] Robert J Palmer. "Polyamides, plastics". In: *Kirk-Othmer Encyclopedia of Chemical Technology* (2000).