

Pablo Mosquera Michaelsen

**A Fluid-Structure-Interaction
Simulation tool for application
in rotating machinery**

Karlsruhe 2015



This document is licensed under the Creative Commons Attribution 3.0 DE License
(CC BY 3.0 DE): <http://creativecommons.org/licenses/by/3.0/de/>

*To God, for calling me into existence;
to my parents, for making me the one
I am; to my wife, for teaching me the
true love.*

**A Fluid-Structure-Interaction simulation tool
for application in rotating machinery**

Zur Erlangung des akademischen Grades
Doktor der Ingenieurwissenschaften
der Fakultät für Maschinenbau
Karlsruher Institut für Technologie (KIT)

genehmigte
Dissertation

von

Ing. Pablo Mosquera Michaelsen
aus Buenos Aires (Argentinien)

Tag der mündlichen Prüfung: 12. Februar 2015
Hauptreferent: Prof. Dr.-Ing. Martin Gabi
Korreferent: Prof. Dr.-Ing. Carsten Proppe

Zusammenfassung

Moderne Strömungsmaschinen weisen strömungsinduzierte Vibrationen auf, deren Minderung eine große Herausforderung für das Design der Maschine darstellt. Um diese zu meistern, können hochaufgelöste, dreidimensionale, gekoppelte CFD/CSD-Simulationen eingesetzt werden. Kommerzielle Softwarepakete unterliegen für den Fall, dass große Modelle berechnet oder spezielle Erweiterungen aus der Forschung implementiert werden müssen, einigen Einschränkungen. Um diese Nachteile zu umgehen, wurde ein Tool zur Fluid-Struktur-Interaktion auf Basis des in-house CFD-Codes *SPARC*, des quelloffenen CSD-Lösers *CalculiX* sowie des eigen entworfenen Kopplungsmanagers *FSIM* entwickelt. Obwohl bereits existierende Löser in der Regel nicht speziell für gekoppelte Anwendungen konzipiert wurden, ist es mit Hilfe des in dieser Arbeit präsentierten Tools möglich, von der Funktionalität und Leistungsfähigkeit der in vielen Anwendungen bewährten und hochspezialisierten Löser zu profitieren. Das Tool kann darüber hinaus flexibel um andere Anwendungsbereiche wie z.B. dem Wärmeübergang erweitern werden. Die neue Entwicklung nutzt die jeweiligen Vorteile blockstrukturierter Netze im Bezug auf das Mehrgitterverfahren, der transfiniten Netzanpassung, einer blockweisen Netzanpassungskontrolle sowie eines effizienten Volumen- zu-Oberflächen-Mappings. Im Rahmen der Verifizierung wurden jeweils ein spezifischer Testfall für das Phänomen des Flatterns sowie der erzwungenen Oszillation analysiert. Schließlich wurde eine geeignete Formulierung für die Kompatibilität rotierender Fluid- und Strukturmodelle entwickelt und anhand eines einfachen Beispiels das Potenzial des neuen Tools zur Untersuchung von fluidinduzierten Vibrationen in rotierenden Systemen diskutiert.

Abstract

Modern turbomachinery exhibit flow-induced vibrations, whose reduction has become a challenging design issue. To cope with it, high-resolution three-dimensional, coupled CFD/CSD simulations can be employed. Commercial simulation software packages show some limitations when the computation of very large models is intended or special extensions resulting from research are required. To overcome them, a tool for Fluid-Structure-Interaction simulation was developed on the basis of the in-house CFD code *SPARC*, the open-source CSD solver *CalculiX* and the specially designed coupling manager *FSiM*. Although already existing solvers have usually not been conceived for coupled use, the tool presented in this work makes it possible to profit from the functionality and capability of well-proven, highly specialized solvers. The tool is therefore flexible and can be easily expanded to account for other applications such as coupled heat transfer. The new development exploits the advantages of block-structured meshes related to the multigrid acceleration technique, the transfinite mesh adaption method, a block-wise mesh adaption control and an efficient volume to surface mapping. A verification has been performed for tests cases representing flutter and forced oscillations. Finally, the appropriate formulations for the compatibility of the fluid and structural rotating models have been developed, and the capabilities of the new tool for the investigation of fluid-induced vibrations in rotating machinery are discussed on a simple test case.

Acknowledgments

This doctoral thesis is the result of my work at the Institute of Fluid Machinery (FSM) at the Karlsruhe Institute of Technology, from October 2009 to March 2014. I would like to express my deepest thanks to Prof. Dr.-Ing. Martin Gabi for making it possible to do my doctoral studies at FSM. Without his continuous academic and financial support this work would not have been possible. Also my acknowledgement to Prof. Dr.-Ing. Carsten Proppe for assuming the coreference of my thesis. I also feel greatly indebted to the German Academic Exchange Service (DAAD) for the financial support of the first years. My warm gratitude to all of my colleagues at FSM for the great time spent working together, especially to Dr.-Ing. Bálazs Pritz for the technical tutoring of my work, and to Benjamin Jastrow and Florian Trimborn for the many constructive discussions and their support with the German language. An additional special thanks to Christian Glück for his enthusiastic support with computer-related issues and to Dr.-Ing. Iris Pantle for her initial tutoring and the correction of this work. I would also like to thank my parents and family in Argentina for their encouragement and support in various forms. To my wife Gisela my earnest and heartfelt gratitude for her enormous patience and often silent but indispensable support, which motivated and strengthened me to overcome every obstacle.

Essen, in September 2015

Pablo Mosquera Michaelsen

Nomenclature

Latin symbols

b	Volume forces
C	Constitutive tensor
<i>C</i>	Damping coefficient
<i>c</i>	Convective velocity
d	Node displacement
<i>e</i>	Internal energy
<i>E</i>	Total energy, Green-Lagrange tensor
f	Node force
<i>f</i>	Frequency
F	Deformation gradient tensor, exciting force
<i>I</i>	Moment of inertia
<i>K</i>	Stiffness coefficient
<i>M</i>	Torque
n	Surface unit normal vector
<i>p</i>	Pressure
<i>q</i>	Heat flow
<i>R</i>	Gas constant
S	Area vector, second Piola-Kirchhoff tensor
<i>t</i>	Time
<i>T</i>	Temperature
u	Velocity
<i>U</i>	Reference flow velocity
<i>w</i>	Weighing factors for interpolation
W	Fluid solution vector
x	Node coordinates

Greek symbols

δ	Decay rate
η	Normalized exciting frequency
γ	Isentropic exponent
Γ	Interface between fluid and structural domains
μ	Laminar viscosity
λ	Tip-speed ratio
ω	Rotational speed
ω_0	Natural angular frequency
Ω	Oscillation angular frequency, fluid domain interior
ϕ	Rotation angle
ρ	Density
σ	Stress tensor
τ	Deviatoric stress tensor
θ	Torsional displacement
ϑ	Damping degree
ζ	Phase angle

Superscripts

f	Fluid
s	Structural
0	Reference configuration

Abbreviations

ALE	Arbitrary Lagrangian Eulerian
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CSD	Computational Structural Dynamics
DOF	Degree of freedom
FEM	Finite Element Method
FSI	Fluid-Structure Interaction
FSM	Institute of Fluid Machinery (Fachgebiet Strömungsmaschinen)
FVM	Finite Volumes Method
IBM	Immersed Boundary Method
Int. Surf.	Interaction Surface
PDE	Partial Differential Equation
PIV	Particle Image Velocimetry

Contents

- Abstract **vii**
- Acknowledgments **ix**
- Nomenclature **xi**
- List of Figures **xiv**
- List of Tables **xvi**
- 1 Introduction 1**
- 2 Theoretical Basis 3**
 - 2.1 Introduction 3
 - 2.2 Physics of flow-induced vibrations 3
 - 2.2.1 Aeroelastic instabilities 3
 - 2.2.2 Forced oscillations 5
 - 2.3 Flow dynamics 6
 - 2.3.1 Physical model 6
 - 2.3.2 Discretization of the ALE equations 9
 - 2.3.3 Numerical solution 11
 - 2.3.4 Rotating models 12
 - 2.4 Structural dynamics 12
 - 2.4.1 Nonlinear dynamic analysis 12
 - 2.4.2 Rotating models 14
 - 2.5 Fluid-Structure Interaction 14
 - 2.5.1 Classification of FSI approaches 15
 - 2.5.2 Partitioned approaches 17
 - 2.5.3 Loosely coupled schemes 18
 - 2.5.4 Strongly coupled schemes 20
 - 2.5.5 Interface treatment 21
 - 2.5.6 Other considerations 22
- 3 Development of the coupling tool 23**
 - 3.1 Introduction 23
 - 3.1.1 Alternatives for the coupling of existing codes 23
 - 3.1.2 Structural solver 24
 - 3.2 General approach in FSIM 25

3.2.1	Multigrid	27
3.3	Implementation in MPI	28
3.4	Mesh adaption	30
3.5	Interface treatment	32
3.5.1	Neighbour search	34
3.5.2	Interpolation	37
3.5.3	Displacement interpolation	40
3.6	Rotating models	40
3.6.1	Measurement of moments	46
4	Verification and performance of FSiM	49
4.1	Alternatives for benchmarking	49
4.2	Panel flutter problem	50
4.3	Flexible obstacle	52
5	Application to rotating machinery	61
6	Conclusions and outlook	69
6.1	Conclusions	69
6.2	Outlook	70
	Bibliography	71
	Curriculum Vitae	81

List of Figures

2.1	Single DOF flutter system. Taken from [1]	3
2.2	Single DOF oscillatory system. Taken from [2].	5
2.3	Response amplitude and phase of the single DOF system. Taken from [2].	6
2.4	Numbering scheme of structured and unstructured meshes.	7
2.5	Different types of structured meshes. Taken from [3].	8
2.6	The Full Multigrid scheme.	11
2.7	Reference and material configurations.	13
2.8	The Conventional Sequential Staggered (CSS) procedure.	19
2.9	The Conventional Parallel Staggered (CPS) procedure.	20
3.1	Example case. Left: Geometry. Right: block topology and structural mesh.	25
3.2	Mapping from volume to surface structured mesh.	26
3.3	Surface interaction pairs. Left: In fluid. Right: In structure	27
3.4	Face-numbering convention for hexahedron elements.	27
3.5	Sequence of an FSI computation with the CSS procedure.	28
3.6	Block generations. 1 st gen.: yellow; 2 nd gen.: green; 3 rd gen.: blue; 4 th gen.: red	31
3.7	Simple mesh deformation treatment. Only the first block layer absorbs the interface movement.	32
3.8	Improved mesh deformation treatment. Boundary layer resolution is preserved.	32
3.9	Mesh deformation treatment for a complex blade topology	33
3.10	Mesh deformation treatment for a wind turbine blade.	34
3.11	Propagation of interface displacements across the block structure.	35
3.12	Algorithm to find a point in a triangle. Taken from [4]	36
3.13	Fluid-node-based neighbour search.	37
3.14	Structural-face-midpoint-based neighbour search	38
3.15	Neighbour associations. Linking lines between fluid nodes and structural element-face midpoint.	39
3.16	Structural-face-midpoint-based neighbour search. Linking lines between structural element-face midpoint and fluid cell-face midpoint.	40
3.17	Conservative force interpolation.	41
3.18	Pressure interpolation.	41
3.19	Displacement interpolation.	42
3.20	Block topology and meshes for a rotor	44
3.21	Sequence to handle rotating models	44
3.22	Torsional system for a turbomachine. a) Real system. b) Simplified equivalent system.	46
4.1	Geometry of the panel flutter test case.	50
4.2	Data extraction location for panel deformation.	51

4.3	Comparison of results against the reference data.	52
4.4	Mach number distribution, showing a traveling pressure shock.	53
4.5	Geometry of the flexible obstacle test case.	54
4.6	Simulation domain.	55
4.7	Comparison of tip deflection.	55
4.8	Frequency spectrum of tip deflection.	56
4.9	Pressure contour plot of FSiM.	57
4.10	Pressure contour plot of ADINA.	57
4.11	Velocity contour plot of FSiM.	58
4.12	Velocity contour plot of ADINA.	58
4.13	Comparison of pressure and velocity profiles between FSiM and ADINA.	59
4.14	Comparison of pressure and velocity profiles between FSiM and ADINA. Detail.	59
5.1	Geometry of the application example.	61
5.2	Fixed nodes to model the rotor shaft.	62
5.3	Frequency spectrum of the blade moment.	64
5.4	Computation results at different times.	65
5.5	Influence of rotor flexibility on fluid moment.	66
5.6	Frequency spectrum of fluid moment.	66
5.7	Influence of rotor flexibility on shaft reaction moment.	66
5.8	Time history of excitation and displacement for a blade.	67
5.9	Vibration modes of the rotor.	68

List of Tables

3.1 Nomenclature for fluid and structural meshes. 25

3.2 Resolved physics for the different model complexity. 42

4.1 Properties for the panel flutter problem. 50

4.2 Computational time fraction for each coupling component. 51

4.3 Properties of the beam obstacle. 54

4.4 Computational time performace. 60

4.5 Computational time fraction for each coupling component. 60

5.1 Properties of the application example. 61

5.2 Natural frequencies of the rotor. 63

1 Introduction

Turbomachinery is a key mechanical component of today's society, both for power generation and handling of fluids. In order to design more efficient, cheaper and reliable turbomachinery, a deeper understanding of diverse physical phenomena is required. One of these are flow-induced vibrations, which have a great influence on the life-time and stability of a turbomachine [5]. Challenging issues in this area are found in three important types of turbomachinery: thermal, hydraulic and wind turbines.

In the field of thermal turbomachines, new aggressively lightweight blade designs are pushing current aeroelastic analysis to its limits [6]. Large efforts are being done through research projects such as *FUTURE* (Flutter-Free Turbomachinery Blades) to improve flutter prediction tools. As regards hydraulic turbomachines, new developments of sea-current turbines show that the dominating flow conditions are highly fluctuating [7], inducing vibrations whose amplitude is not negligible in new slender blades for improved hydraulic performance. Other difficulties include hydroelastic instabilities of flows in deforming gaps and the lock-in phenomenon resulting in large amplitude vibrations and fatigue problems [8].

Particularly for wind power, aeroelasticity is becoming an issue, as evidenced by European research projects such as *STABCON*. Indeed, the increasing use of aeroelastic tailoring in innovative blade designs, wherein the blade twists as it bends under the action of aerodynamic loads [9], [10] together with the increasing sizes of projected offshore turbines [11], turn the prevention of flutter a challenging task [12]. As reference, the predicted flutter speed of a MW-sized blade is slightly greater than twice the operational speed of the rotor, while for a smaller rotor with relatively stiff blades the predicted flutter speed is approximately six times the operating speed [13]. Cracks due to edgewise vibrations were observed on stall-regulated rotors as a result of negative aerodynamic damping [14], [15]. Unlike structural damping, which can be measured quite accurately if the blade exists, aerodynamic damping cannot be included as a simple model parameter [16].

Experimental investigations can provide many answers to these issues, but exhibit some shortcomings as well. For example, resonant vibrations cannot be appropriately reproduced in reduced scale setups [7]. For this reason, theoretical methods to analyse flow-induced vibrations have evolved in parallel. Depending on the type of turbomachinery, different approaches can be applied. For example, in the case of gas turbine blades, energy methods [17] and linearized structural dynamics equations [18] have been used, while for wind turbines linearized fluid dynamics models [19] [20] have been preferred. Such methods provide fast and robust means of addressing the vibrations and stability problems, but their accuracy is limited. For example, linear structural models are too conservative and therefore hinder optimization [21], [19]; two-dimensional models cannot reproduce some important effects in wind turbine blades as induced flow [22] and delayed stall [23]; the fluid damping properties can only be accurately reproduced with unsteady two or three-dimensional models [7], [8].

With the growth of computing capacity, the alternative of costly but highly detailed three-dimensional, time-resolved, coupled simulations has regained interest. For example, May et al. [6] and Doi [24] investigated the flutter problem in thermal turbomachinery whereas Lippold [7] and Bazilevs [11]

applied such methods to hydraulic and wind turbines respectively. Particularly this last kind of numerical method is described in the literature with the term *Fluid-Structure Interaction* (FSI).

In principle it would be possible for a research group to create a new computational tool for the special purpose of Fluid-Structure Interaction. It is, however, often more convenient to make use of existent, well-proven software packages, which already include many advanced modeling features. In this perspective, the Institute of Fluid Machinery is developing the Computational Fluid Dynamics (CFD) code *SPARC* (Structured PARallel Research Code) [25] [26] since more than 15 years, and an extension to investigate flow-induced vibrations eventually became relevant. An in-house development such as *SPARC* provides a series of advantages in comparison with commercial simulation tools:

1. Allows the researcher to access and improve any internal routines
2. Avoids the cost of licenses, which effectively limits the number of cores and therefore the size or resolution of the examined models
3. Bestows on the researcher the full control over acceleration and stabilization procedures, which might complicate the separation of numerical from physical effects

These advantages can be extended to Fluid-Structure-Interaction simulations taking advantage of an appropriate Computational Structural Dynamics (CSD) solver and devising a mechanism to synchronize the time solutions and to solve the communication between the two solvers. In the present investigation, the work of some researchers [27], [28] which have dealt with these issues is taken as basis, but the focus is now placed on two additional aspects:

- The development of a tool for Fluid-Structure Interaction that takes advantage of the benefits of block-structured meshes
- The application of Fluid-Structure-Interaction to rotating systems, i.e., for turbomachinery

While the use of existing solvers promises a powerful coupled simulation tool, some implementation issues might be expected due to the fact that existent codes have not been conceived for modular use.

The objective of the present work is therefore, to design a computational tool that provides an efficient and high-accurate solution of Fluid-Structure-Interaction phenomena for turbomachinery, taking advantage of existent, well-proven, non-commercial solvers. In particular, the benefits of the block-structured meshes are to be exploited along the different stages of the coupling process.

This work is outlined in the following way: Chapter 2 presents the theoretical background of flow-induced vibrations, Computational Fluid Dynamics, Computational Structural Dynamics and the methods for Fluid-Structure Interaction. Along the chapter, the state of the art is given to put the chosen methods in context; Chapter 3 exposes the details of the undertaken implementation; Chapter 4 discusses the test cases used to verify the developed tool; in Chapter 5 the capabilities of the new tool are presented with an example; in Chapter 6, finally, conclusions are drawn and outlook is given.

2 Theoretical Basis

2.1 Introduction

Real-life physics is complex by its nature. An engineer's task is to identify the decisive aspects and select the simplest mathematical models that describe those aspects accurately enough. In this chapter, the theoretical basis for the study of flow-induced vibrations in turbomachinery is presented. In the first section, the behavior of oscillating systems with concentrated parameters is discussed. Then the mathematical models for the accurate solution of three-dimensional flow and structural dynamics are exposed. The last part of the chapter deals with the solution of the general coupled fluid-structural problem.

2.2 Physics of flow-induced vibrations

When studying flow-induced vibrations, two dangerous mechanisms can be clearly identified: aeroelastic instabilities and forced oscillations. As a starting point, the mathematical models that rule such behaviors are exposed, in order to acquire a first understanding with the help of simple cases.

2.2.1 Aeroelastic instabilities

Aeroelastic instabilities are the result of a highly coupled fluid-structure system and are potentially very dangerous. The basic mechanism is illustrated for the particular case of classical flutter. In classical flutter, a coupling between the bending and torsional degrees of freedom of a turbomachinery blade or airplane's wing lead to vibrations. This happens because the torsional motion is associated with a change in angle of attack and consequently of aerodynamic forces and, simultaneously, the motion is associated with elastic deformations of the blade structure, leading to an oscillating system. For small angles of attack, this is a linear phenomenon since the flow surrounding the blade is attached, and it can be then adequately described with non-viscous formulations [15]. Every airfoil has a flutter boundary at some speed; for wind turbines, for example, the boundary is defined as the rotational speed at which the blade will flutter [10].

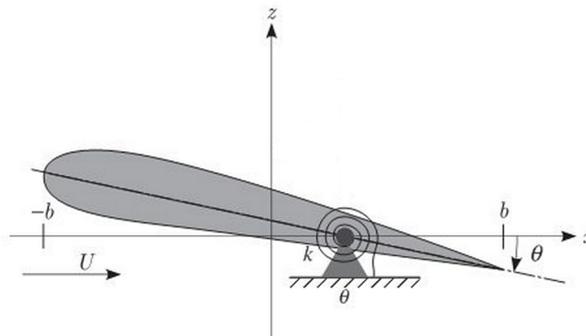


Figure 2.1: Single DOF flutter system. Taken from [1]

For the single degree-of-freedom (DOF) system of figure 2.1, representing a wing, the equation of motion reads:

$$I \ddot{\theta} + K \theta = M \quad (2.1)$$

where K is the stiffness of the equivalent torsional spring representing the wing's torsional rigidity, I is the polar moment of inertia of the wing, θ is the torsional displacement and M is the fluid torque. In classical flutter the motion is presumed harmonic:

$$\begin{aligned} \theta &= \hat{\theta} \sin(\Omega t) \\ M &= \hat{M} \sin(\Omega t) \end{aligned} \quad (2.2)$$

where the hat means amplitude of the corresponding variable. The distinctive characteristic of an aeroelastic instability is that the amplitude of the load depends on the amplitude of the displacement:

$$\hat{M} = L(U, \rho_{\infty}, b, K, \Omega, \dots) \hat{\theta} \quad (2.3)$$

where L is a model constant related to the aerodynamic load which is, in turn, a function of physical and geometrical parameters such as the undisturbed fluid density ρ_{∞} , the airfoil size b , the stiffness K , the frequency of oscillation Ω , among others. This helps to distinguish instabilities from forced oscillations. In the former, the applied force is coupled with the system (self-excited) and grows exponentially, while in the latter the forces are independent of the system, leading to a linear growth. More complex forms of classical flutter appear in real systems, for which more DOF are considered, for example for the flapping motion.

Other types of instabilities which might occur are:

Panel flutter This type of flutter takes place in thin elastic structural panels, for example in a plane's fuselage.

Divergence Divergence is a quasi-static condition where the blade or wing twists in response to increasing load in a direction that further increases the load. If this condition exists on a blade, an operating speed can be found at which the increase in loads caused by the deformation exceeds the ability of the blade to resist the load. This is called divergence [10].

Stall flutter Also called "stall-induced vibrations". For airplane wings, this is the non-linear extension of classical flutter, when the amplitude of variation in the angle of attack is so high that the stall region is reached. In the context of wind turbine blades, it is associated with a translatory movement of a blade section rather than a rotation, as in classical flutter. Furthermore, the flap and edgewise oscillations are usually analysed simultaneously [21]. For that reason, this kind receives also the name of "Flap-lead/lag flutter". For example, recent investigations [29] showed for a particular case, that the vibrations of the blades for the backward and forward edgewise whirling modes are not equally damped. The forward edgewise motion is coupled with a flapping motion of greater amplitude than the corresponding for the backward edgewise motion. Therefore, the damping of the coupled flapping motion is decisive for the apparent damping of the edgewise motion. Stall flutter is mostly found in stall-regulated rotors [30].

Unlike classical flutter, this phenomenon is dominated by **dynamic-stall**, a strongly viscous case [15]. Dynamic stall occurs on a blade section when the local angle of attack rapidly increases through the

static stall point [31]. The fast shift in the angle of attack is driven by the unsteady nature of wind in form of turbulence, shifts in wind direction or magnitude, wind shear, etc. In dynamic stall, extremely large suction peak values are found at or near the leading edge, leading to cyclic de- and reattachment, either on part or the whole wing surface. Dynamic stall exhibits hysteresis [32], which contributes to its non-linear character.

2.2.2 Forced oscillations

To study the resonance phenomenon, the case of forced oscillations with damping is explained. For a linear, single DOF system like the one in figure 2.2, the equation of motion reads:

$$m\ddot{x}(t) + c\dot{x}(t) + kx(t) = F(t) \quad (2.4)$$

The oscillatory motion results of the solution of equation 2.4 and consists of the superposition of the homogeneous and an inhomogeneous solution. The latter describes the transient response, while the former describes the steady-state oscillatory motion. In the case of a harmonic exciting force, the particular solution is also harmonic and of the same frequency as the force:

$$x(t) = \hat{x} \sin(\Omega t - \zeta) \quad \text{for} \quad F(t) = \hat{F} \sin(\Omega t) \quad (2.5)$$

where \hat{x} and \hat{F} are the amplitude of the displacement response and exciting force respectively, and ζ is the phase angle. The response function x can be further characterized with the help of the the normalized exciting frequency:

$$\eta = \frac{\Omega}{\omega_0} \quad (2.6)$$

the damping degree:

$$\vartheta = \frac{\delta}{\omega_0} \quad (2.7)$$

and the amplification factor:

$$V_1 = \frac{\hat{x}}{\hat{x}_{stat}} \quad (2.8)$$

where δ is the decay rate, ω_0 is the natural frequency of the system and \hat{x}_{stat} is the static deflection. Figure 2.3 shows the variation of the amplitude and phase with respect to the aforementioned parameters. A particularly

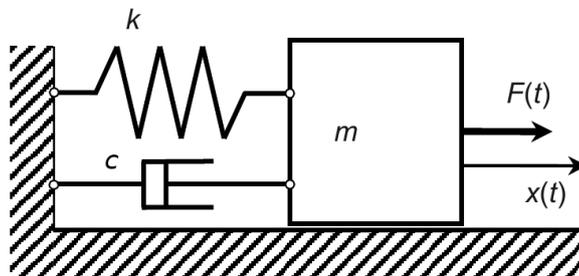


Figure 2.2: Single DOF oscillatory system. Taken from [2].

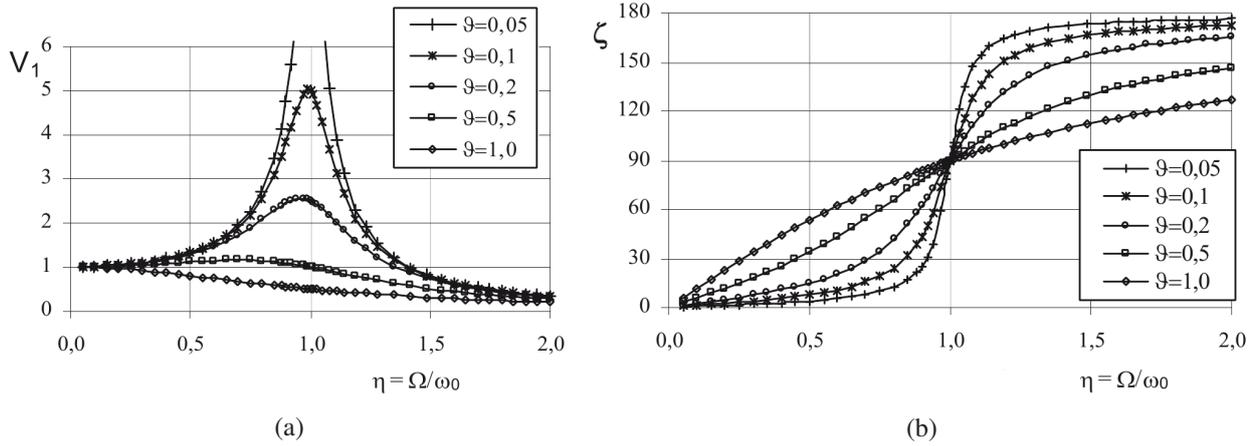


Figure 2.3: Response amplitude and phase of the single DOF system. Taken from [2].

dangerous case, called resonance, occurs for $\eta = 1$. Under these conditions, the amplitude grows to infinity if no damping is present.

In the case of more complex systems with many DOF, equation 2.4 can be analogously rewritten in matrix form. From the system matrix the eigenfrequencies and deformation eigenmodes can be obtained. If the equation of motion allows modal decoupling, which is the case for many common engineering applications, the eigensolutions provide the basis for the modal decomposition [33]. In that method, the eigenmodes are used as an algebraic basis to express the steady-state system deformation to an exciting harmonic load. Theoretically, there can be as many eigensolutions as original DOF, but for most applications only the lowest modes are relevant. In this way, a reduction of DOF is achieved. The solution of the new algebraic system consists in obtaining the contribution of each eigenmode to the deformation. A load with a spatial distribution that coincides with a particular eigenmode, will only trigger a deformation with that eigenmode. Loads with a general spatial distribution will trigger many eigenmodes. The amplitude and phase of each eigenmode behave analogously as for the 1-DOF system, figure 2.3. The response history results from the superposition of contributions of all triggered eigenmodes. Since this method computes the forced response, the oscillation frequency is determined by the harmonic load. Periodic but not harmonic loads can be additionally expressed as a Fourier series and, again for linear systems, the superposition method allows to obtain the total response by summing up the particular solutions of each term.

2.3 Flow dynamics

2.3.1 Physical model

Flow physics can be modeled with different approaches such as the Lattice Gas Method, the Panel and Vortex Panel Methods and the solution of the Navier-Stokes equations. The model choice depends on the required level of accuracy and detail, and the available computing resources. In this work, the flow dynamics are solved with the compressible Navier-Stokes equation since, despite their high computational cost, they are quite general and offer a high degree of detail. This is implemented in the in-house CFD solver *SPARC*.

Unlike the previous section, we deal now with models with distributed parameters, belonging to the field of continuum mechanics. This implies the formulation of partial differential equations (PDEs) which are to be solved in a volume. The accurate manipulation of these equations requires tensor analysis [34]. The general solution of PDEs is done through the application of specialized numerical methods and require the

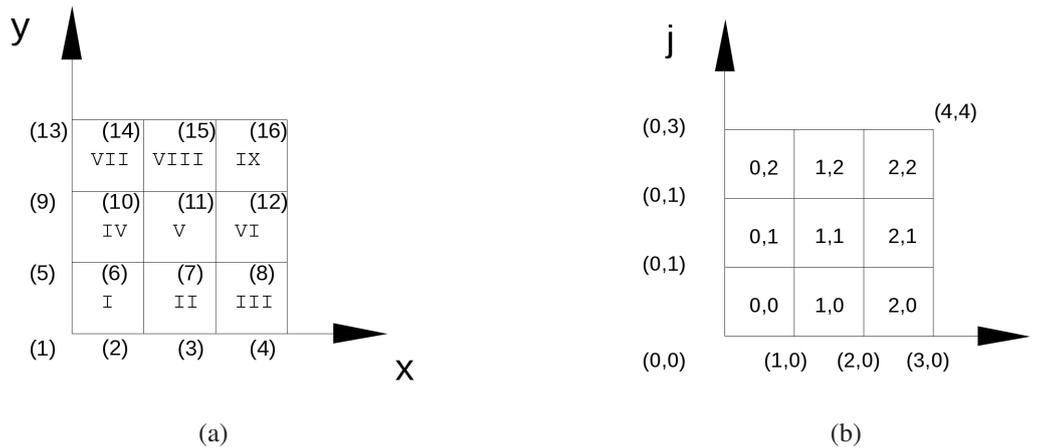


Figure 2.4: Numbering scheme of structured and unstructured meshes.

partition of the problem geometry, i.e. a “space discretization”. There are three major methods to deal with space discretization: the Finite Differences (FD), the Finite Volumes (FV) and the Finite Element (FE) method.

The partition of the simulation domain is known as mesh. According to Löhner et al. [35], there are two ways how a mesh resolves the geometry:

Body-Fitted approaches Here the boundaries are defined by adapting the mesh around them, so that no physical boundaries cross the cells. This is more accurate but demands some effort for designing the mesh. The term “Interface Fitting” is also used in this context.

Immersed Boundary Method (IBM) The computational domain is filled with a cartesian grid and the physical boundaries are “captured” within the mesh cells, requiring interpolation procedures [36]. These methods are alternatively known as “Embedded Approaches”.

As regards its topology, a mesh can be classified into:

Unstructured mesh Here, the cells and nodes are numbered without any prescribed pattern (figure 2.4a).

A connectivity matrix is needed to resolve the relationship between cells and nodes. A greater flexibility is achieved at the cost of less regular meshes. The application of multigrid strategies is less efficient in unstructured meshes since agglomeration algorithms are required to generate a coarser mesh. Regarding parallelization, although a very good load balance can be achieved, the domain partition is not straightforward and requires additional data structures and mapping algorithms.

Block-structured mesh Here, the cells and nodes are numbered with the help of indexes (figure 2.4b).

The cost of generation is high since the process of adapting the block-topology to the boundaries is difficult to automate, but more regular grids can be achieved. Regular grids are normally less distorted and better aligned with the dominating flow direction, which improves the conditioning of the coefficient matrix of the algebraic system of equations. Additionally, the index-based data organization results in a structured form of the coefficient matrix. This has an important positive impact on the computational performance:

- Particularly efficient methods can be used to solve the system of equations

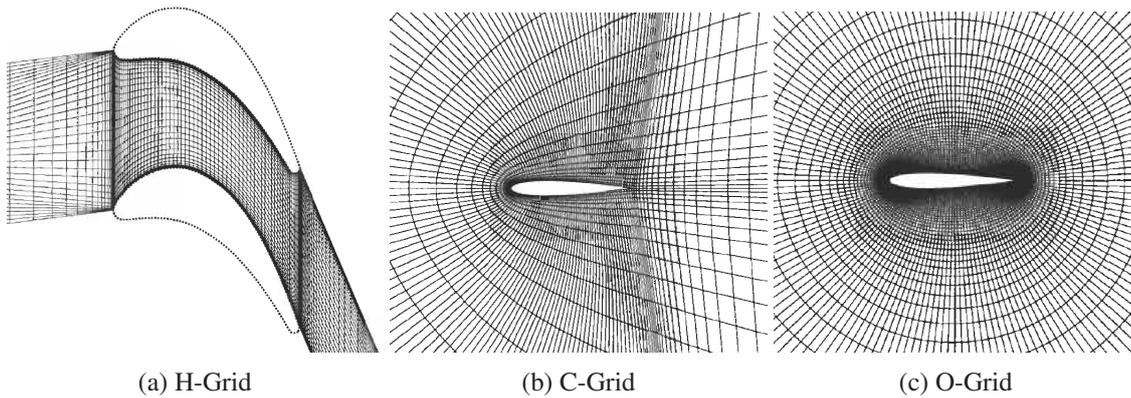


Figure 2.5: Different types of structured meshes. Taken from [3].

- The data storage requires less memory, since no connectivity matrix is needed, and is more contiguous, which leads to a faster memory access.

Block-structured meshes also simplify the task of dividing the domain for parallel solution. The block boundaries provide a natural partition of the simulation domain, and the structured data arrays allow for a direct exchange of dummy cells. Finally, geometric multigrid strategies are particularly efficient on block-structured meshes since the generation of coarser meshes is trivial. Three types of basic grid are typically found: C-, O- and H-Grids (figure 2.5).

In the context of Fluid-Structure Interaction, the fluid domain does not remain static but moves and deforms. For the IBM no additional treatment is needed in the equations. For Body-Fitted approaches, a series of methods has been developed to deal with moving meshes, among which the most notable are the Dynamic Meshes approach [37], the Co-rotational approach [38], [39], the Space-Time formulation [40] and the Arbitrary-Lagrangian-Eulerian (ALE) formulation [41]. The Space-Time formulation fulfills the GCL (see section 2.3.2) automatically [42] but has the disadvantage of a significantly increased size of the resulting equation systems [40]. In particular, when higher order accuracy in time is desired, sequential discretization is preferred [43].

Van Loon et al. [44] and Dam et al. [45] performed a comparison between Body-Fitted and Immersed Boundary approaches, confirming that Body-Fitted approaches are the best methods when too large deformations are not expected (as is the case for turbomachinery). When large deflections are present, however, techniques using Body-Fitted grids may have to deal with distorted meshes or even be hindered if the structural surfaces come in contact with each other [46]. The IBM, on the other hand, demand no computational effort for the determination of a new mesh, but calculations are needed for the interpolation procedures for each new geometry.

The ALE approach is by far the most common method used in Fluid-Structure Interaction and is also used in this work. This method (and all Body-Fitted approaches) demands *Mesh-update procedures*, which can be classified according to [47]:

Mesh regularization techniques The strategy of these procedures is to avoid changing the mesh dramatically, in particular, to maintain the number of cells. The existent mesh is “regularized”, that is, adapted to new boundaries, by means of smooth displacement of its nodes. The most representative methods are the Transfinite Mapping [48], the Delaunay graphs [7], Elastic Analogy methods [49] and

the Elliptic Mesh movement [50]. Glück [51] remarks that algebraic methods, like the Transfinite Mapping, have the advantage that they are robust and fast as compared with Elastic Analogy methods.

Mesh adaptation techniques These techniques rely on the partial or complete regeneration of the mesh, by which the number of cells can increase or decrease. These methods are almost only suitable for unstructured meshes. An interpolation of the solution to the new mesh is required.

In *SPARC* and along the present work, the Navier-Stokes equations are discretized in space using the FV method on block-structured, body-fitted grids. The ALE formulation is then applied to deal with the moving boundaries. For the derivation of the ALE equations, a spatial and a reference configuration are identified, after Donea et al. [47]. The reference configuration, indicated with the “0” superscript, is identical with the global coordinate system, and the spatial configuration is given by the mesh. The integral form is needed as a starting point for a FV discretization. The compressible Navier-Stokes equations in conservative variables can be then written as:

$$\begin{aligned}
\frac{\partial}{\partial t} \Big|_{x^0} \int_{V_t} \rho_f dV + \int_{S_t} \rho_f \mathbf{c} \cdot \mathbf{n} dS &= 0 \\
\frac{\partial}{\partial t} \Big|_{x^0} \int_{V_t} \rho_f \mathbf{u} dV + \int_{S_t} \rho_f \mathbf{u} \mathbf{c} \cdot \mathbf{n} dS &= - \int_{S_t} p \mathbf{n} dS + \int_{S_t} \boldsymbol{\tau} \cdot \mathbf{n} dS + \int_{V_t} \rho_f \mathbf{b}_f dV \\
\frac{\partial}{\partial t} \Big|_{x^0} \int_{V_t} \rho_f E dV + \int_{S_t} \rho_f E \mathbf{c} \cdot \mathbf{n} dS &= - \int_{S_t} p \mathbf{u} \cdot \mathbf{n} + \int_{S_t} \mathbf{u} \boldsymbol{\tau} \cdot \mathbf{n} dS - \int_{S_t} \mathbf{q} \cdot \mathbf{n} dS
\end{aligned} \tag{2.9}$$

$$\mathbf{c} = \mathbf{u} - \mathbf{u}_g$$

where \mathbf{u} is the fluid velocity, \mathbf{c} is the convective velocity, \mathbf{u}_g is the grid velocity, ρ_f is the fluid density, p is the static pressure, $\boldsymbol{\tau}$ is the deviatoric stress tensor, \mathbf{b}_f are the body forces, E is the total energy and \mathbf{q} is the heat flux, and V_t and S_t are the time-varying volume and surface of the control space. Equation 2.9 requires the following additional relationships:

$$\begin{aligned}
q_i &= k \frac{\partial T}{\partial x_i} & T &= \frac{p}{\rho_f R} \\
E &= \frac{\mathbf{u} \cdot \mathbf{u}}{2} + e & e &= \frac{p}{\rho_f (\gamma - 1)} \\
\boldsymbol{\tau} &= \mu \frac{\nabla \mathbf{u} + \nabla \mathbf{u}^T}{2} + \lambda (\nabla \cdot \mathbf{u}) \mathbf{I}
\end{aligned} \tag{2.10}$$

where q_i is the heat flux, k is the thermal conductivity, T is the thermodynamic temperature, R is the gas constant, e is the internal energy, γ is the isentropic exponent, μ is the laminar viscosity and λ is a coefficient that takes the value $-\frac{2}{3}$ for some common gases.

2.3.2 Discretization of the ALE equations

The discretization method applied to the ALE equations has an influence on the stability and accuracy of the numerical solution. Two issues have been investigated in the literature in order to assess the quality of discretization: the Geometric Conservation Law (GCL) and the Time-Averaging Method.

The terminology “Geometric Conservation Law” [52] is used to characterize geometrically conservative schemes, as algorithms that preserve the entire state of a uniform flow. The importance of the GCL was

adverted by Trullio and Trigger as early as 1961 [53]. Good reviews of this topic can be found, for example, in Förster [54] and Donea et al. [47].

The continuous GCL can be derived by taking the ALE equation of mass balance and assuming uniform fields of material velocity and density:

$$\frac{\partial}{\partial t} \bigg|_{x^0} \int_{V_t} dV = \int_{S_t} \mathbf{u}_g \cdot \mathbf{n} dS \quad (2.11)$$

The GCL influences the way the grid velocity \mathbf{u}_g is computed. The calculation of the grid velocity is not solely determined by the position of the grid points in the different time stages (determined in turn, by the mesh update algorithm), but also by the finite differences scheme used. The structural solver does only determine the position of the interface, but the discrete velocity used by the fluid and structural solver at the interface need not be the same, due to the different time integration schemes. Besides, the structural velocity is not transferred as variable to the fluid field, but remains as an internal information for the structural solver.

As noted by Lesoinne et al. [42], second order finite differences rules were initially used in an intuitive manner for the calculation of the grid velocity. For example, this Backward-Second-Order Finite Differences (BDF2) scheme:

$$\mathbf{u}_{wall}^{n+1} = \frac{3\mathbf{x}^{n+1} - 4\mathbf{x}^n + \mathbf{x}^{n-1}}{2\Delta t} \quad (2.12)$$

is used in *SPARC* to compute the velocity for the interaction walls. If the same scheme is applied to the grid velocities, fictive fluxes appear due to the violation of the GCL. To overcome this problem, Ferziger et al. [55] and Lai et al. [56] proposed methods where the volume swept by a finite-volume cell-face is introduced in the GCL to compute the grid velocity:

$$\mathbf{u}_{g,c}^{n+1} = \frac{3\delta V_c^{n+1} - \delta V_c^n}{2\Delta t A_c} \quad c = \text{faces of finite volume cell} \quad (2.13)$$

where δV_c is the volume swept by a cell face c and A_c is the area of face c . This scheme ensures the fulfillment of the Geometry Conservation Law and is the approach implemented in *SPARC*.

Despite the successful implementations, some controversy still exists regarding the necessity and sufficiency of the GCL to guarantee stability [43], [47], [57].

The second issue is about the Time-Averaging Procedure, as designated by Farhat et al. [58]. In the ALE equations 2.9, the first term requires the evaluation of a flux on a time dependent volume. The other terms, conversely, contain surface and volume integrals which do not change because they do not contain time derivatives, and are therefore determined at a certain time point, in which S and V can be looked upon as constant [51]. The evaluation of the first term depends on what time discretization scheme is used, i.e. on the Discrete Geometric Conservation Laws (DGCL).

To address this problem, a sequence of carefully chosen mesh configurations must be identified and evaluated, from where the name originates. The design of an ALE time-integrator that preserves the order of accuracy of its fixed-grid counterpart and simultaneously fulfills the DGCL is no trivial matter [57]. For incompressible flows, an alternative is proposed by Förster et al. [54], by applying the Continuous GCL before the discretization, thus avoiding the need of time-averaging procedures. In this way, equation 2.12 can be applied without violating the GCL and preserving the order of accuracy of the fixed-grid variant.

A last word must be said as regards the boundary conditions in the ALE formulation. For the calculation of prescribed velocities at the interaction surface, a scheme consistent with the overall time integration must be used, as given for *SPARC* by equation 2.12. This is independent of the abovementioned considerations for the GCL.

2.3.3 Numerical solution

In *SPARC*, a highly efficient solution of steady-state problems is achieved by the combination of the two schemes: a Runge-Kutta-based iterative solver, after Jameson et al. [59], and the multigrid strategy, particularly efficient on block-structured meshes. In *SPARC*, the “Full Multigrid” scheme has been implemented [60], [61]. The basic concept is to use coarser versions of the computational grid to make the high wave-length components of the error converge faster. For this purpose, a so-called “V-cycle” is performed, which consists of “restriction” and “prolongation” operations, as shown with the double lines in figure 2.6. n_2 , n_3 and n_4 indicate the number of V-cycles in each multigrid level. n_1 is not indicated since for initialization in the coarsest level no V-cycle is possible. The restriction transfers the solution and residuals of a fine grid to a coarser grid, where a computation is carried out. This last solution is used to correct the one on the finer grid in the prolongation step. This basic scheme can be combined in several different ways to achieve an acceleration factor between 2 and 10 in comparison with a single-grid computation. An additional advantage of the Full Multigrid scheme is that computations can be started on the coarsest grid, MGL=1 in our example. In that case, an interpolation is performed (dashed lines) from coarse to fine grid to provide an initial solution, since the V-cycle starts on a fine grid. Due to the fact that in a block-structured mesh every coarsening step produces an 8-times smaller grid, very fast computations can be performed in the beginning of a simulation, where a high resolution is still not necessary. The fast computations also allow to easily check the model assumptions in a short time.

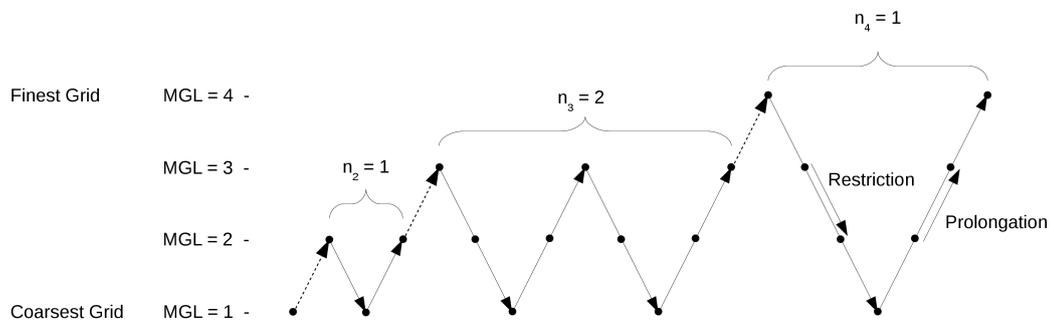


Figure 2.6: The Full Multigrid scheme.

These efficient methods can be extended to unsteady solutions by means of a dual-time stepping scheme [62] with second-order accuracy in time.

2.3.4 Rotating models

There are basically two ways to handle fluid rotating models:

Rotating mesh A transformation is applied to the coordinates of all mesh points to account for the rotation of the model. Rotating accelerations can be easily taken into account. All forces resulting from the rotation are naturally computed.

Rotating frame of reference The mesh coordinates remain constant but a rotating coordinate system is used. Due to this, the body forces due to the rotation must be added as extra source terms. This is usually applied to steady-state computations, for which constant rotating speeds are normally used.

When stationary and rotating domains are simulated together, a series of techniques are to be applied to connect the two domains. For example, in *SPARC* a sliding interface [63] can be applied to unsteady computations. Liu et al. [64] compared some common techniques used for industrial applications showing the different results they produce.

2.4 Structural dynamics

2.4.1 Nonlinear dynamic analysis

The formulations presented here correspond to standard FE methods implemented in the structural solvers used in this work, *ADINA* and *CalculiX*. In a fully-coupled Fluid-Structure-Interaction problem, it is in general considered that the change in geometry of the structure cannot be neglected. An appropriate formulation to account for large displacements is then required. The corresponding kinematics in Lagrangian formulation are derived with the help of two configurations, the reference configuration and the material configuration. The first refers to some known state of the structure (usually the initial) and the second to any other state. The position of every particle of the structure is described with the vectors \mathbf{x}^0 and \mathbf{x} , as seen in figure 2.7. The displacement vector is therefore defined as the difference between the position of a particle in the two configurations: $\mathbf{d} = \mathbf{x} - \mathbf{x}^0$. The deformation gradient tensor can then be readily defined as:

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{x}^0} \quad (2.14)$$

From the deformation gradient, the more convenient Green-Lagrange deformation tensor can be constructed:

$$\begin{aligned} \mathbf{E} &= \frac{1}{2}(\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) \\ \mathbf{E} &= \frac{1}{2}(\nabla \mathbf{d} + \nabla \mathbf{d}^T) + \frac{1}{2} \nabla \mathbf{d} \cdot \nabla \mathbf{d}^T \end{aligned} \quad (2.15)$$

The Green-Lagrange tensor is a nonlinear deformation measure which reduces to the well-known infinitesimal deformation measure when the displacement gradient tends to zero:

$$\lim_{\nabla \mathbf{d} \rightarrow 0} \mathbf{E} = \frac{1}{2}(\nabla \mathbf{d} + \nabla \mathbf{d}^T) =: \boldsymbol{\varepsilon} \quad (2.16)$$

A formulation that uses the Green-Lagrange strain measure is known as a “large displacements” formulation whereas one which uses the infinitesimal deformation measure is known as a “small displacements”

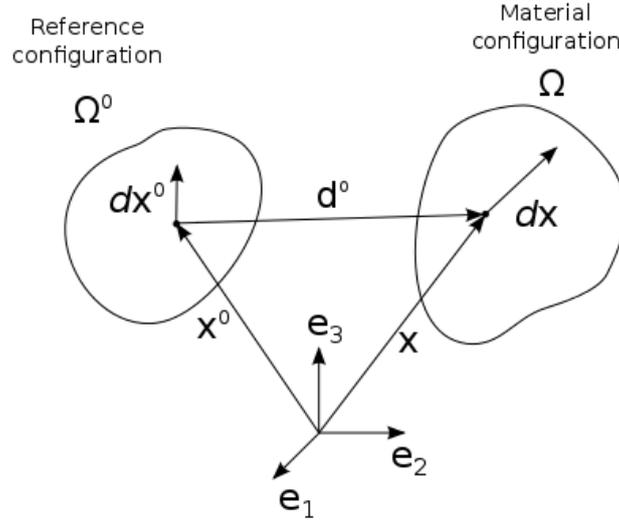


Figure 2.7: Reference and material configurations.

formulation. The necessity of a large displacements formulation depends on how fast the rigidity of the structure changes with the deformation. For beam type structures, for example, the increase of rigidity with deflection is slow. For a flat panel, conversely, a slight deflection produces an enormous increase in the rigidity. Therefore, a trade-off between computational cost and accuracy is to be made for each particular case.

Other types of nonlinearities include inelastic material behaviour due to high-strains, contact and fracture, which are seldom considered in a Fluid-Structure-Interaction problem.

For nonlinear dynamics, the equations ruling the structural deformation correspond to Newton's second law of motion, which are naturally posed on a Lagrangian frame:

$$\frac{D}{Dt} \left(\rho_s \frac{D \mathbf{d}^0}{Dt} \right) - \nabla \cdot \boldsymbol{\sigma}_s = \rho_s \mathbf{b}_s \quad \text{in } \Omega \quad (2.17)$$

where \mathbf{d}^0 are the displacements, ρ_s is the structural density, $\boldsymbol{\sigma}_s$ is the Cauchy stress tensor and \mathbf{b}_s is the body force.

This equation is posed in the material configuration and, therefore, contains physically meaningful, measurable quantities. For the numerical solution it is, however, more convenient to rewrite it in the reference configuration:

$$\rho_s^0 \ddot{\mathbf{d}}^0 - \nabla_0 \cdot (\mathbf{F} \cdot \mathbf{S}) = \rho_s^0 \mathbf{b}_s \quad \text{in } \Omega^0 \quad (2.18)$$

where the introduction of the second Piola-Kirchhoff stress tensor \mathbf{S} allows for the use of the St.Venant-Kirchhoff material model:

$$\mathbf{S} = \mathbf{C} : \mathbf{E} \quad (2.19)$$

where \mathbf{C} is the constitutive tensor which depends on the material.

Equation 2.18 is discretized in *CalculiX* according to the Finite Element Method on unstructured meshes and integrated in time with the alpha Hilber-Hughes-Taylor method. The solution of the system of equations is carried out by means of the sparse direct solver *SPOOLES* [65]. Further *CalculiX* implementation details can be read in the book by Dhont [66]. For a deeper discussion on the mathematical formulation the reader is referred to specialized texts in the Finite-Element Method and continuum mechanics such as Ibrahimbegovic [67] and Dvorkin [34].

2.4.2 Rotating models

Similarly as described for CFD, there are also two ways to handle rotating models and, in particular, the rotor of a turbomachine:

Rotating mesh This is only possible if large displacements are accounted for in the formulation. All forces resulting from the rotation are then naturally computed. This includes the Coriolis force which is usually neglected in the rotating frame of reference approach. There are two possibilities:

Constant rotation speed (prescribed rotation): the rotational DOFs of the nodes on the rotation axis are prescribed with a time-dependent function, normally a linear function with constant slope (constant velocity). The translatory DOFs, conversely, are fixed.

Free rotor: The translatory DOFs of the nodes on the rotation axis are fixed as in the previous case. The rotational DOFs, however, are left free. The additional resisting (generator) or driving (motor) torque from the shaft must be provided through external calculations. In this way the effect of the attached electrical machine and other mechanical components is modeled and the velocity of the rotor is determined by the balance between fluid, dynamic, and shaft forces.

Rotating Frame of Reference In this case, the shaft is modeled with zero-displacement boundary conditions on some chosen nodes that represent the shaft. The body forces due to a constant rotating velocity are included in the source term \mathbf{b}_s in equation 2.18. These forces can be of two types: the centrifugal force and the coriolis force. The first contributes with a stiffening effect and adds a considerable load on the rotor. The contribution of the coriolis forces is normally not taken into account since for rigid rotors the relative velocity is negligible. The term due to rotating accelerations i.e., due to the change of rotating velocity, is, in general, not included in the formulations for a rotating frame of reference.

2.5 Fluid-Structure Interaction

The term Fluid-Structure Interaction refers to a very broad spectrum of physical phenomena where an object is in contact with some sort of flow, and both the object and the flow are examined together. Besides turbomachinery, reliable FSI solvers are demanded in areas as diverse as civil engineering [51], fabric-like structures such as windsocks, parachutes and sails [68], hemodynamics [69] or noise control in flow ducts [70]. The mutual influence that takes place between the flow and the object might be of very different kind, thus defining different fields of research. The solution methods applied to each of them might therefore be also very different. So, for example, in a rigid water pump the vibrations might be solely determined by the pressure loads, the deformation of the vanes having no influence on the flow. In contrast, in medical applications a strong coupling exists between the vein tissues and the blood flow through it. For some

applications, zero- or one-dimensional models might be accurate enough, while for others expensive three-dimensional simulations are required. This section describes only methods that match the formulations for the solution of flow and structural dynamics presented previously. The focus is placed on the additional conditions that result from the interaction, and how the resulting coupled system of equations can be solved. To gain more insight, the section begins with a classification of the different approaches that can be found in the literature.

2.5.1 Classification of FSI approaches

The work of Felippa et al. [71] provides a good starting point to characterize the existing FSI approaches, with the identification of the four components that are found in any computational FSI approach:

1. Fluid solver
2. Structural solver
3. Time stepping approach
4. Interface treatment.

For the remaining discussion, the term *part-solver* is introduced to refer to either the fluid or structural simulation software packages. The first two components have already been presented in the previous sections. The other two can be used to classify most of the existing FSI approaches:

1. Time-stepping approach:

An important classification, introduced by Park et al. [72], is in terms of Monolithic or Partitioned approaches. This differentiation has been traditionally understood related to the way the algebraic system of equations is solved: as a whole, or fluid, structure and interface separately. More currently, Felippa et al. [71] presents it as a means of understanding how the time is advanced.

a) Monolithic approaches:

In these approaches, a single spatial and time discretization method is applied to both the fluid and structural fields, leading to a single coupled system of equations, which is also solved simultaneously. Since a single time-integration scheme is applied to the whole system of equations, a certain degree of inefficiency is implicit. Monolithic approaches have a number of drawbacks like loss of software modularity, limitations with respect to the application of different sophisticated solvers in the different fields, and challenges with respect to the problem size and conditioning of the overall system matrix [73]. Some implementation examples of the Monolithic approach can be found in the works of Hron et al. [74], Dunne [75], the interesting platform by Heil [76] and the commercial software package *ADINA* [77].

b) Partitioned approaches:

Here, specific discretization methods are applied separately to the fluid and structural fields, leading to independent systems of equations that are also solved separately. This can in practice be achieved by completely independent software packages. The two solutions are to be coupled through appropriate interaction boundary conditions and synchronized by appropriate coupling schemes. Partitioned approaches exhibit, according to Farhat et al. [78], the following advantages:

- Reduce the computational complexity per time step
- Simplify explicit/implicit treatment
- Different time-integration schemes and time steps (sub-cycling) can be used for each of the fields, leading to a possibly more optimal solution
- Ease computational load balancing (better work load distribution among CPUs)
- Achieve software modularity, making replacement relatively simple
- Enable the use of specialized and well tested pre-existent software packages for the individual fields

Some disadvantages can also be found in the literature, such as a more difficult treatment of the interface [74] and convergence problems for strongly-coupled problems.

For a comparison between Monolithic and Partitioned approaches in terms of computational cost the reader is referred to the work of Heil et al. [79].

2. Interface treatment:

If the meshes at both sides of the interface match, then the transfer of variables from one field to the other is a relatively trivial task. In most realistic applications, however, the fluid and structure meshes are incompatible along the fluid/structure interface, either because they have been designed by different analysts or because the fluid and structure problems have different resolution requirements [80]. For example, in aeroelastic computations, the fluid grid is typically finer than the structure mesh [81]. Furthermore, matching meshes might be impossible for some combination of discretization methods (in the partitioned approach) or mesh types. For example, if hexahedral finite-volume cells are used for the fluid mesh and parabolic hexahedral elements are used for the structural mesh, an interface treatment is needed even if both meshes might look identical. Actually, parabolic elements have midside-nodes which are not automatically matched with the finite-volume mesh nodes. Additionally, the finite-volume cells deform in a bilinear manner, while the finite-element hexahedra can assume a curvature.

The Monolithic approach does not automatically exclude the necessity of interface treatment either, since fluid and structural meshes need not be matching.

Other classification criteria

Another classification can be done in terms of two- or three-field formulation. FSI usually comprises the solution of two fields of mechanics, namely, the fluid and the structure. Lesoinne and Farhat [82] solved the FSI problem on unstructured meshes and used an elastic analogy method to deal with the fluid grid adaption. Therefore, they considered the addition of this third set of partial differential equations to lead to a three-field formulation.

FSI methods can be further classified according to the direction of coupling. The most simple way to couple the fluid and structural mechanics fields is to transfer only the loads calculated with a CFD solver into a CSD solver. In this case, the fluid problem receives no influence of the structural deformation, saving the interpolation of displacements and the adaption of the fluid mesh. This is known as *one-way coupling* and is widely used for turbomachinery simulations, for example, in the work of Filsinger et al. [83].

This is accurate enough as long as the structure is stiff, but rigorously, this does not represent a coupling in the sense of the general definition coined by Zienkiewicz et al. [84], according to which, “no field can be solved independent of the other”. This last definition is, therefore, often referred to as *two-way coupling*.

2.5.2 Partitioned approaches

Due to the already exposed advantages, this work follows the partitioned approach. In order to solve the Fluid-Structure-Interaction problem accordingly, the non-linear systems of algebraic equations presented in the previous sections are to be coupled so as to fulfill the interaction conditions. The different alternatives are referred to as *partitioning types*. Once the interaction conditions have been applied, the second step is the solution of the resulting coupled nonlinear system of equations. These two steps are detailed along the section.

Partitioning type

The predominant partitioning type in FSI solvers and also used in the present work is the Dirichlet-Neumann partitioning [85]. Other possibilities are described in the works of Von Scheven [49] and Mok [86].

In a Dirichlet-Neumann partitioning, a Dirichlet boundary condition is imposed to the fluid field at the interaction boundary by means of prescribed displacement and velocity, and a Neumann boundary condition is applied to the structural field by means of interaction forces [49]. Consequently, the interaction condition is the continuity of displacements and surface tractions [74], whereby the slip or no-slip condition might apply. For viscous flows (no-slip condition) [80] the continuity of surface tractions across the fluid-structure interface is expressed as:

$$\boldsymbol{\sigma}_s \cdot \mathbf{n} = \boldsymbol{\tau} \cdot \mathbf{n} - p \mathbf{n} \quad (2.20)$$

and the continuity of displacements:

$$\mathbf{d}^s|_{\Gamma} = \mathbf{d}^f|_{\Gamma} \quad (2.21)$$

where $\mathbf{d}^s|_{\Gamma}$ and $\mathbf{d}^f|_{\Gamma}$ are the displacements of the fluid and structural meshes at the interface respectively. These kinematic conditions also imply the respective time derivatives.

The accuracy in the fulfillment of the above equations depends not only on the interface treatment, (see section 2.5.5), but also on the time integration schemes described in the next two sections. An accurate solution near the interface is crucial for the overall exactness and performance of the coupling. Indeed, the order of accuracy of the whole partitioned method is determined by the order of the method with lower accuracy [43]. Consequently, a poor handling of the interaction conditions might degrade the spatial and time order of accuracy of the part-solvers.

Solution of the nonlinear system

Despite being solved separately, it is convenient to think of the whole coupled system as a system of non-linear algebraic equations, where not only the part-solvers are included but also the algorithms for interface treatment. The approaches for the solution of this nonlinear system are usually divided into loosely coupled or strongly coupled. In both cases, an implicit or explicit time integration scheme can be used, either in

the fluid or structural solver. Whether the overall coupled behaviour is described as explicit or implicit, however, also depends on the type of coupling:

- **Loosely coupled**

In loosely coupled approaches, also called “simple staggered”, “weak coupled” or “explicit” [81], only one data exchange takes place per time step [87], making them particularly efficient [49].

- **Strongly coupled schemes**

These are also called “iterative staggered” or “implicit”. In these methods, the data is exchanged between solvers iteratively until the coupling conditions are fulfilled.

The existing approaches are presented in the next two sections.

2.5.3 Loosely coupled schemes

Despite their efficiency, some authors [86] argue that loosely coupled schemes exhibit an inherent instability, which cannot be avoided by small time steps (actually increases). Other authors [78], on the contrary, sustain that the instability is not inherent to the loosely coupled approach but rather results due to inappropriate time integrators.

The reported instability appears as an error in the transferred fluid forces which act as an extra mass on the structural interface degrees-of-freedom. Since the instability highly depends on the ratio between fluid and structural density ρ_f/ρ_s it has been called “artificial added mass effect”. As explained by Idelson [88], in fluid mechanics, added mass or virtual mass is the inertia added to a system because an accelerating or decelerating body must move some volume of surrounding fluid, as it moves through it, since the object and fluid cannot occupy the same physical space simultaneously. The instability typically appears in incompressible fluids whose density is close to the structure density, such as in biomechanics applications. Some analyses show that the instability is caused by too large eigenvalues of the amplification operator of the explicit time step [89], [90]. Therefore, for some authors, the instability effectively prevent stable computations by means of staggered algorithms, and an iterative coupling is necessary [73]. Other authors [90] point out that when the added-mass effect is important, even the iterative process in the strongly coupled scheme may be non-convergent, and a monolithic scheme might be necessary to avoid numerical instabilities. For typical turbomachinery applications such as aeroelasticity, the added-mass effect is not a problem, since the density ratio is low enough [88].

Depending on whether the part-solvers are executed sequentially or simultaneously, loosely coupled approaches can be divided into serial or parallel:

Serial

A number of algorithms have been proposed:

Conventional Sequential Staggered (CSS) A classical algorithm is the Conventional Sequential Staggered, proposed by Farhat et al. [91], which has merely first order accuracy in time even if a second order accuracy scheme is used for the fluid and structural solvers [92]. The original algorithm, also implemented in the present development, is depicted in figure 2.8, where \mathbf{X} are the coordinates of the fluid mesh, \mathbf{W} is the fluid solution vector (a combination of pressure, velocity, density and energy,

that depends on the particular form used of the Navier-Stokes equations), \mathbf{F} are the forces determined from the fluid solution, \mathbf{u} is the structural displacement vector, and the subscripts indicate the time step. An alternative of the same algorithm is used by Massjung et al. [93], where the structure is solved first. The CSS introduces a one time-step lag between the fluid and structure solution and therefore, the displacement continuity is not fulfilled. This can lead, for example, to a misprediction of panel flutter period as high as 24% [93] if the time-step size is too large. In this algorithm, the time-step size is the same for the fluid and structural solver. The determination of the appropriate time-step size depends on the numerical properties of each solver and physics of the particular problem. Depending on the case, either the fluid or structural problem might dominate the time-integration resolution.

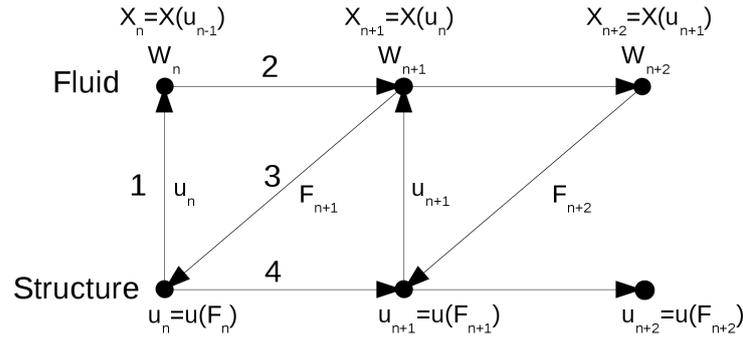


Figure 2.8: The Conventional Sequential Staggered (CSS) procedure.

Improved Serial Staggered (ISS) The previous method is synchronous, since both the fluid and the structural solver begin and end at the same time point. Farhat et al. [94] proposed an asynchronous method, with which continuity in displacement and velocity can be achieved. According to these authors, the allowed time step for this algorithm is five times larger than for the CSS. In connection with incompressible fluids it shows, however, poor stability behaviour [86].

Generalized Sequential Staggered (GSS) The most recent advance in loosely coupled algorithms is represented by the Generalized Sequential Staggered (GSS) algorithm [78]. This advanced method is second-order in time and must be implemented together with advanced ALE time integration schemes.

Parallel

Conventional Parallel Staggered (CPS) The basic scheme of this algorithm is shown in figure 2.9.

Farhat et al. [94] report that the CPS requires relatively small time-steps in order to be numerically stable and sufficiently accurate. A variant of CPS is also reported by Von Scheven [49].

Improved Parallel Staggered (IPS) Steindorf [81] and Farhat et al. [94] present similar improved versions of the parallel staggered algorithm. Despite performing better than the CPS, it is reported to be 1.6 times slower than the ISS [94].

Further improvements Further improvements can be achieved with the use of a prediction step, as proposed for example by Massjung et al. [93], Farhat et al. [94] and Piperno et al. [95]. The prediction is usually and most effectively done on the loads.

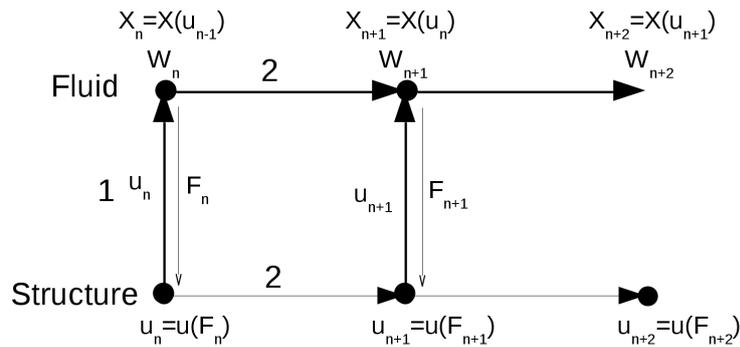


Figure 2.9: The Conventional Parallel Staggered (CPS) procedure.

Another possibility is the so called “subcycling”. Different time characteristics often exist in the fluid and structural fields. This leads to the need of different time-step sizes for each time-integration scheme. The fluid field needs usually smaller time steps [96]. Despite the advantage of allowing larger coupling time-steps, some measures are required to avoid instabilities [92], [96]. Subcycling is usually applied when explicit time integration is used for the fluid solver, due to the small time steps needed.

2.5.4 Strongly coupled schemes

These methods are based on the iterative data exchange between solvers until the coupling conditions are fulfilled, which improves the stability and energy conservation properties of these algorithms [97]. There exists a great variety of methods and no generally accepted classification can be found in the literature. A thorough work was done by Zhang and Hisada [98], who derived several algorithms using the monolithic approach as a starting point. The following is a short review of the most important alternatives. Since each author uses a different platform, a direct performance comparison is difficult.

Semi-implicit The use of this method is reported in the works of Grandmont [99], Badia and Codina [100] and Fernández [101]. Grandmont [99] shows that, for a simplified, one-dimensional case, this method is consistent and stable and consequently convergent.

Full-implicit Le Tallec and Mani [102, 97] present a full-implicit iterative staggered algorithm and perform an analysis on the energy conservation and convergence properties.

Predictor-corrector Examples of this kind are the works of Mouro [103], Schäfer et al. [104] and Lippold [7]. Schäfer et al. [104] show that, for a given case, an acceleration factor of more than 5 can be achieved.

Block iterative methods Von Scheven [49] classified several methods into Block-iterative or Newton-Krylow. The simplest block-iterative method is a block-Gauss-Seidel iteration scheme [105] which is easy to implement since it can be obtained by simply adding outer iterations to the CSS. A nonlinear improvement (NLBGS) has been proposed by Barcelos and Maute [106].

Newton-Krylow methods Although simple, coupling strategies based on fixed-point or Gauss-Seidel iterations are known to converge poorly or fail to converge for some specific configurations. Newton-Krylow, also called “block-newton methods” [107] provide a radical improvement. An “exact” Newton strategy requires the calculation of all coupling terms (cross-derivatives) between the different

fields. The corresponding expressions are often difficult to derive, difficult to implement and costly to evaluate. For this reason, “inexact block-Newton methods” have been developed, which are based on the approximation or neglect of selected coupling terms [108]. Von Scheven [49] provides a thorough description of the different alternatives. Examples of applications can also be found in the works of Tezduyar et al. [109], [110], Steindorf [81] and Vierendeels [111]

An additional important component of strongly coupled schemes are the “underrelaxation methods”. This is a common technique applied to iterative schemes in order to improve stability, in contrast to “overrelaxation methods” whose objective is to improve convergence rate. In a Dirichlet-Neumann partitioning, the underrelaxation can be applied either to the displacements, or to the loads (or both), or to the change of these variables [7]. A widely extended method is the “Aitken iterations”, successfully applied to FSI by Mok [86]. Another important aspect is an appropriate “convergence criterion”, which is needed whenever an iterative scheme is used. Some alternatives can be found in the works of Lippold [7] and Von Scheven [49].

2.5.5 Interface treatment

When non-matching grids are used, some sort of mechanism for the exchange of data between meshes is to be devised. This takes place in two directions:

- From fluid mesh to structure mesh for the load transfer
- From structure mesh to fluid mesh for the displacement transfer

For more accuracy, the condition that the virtual works are to be conserved through the interface can be applied [71], [112]. This leads to a direct dependency of the two data exchange procedures. The lack of conservation properties does not, however, automatically lead to a worse performance [80], but is rather only necessary when strong discontinuities are present in the solution, such as pressure shocks [113].

The classification proposed by Felippa [71], based on the presence or absence of additional interface variables as well as their type, is adopted here to introduce some important existing interface treatment methods:

Primal

There are no additional interface variables. The prototype for this group is the Direct Force-Motion Transfer (DFMT), which corresponds to the most intuitive approach. These methods can be additionally classified as local or global, depending on whether only adjacent or all degrees-of-freedom take part in the interaction. Two stages are usually needed, namely, a neighbour-search stage and an interpolation stage. In the former, groups of cells or nodes at each side of the interface are formed with a certain criterion. Some alternatives have been proposed by Löhner [114], Maman and Farhat [115] and Ahrem et al. [112]. It is worth noting that different neighbour-search procedures are usually needed for the load and displacement transfer.

The second stage makes use of the previously formed groups to perform the interpolation itself. The interpolation stage also comprises two different procedures for the load and displacement transfer. Some authors [51], [116] divide the interpolation schemes depending on whether the interpolated quantities are

conservative and non-conservative. Conservative quantities are, for example, the heat flux through an element area and the force exerted on an element area. The integral of such quantities must be preserved during interpolation. Non-conservative interpolated quantities are functions of the spatial coordinates and time, such as temperature, pressure and displacement.

There are two classical primal interpolation schemes:

Consistent interpolation This procedure can be better understood if a source and a target domain are identified first. For example, if pressure is being transferred, the target can be a Gauss integration point on a structural element face, and the source would be the projection of that point on the fluid cell face. If the “natural” interpolation functions for the pressure in the fluid mesh are used to get the value of pressure in the source point, this is said to be “consistent” [80], [113]. This is the approach followed in the present work for the transfer of pressure loads. Conversely, to interpolate the displacements, the consistent choice would be the structural interpolation functions (FEM form functions). The bilinear interpolation described by Glück [51] can be understood as a particular case of the consistent interpolation. This method has been adopted in the present work for the interpolation of displacements.

Conservative interpolation This method was developed by Farhat et al. [80] in order to ensure the conservation of forces and virtual works through the interface. Despite its conservation properties, Bardossy [117] points out that for certain mesh combinations this method might lead to an unphysical distribution of forces, since a force is not assigned to every structural node.

Dual

In the second group, we find as prototype the Mortar methods [118], which resort to Lagrange multipliers. The Mortar method has the advantage, against the consistent interpolation scheme, that the global error of discretization of the coupled problem does not become worse than the local discretization error in the part problems [119], [80]. A disadvantage of the Mortar method is that at each time step an expensive linear system of equations is to be solved [81].

2.5.6 Other considerations

Several authors [49],[120],[121] have devoted efforts to extend the advantages of the multigrid method (usually applied only to the fluid solver) to the whole coupled solution. The matter is non-trivial since the multigrid strategies in both solvers might be very different. Additionally, the multigrid method is not very widespread among structural solvers.

The conservation of energy plays an important role in the development of efficient partitioned methods, given its close relationship to the stability of numerical methods [96], [97]. Different authors [78], [122] have investigated the energy properties of coupling algorithms. For example, Farhat et al. [78] showed that monolithic and strongly coupled schemes conserve energy transfer at the fluid-structure interface boundary, whereas loosely-coupled algorithms introduce a certain amount of artificial energy.

Lastly, appropriate modelling assumptions are required for coupled problems, which might differ from those applied to the single field case [73]. A typical example is an appropriate initialization of the fluid field before the interaction starts, to avoid undesirable vibrations in the structure.

3 Development of the coupling tool

3.1 Introduction

In the previous chapter, the corresponding formulations of the underlying physics for a Fluid-Structure-Interaction problem have been presented. This chapter is devoted to exposing the approach followed for the implementation of the computational tool for the analysis of Fluid-Structure-Interaction problems. As already stated, a partitioned approach is followed to take advantage of well-proven part-solvers. For such an approach to work, an efficient software-wise solution must be found. This solution provides the framework for the development of the numerical methods that are needed at the different stages of an FSI computation.

3.1.1 Alternatives for the coupling of existing codes

In this section, the focus is placed on the issue of linking two simulation software packages in an efficient and flexible way. This problem goes beyond the particular application of FSI. It is indeed, an issue for many research groups that seek to extend the functionality of their codes. Different strategies have been considered:

- Use a scripting language such as *python* or *bash* to call the executable files making all communication through text files. Here, it is assumed that the solvers are compiled independently. This approach has been used at FSM for an optimization program, since it is fast to implement, but has the disadvantage of a sequential workflow and a slow communication.
- Embed the simplest code into the more complex one. This strategy had been followed at FSM for previous developments in FSI [4], [123] and for spray modeling. This approach has the advantage of an easy debugging, since a single executable is compiled, but has a series of drawbacks: works only for strictly sequential workflow, makes the handling of dissimilar data types cumbersome, complicates the compilation and the task of updating the third party source code after each new release, and does not facilitate a modular handling. The code structure asymmetry between the CSD and CFD solvers hinders flexible management of loops, needed for implementation of advanced features.
- Develop an interface using the idea of an “Algorithmic Skeleton”, as proposed by Jürgens [124]. This alternative requires much more insight in the code structure and appears impractical for existing solvers.
- Use a coupling manager based upon the Message-Passing Interface [125], [126] to account for the communication between programs. This alternative provides a great degree of modularity, a fast communication and a relatively straightforward implementation, since little intervention in the codes is required. The only disadvantage is that it results in a more complex software solution, for example, for debugging. Some research groups [7], [51] have resorted to commercial alternatives, such as the de-facto standard software package *MpCCI* [116], developed by the Fraunhofer Institute SCAI.

This however, prevents the developer from accessing and modifying important parts, such as the interpolation between meshes. To avoid that, some research groups have developed their own tools, such as *CoMA* [27], *preCICE* [28] and *MATCHER* [115].

Although the last approach requires some additional software provisions, it was judged to be very advantageous, leading to the creation of *FSiM*, the *Fluid-Structure-Interaction Simulation Manager*. This framework provides great advantages for the long-run code development at FSM, since other kind of couplings can be easily accommodated, such as fluid-structure heat transfer.

3.1.2 Structural solver

The choice of an appropriate structural solver is a delicate issue, since a wrong choice might lead to cumbersome data process or insufficient capabilities of the resulting coupled tool. For this reason, a variety of codes have been examined in order to assess how adequate they are for the proposed objective. The software packages *Calculix*, *OOFEM*, *FEAP* and *Code Aster* have been assessed, considering the following aspects:

Modeling features Features required for FSI simulations in turbomachinery include large-displacements formulation, explicit/implicit nonlinear dynamics, anisotropic and composite material modeling, centrifugal forces for the rotating reference frame approach, and structural (Rayleigh) damping.

Parallelization Parallel-solution support is desired, mainly for large models.

Pre- and post-processing The straightforward use of professional programs for model definition and solution evaluation is an important complement of any simulation tool.

Element library Different element types are necessary to fit particular models. A large element library provides greater flexibility.

Analysis types Apart from the main time-dependent simulation, a static analysis is useful for a fast determination of the stationary deflection and a modal analysis is desired to determine the vibration frequencies and modes.

Programming language This is important because MPI only supports Fortran, C and C++.

Place of development For possible cooperations, an accessible development group is preferred.

Cost and availability An open-source and cost-free program matches our development strategy better.

References A solver that has already been used in previous publications provides better chances of a successful implementation.

After a careful assessment, the open-source software package *CalculiX* was found to be the best candidate for the coupling with *SPARC* and *FSiM*. One of the greatest advantages *CalculiX* provides is an input format compatible with the commercial program *ABAQUS*[®] [127], ensuring a user-friendly model generation. Additionally, *CalculiX* has been used successfully in a similar development [7]. The lack of the structural damping feature was deemed uncritical, since that is a rather advanced modeling feature that can be neglected in many practical cases, for example in hydraulic turbomachinery [7]. The source code and documentation can be found in the official website [128].

3.2 General approach in FSiM

Once the members of the coupled solution are defined, a general approach to handle the different data sets is to be devised. Indeed, each part-solver makes use of very different data structures which must be made compatible in some way. To get an overview of how the data is organized, table 3.1 summarizes the terminology used for both types of meshes.

Fluid	Structure
Finite Volume Method	Finite Element Method
Block-structured mesh	Unstructured mesh
Block	-
Block-side	-
Cell	Element
Cell-face	Element-face
Cell-type: Hexahedron only	Element-type: Hexa and Tetrahedron
Number of nodes: 8	Number of nodes: 8, 4

Table 3.1: Nomenclature for fluid and structural meshes.

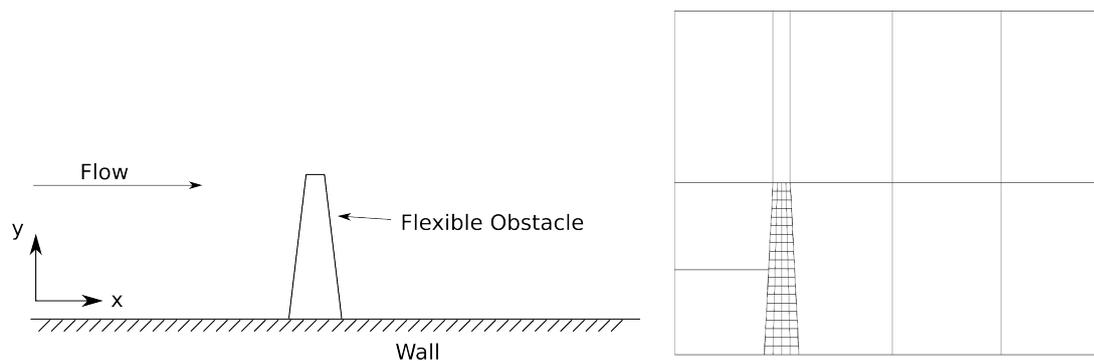


Figure 3.1: Example case. Left: Geometry. Right: block topology and structural mesh.

An example is introduced now to help visualize the procedures explained in this chapter. Figure 3.1 shows a flexible obstacle fixed on the ground and subjected to a flow parallel to it. The right figure shows a reduced computational domain to analyse this case, where the fluid block topology and the structural mesh are displayed. It must be taken into account, that *CalculiX* only allows for three-dimensional models, and that the two-dimensional *SPARC* models retain a three-dimensional data structure. For this reason, even when two-dimensional test cases are simulated, an internal three-dimensional logic exists.

One possibility to match the data sets at both sides of the fluid-structure interface is to generate a pair of unstructured, finite-element-ordered interface meshes, as done in *MpCCI* and *CoMA*. In this case, data is handled within the coupling interface exclusively in an unstructured manner. While this is a more general approach, it has the disadvantage that an extra mapping is needed between the interface and the original meshes. Since this mapping is to be repeated in two directions (from and to *FSiM*) for every coupling iteration, its efficiency is relevant. An alternative, more straightforward, approach is to duplicate the data structures of *SPARC* and *CalculiX* within *FSiM*. In this case, a hybrid unstructured/block-structured data structure is created within *FSiM*. Care must be taken not to duplicate unnecessary information. For that reason, the CFD mesh is not entirely transferred to *FSiM*, but rather only the block-sides that participate

in the interaction. The mapping from volume to surface mesh and back is particularly efficient for block-structured meshes, since the surface mesh has an analogous structure, as can be inferred from figure 3.2. For the fluid blocks data, convenient structures ensure contiguous data storage for cache optimization. The structural mesh is, conversely, entirely transferred to *FSiM*. This is not inefficient for the CSD meshes, because they are much smaller, and it has the advantage that *FSiM* can manipulate the whole structural model for scaling, rotating or to produce output. The *CalculiX* nomenclature was adopted in the *FSiM* data structures to account for the different kinds of elements.

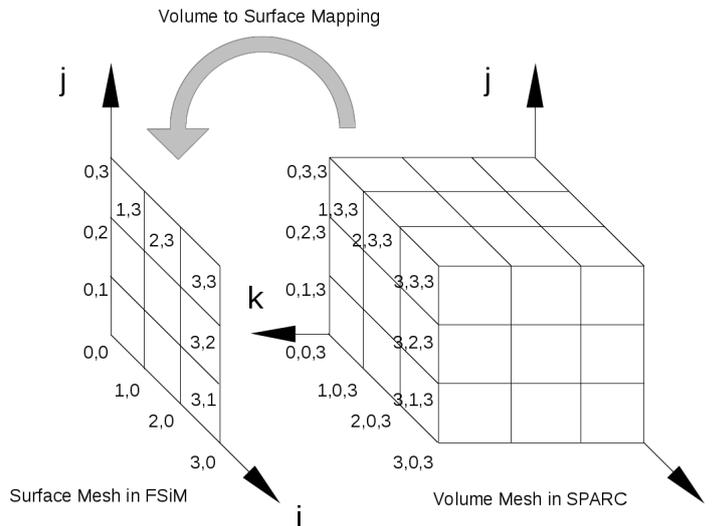


Figure 3.2: Mapping from volume to surface structured mesh.

Once the meshes are appropriately allocated in the coupling manager, provisions must be made to indicate to both solvers which part of the mesh bounds with the interaction surface. It is in general convenient to divide the interaction surfaces into sections, where a pair of interaction surfaces is formed from the fluid and structural side. These are called *Interaction Surface Pairs*. This has the advantage that the neighbour-search algorithm does not have to do an all-to-all search, and discontinuities in the geometry can be handled better.

The fluid input files are generated using *ANSYS*® *ICEM-CFD*™ Release 14.0 [129] where a special routine is embedded to generate the *SPARC* input files. For FSI, a wall-velocity boundary condition must be applied to the block-sides on the interaction surface. Apart from the type of boundary condition, a family number can be defined in such a way that several block-sides can be grouped into a single interaction-boundary-condition family. Each family defines the fluid member of an *Interaction Surface Pair*, as depicted in figure 3.3. This must be matched from the structural side with a group of element-faces, in such a way that the edges of those elements enclose a similar region, i.e. a structural element-face must belong to one and only one *Interaction Surface Pair*. In the example, the *Interaction Surface Pair* 1 consists of two block-sides and the element-faces depicted on the area 1 in the figure. For the generation of the structural mesh, the use of *ABAQUS* is preferred. The most general way of defining an interaction surface in the *ABAQUS* format is to create the so-called *Node Sets*, with all the nodes that lie on an *Interaction Surface Pair*. This information is later used to apply a boundary condition for the structural solver. This can be of two types: a pressure distributed load applied to the element-faces, or a concentrated force load on the nodes. This is elaborated upon in section 3.5.

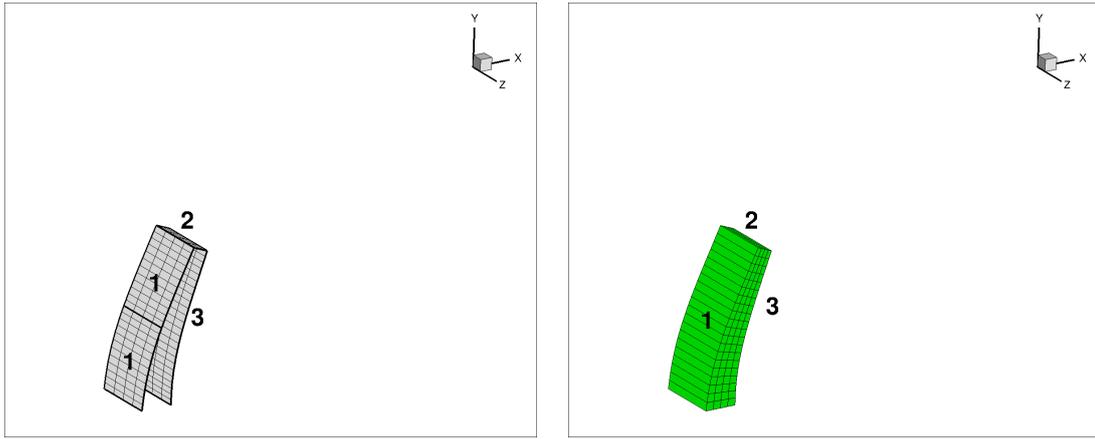


Figure 3.3: Surface interaction pairs. Left: In fluid. Right: In structure

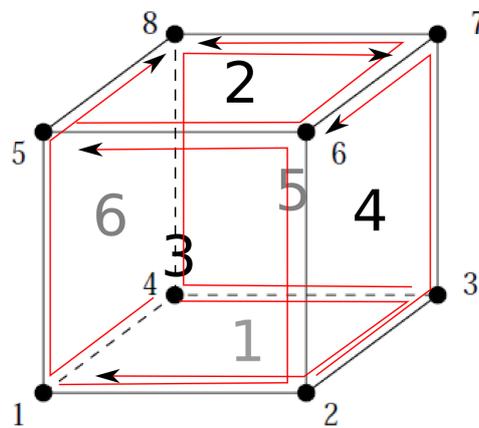


Figure 3.4: Face-numbering convention for hexahedron elements.

The output of the preprocessors *ICEM-CFD* and *ABAQUS*, provides only basic information to identify the Fluid-Structure-Interaction boundary condition. Therefore, before the neighbour search is done at the beginning of the computation, FSiM performs a series of sorting routines.

First of all, the *Node Sets* are converted into a list of elements and face numbers. The face numbering follows a convention that defines a precise sequence of element nodes, as shown exemplarily for hexahedron elements in figure 3.4. When the nodes that define a face are later needed, the information is obtained through an array with the face numbering convention and the connectivity matrix.

Secondly, to optimize the neighbour search, an array is created where, for each structural node in each *Interaction Surface Pair*, the adjacent elements are stored. This initial sorting reduces the computational effort compared to an all-to-all search.

3.2.1 Multigrid

In the *SPARC* fluid solver the use of multigrid¹ allows the user to initialize the computation with a coarse mesh and gradually switch to finer meshes. In this way, long and uninteresting transients can be quickly

¹The discussed feature does not mean the use of restriction and prolongation operation for the coupling computations, as described in section 2.3.3.

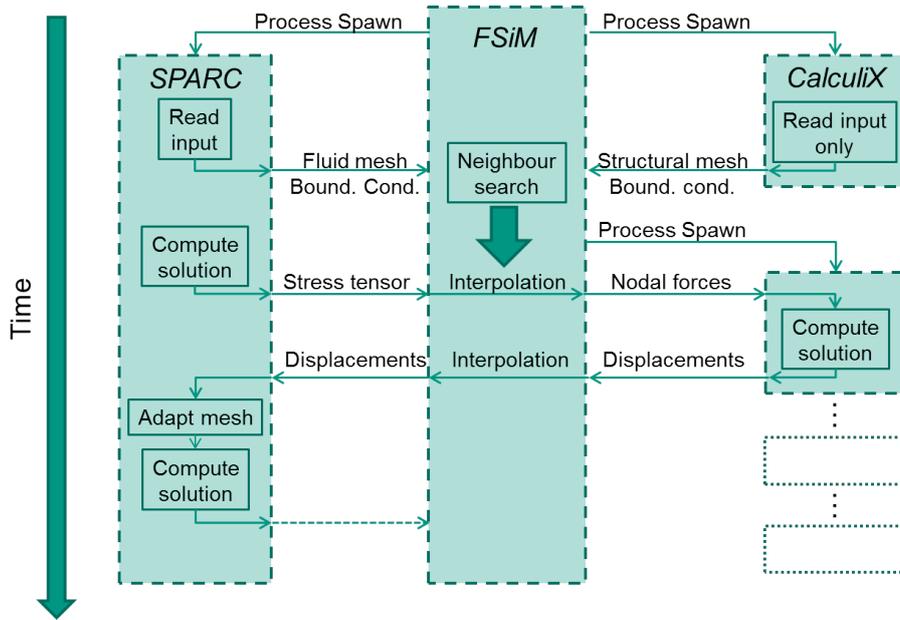


Figure 3.5: Sequence of an FSI computation with the CSS procedure.

skipped. For an FSI computation, this is a desirable feature, but two additional issues must be dealt with when the mesh is refined:

- The displacements must be interpolated to allocate values in the new nodes. This happens in an analogous way as for the solution variables.
- The neighbour search must be done again for every finer mesh. Since the neighbour relations remain unchanged along the computation, this can be done either in the original or in the current (deformed) configuration. In *FSiM* this is done in the original configuration.

3.3 Implementation in MPI

So far, the software framework and the general strategy of the coupling tool have been presented. This section exposes the most important details related to the implementation in MPI. In coincidence with the partitioned approach, it is in general desirable to handle both part-solvers' code as much as possible as a black-box. Nevertheless, depending on how modularly the source code is structured, the modification of some core routines might be unavoidable. In general, the more efficient a program is intended, the deeper the modifications have to be. Figure 3.5 shows the implementation in *FSiM* of the CSS algorithm presented in section 2.5.3. As shown there, unlike *CalculiX*, *SPARC* runs a single time along the whole computation, avoiding recurrent reads of input files and memory allocation. This spawning scheme is not fully consistent with the black-box paradigm and a deeper intervention in the *SPARC* code was needed. This, however, was not problematic since enough experience was available. Fortunately, since structural models have usually relatively few cells, it is not critical to spawn *CalculiX* several times.

The modular approach should be exploited also as regards the input files. It would be senseless to have the coupling manager read in the input files again; that task belongs to each part-solver. Rather, the arrays where the required data (mainly mesh and boundary conditions) is stored are identified in the part-solvers and transferred to the manager.

Some other tasks, however, which might be thought to be part of the FSI and therefore to correspond to the coupling manager, do actually belong naturally to one of the part-solvers. This is the case of the mesh adaptation, which must be dealt with within the fluid solver, for two reasons:

- The parallelization of the fluid solver implies the distribution of all data along the computing cluster nodes memory. To gather all geometric data of the fluid mesh, process it centrally in the manager and redistribute it again, would cost extra communication and would increase memory demand on the manager due to the duplicated data.
- The mesh update procedure is highly dependent on the numerical method of the particular fluid solver.

An heterogeneous framework resulted: on the one hand, *SPARC* and *FSiM* are programmed in FORTRAN whereas *CalculiX* has been developed in C. On the other hand, *SPARC* has a pure MPI parallelization while *CalculiX* takes advantage of the multithreaded version of the SPOOLES solver.

Once *FSiM* is running, *SPARC* is spawned first with the desired number of processes, returning a handle to the new communicator between *FSiM* and *SPARC*. The communicator is an abstract construct of MPI to group up processes for the communication. For the use in network clusters, an info object needs to be created to account for the machine files:

```
CALL MPI_INFO_CREATE(info, ierror)
CALL MPI_INFO_SET(info, 'hostfile', './sparc_mf', ierror)
CALL MPI_COMM_SPAWN(sparc.bin, MPI_ARGV_NULL, 4, &
info, 0, MPI_COMM_SELF, intercomm, array_of_errcodes, ierror)
```

Within *SPARC*, a pure MPI parallelization has been developed throughout the years, using a collection of blocking, non-blocking and collective communication. A master-worker approach had been followed, i.e., the master process is in charge of all input and output and therefore distributes and collects data from all other processes. It is then natural to have only the master communicate with *FSiM*. In the *SPARC* code, right after the `MPI_INIT` line `MPI_COMM_GET_PARENT` is executed to retrieve the communicator between *SPARC* and *FSiM*. Since *SPARC* might also work independent from *FSiM*, it is practical to inquire the value of the communicator. A value `MPI_COMM_NULL` marks standalone operation.

The blocking synchronous send is adopted for all communication between *SPARC* and *FSiM*, for safety and simplicity reasons. For a parallel staggered algorithm, non-blocking communication might be of convenience.

Finally, before `MPI_FINALIZE` the communicator must be freed from both sides using `MPI_COMM_DISCONNECT`.

In *CalculiX* the same strategy is followed, but an additional control variable is passed from *FSiM* to let *CalculiX* know if mesh and boundary conditions, or the displacement of the current time are to be transferred, as shown in figure 3.5. Although *CalculiX* introduces a multithread section due to the *SPOOLES* equation solver, there is no need for the special command `MPI_INIT_THREAD`. This is because no MPI commands take place during the execution of *SPOOLES*.

Despite the successful implementation, some issues result from the fact that existing solvers have not been designed to be coupled with other programs. One aspect is related to the text output. With the above described MPI implementation, it is not possible to control the output appropriately and the messages of *SPARC*, *FSiM* and *CalculiX* are printed in a non-deterministic way. The other aspect is the error handling.

Since this is solved in each code in a different, non-centralized way, it is not possible to inform the other two programs if one of them has encountered an error and stops. A more detailed discussion of this issues has been done by the author in a separate publication [130].

3.4 Mesh adaption

Along the Fluid-Structure-Interaction process, the fluid mesh must be continuously adapted to the deformation of the structure. Block-structured meshes present a series of advantages in this regard:

- Fast and robust algebraic algorithms can be used. The basic algorithm used for the present work is the Transfinite Mapping, already introduced in section 2.3.1
- The blocks allow to easily define different zones to reach a good control over the mesh deformation
- Some blocks can be completely excluded from the mesh deformation, saving computational effort

Although it is intended to obtain modified coordinates for the mesh, the whole interpolation is done on the displacements. This is more convenient, for example, to combine FSI with other mechanisms that modify the mesh, such as rotations. The kinematic continuity, eq. 2.21, provides the displacements of the fluid mesh at the interface $\mathbf{d}^f|_{\Gamma}$. With the Transfinite Mapping, the displacements in the interior nodes of the fluid mesh $\mathbf{d}^f|_{\Omega}$ are computed thus obtaining:

$$\mathbf{d}^f = \mathbf{d}^f|_{\Gamma} \cup \mathbf{d}^f|_{\Omega} \quad (3.1)$$

where \mathbf{d}^f are the displacements at every fluid node. To obtain the updated mesh coordinates at an arbitrary time t_n , the displacements at time t_n are added to the original mesh coordinates at t_0 :

$${}^{t_n}\mathbf{x}^f = {}^{t_0}\mathbf{x}^f + {}^{t_n}\mathbf{d}^f \quad (3.2)$$

Since almost any fluid mesh consists of more than one block, a strategy is needed to determine how the deformation of the structure is propagated from the interaction wall to the blocks away from it, in the correct sequence. The first step is to sort out the blocks that participate in the mesh adaption. For this purpose, the concept of *block generations*, introduced by Bardossy [4], is adopted, as depicted in figure 3.6. The different block generations are defined in the following way:

First generation: blocks facing the interaction surface

Second generation: blocks that do not belong to the first generation, and have a face connection to a first generation block

Third generation: blocks that belong neither to the first nor to the second generation, and have a face connection to a second generation block

Fourth generation: blocks that belong neither to the first nor to the second and third generation, and have a face connection to a third generation block

The remaining blocks do not participate in the mesh adaption. This approach has proven very flexible and can accommodate many different topologies. Three variants have been developed to control the mesh deformation:

- For simple cases, like the one in figure 3.7, it is sufficient that the deformation of the structure be “absorbed” by the block layer next to the interaction surface. In this example, the first block layer is designed broad enough to avoid a too high cell-volume change.
- In some topologies, it might be convenient that the first block layer be thin to resolve only the boundary layer. The thickness of this block layer is then not enough to accommodate the deformation of the structure. To solve this problem, Bardossy [4] introduced the possibility that every block-side opposite to the interaction surface move parallel to the latter, thus preventing the thin blocks to collapse, as can be appreciated in figure 3.8. The deformation of the interface is then absorbed by the second block layer.
- For complex three-dimensional block topologies as the ones required by turbomachinery, enhanced mesh control is necessary. A turbomachine’s blade is typically meshed with the help of a “C-Grid”. Figure 3.9 illustrates this with the example of a wind turbine blade. The blade profile is resolved with a “C-Grid” and two “O-Grids” around it, an “H-Grid” filling the rest of the computational domain, see figure 3.9a. Since only the yellow blocks (Figure 3.9b) would move with the blade, the trailing edge part of the “C-Grid” (red line in Figure 3.9c) would be quickly distorted. The solution is to force the yellow-striped blocks to belong to the first generation and the creation of a fictive interaction surface, as shown in figure 3.9c. This block-side receives the displacement of the node in the trailing edge. Additionally, the block-sides between the second and third “O-Grids” (blue line in figure 3.9c) are assigned the displacement of the interaction surface with a relaxation factor. In this way a proper behaviour in the mesh adaption is achieved, as can be appreciated in figure 3.10.

The Transfinite Mapping algorithm needs information of one dimension lower than the geometrical object to be modified. To modify the coordinates of a block-edge, two points are needed; to modify a block-side, the four bounding edges are needed; to modify a block, the six block-sides are required. For this reason, the global mesh adaption algorithm works in a sequential way, first adapting edges, then block-sides, and then the block interior coordinates. This also requires sorting out the block-edges and block-sides that

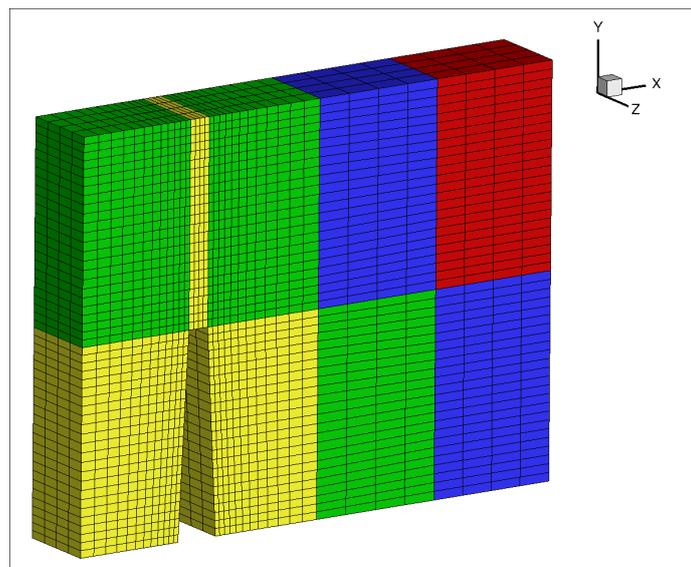


Figure 3.6: Block generations. 1st gen.: yellow; 2nd gen.: green; 3rd gen.: blue; 4th gen.: red

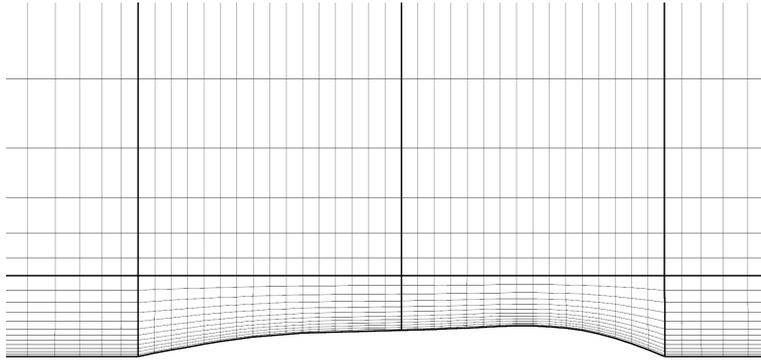


Figure 3.7: Simple mesh deformation treatment. Only the first block layer absorbs the interface movement.

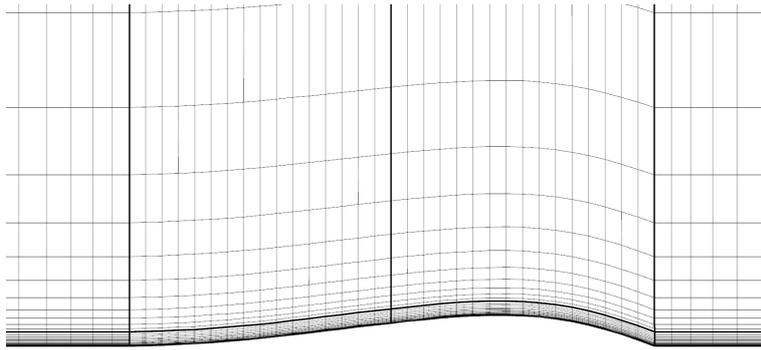
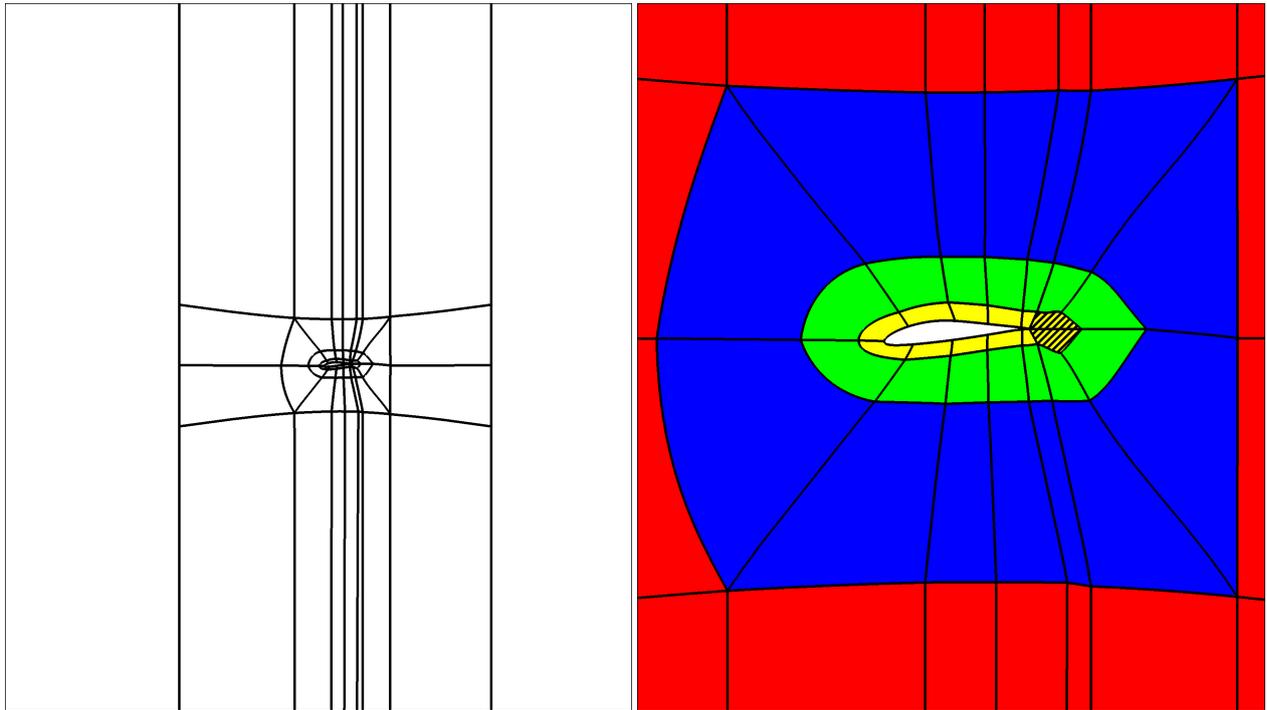


Figure 3.8: Improved mesh deformation treatment. Boundary layer resolution is preserved.

do not need to be adapted. A certain degree of parallelization is easily given, due to the fact that for each block of a certain generation, only information of the previous generation is needed, allowing all other blocks of the current generation to be updated in parallel. Between one generation and the next, the modified coordinates are transferred taking advantage of the dummy cells. This is done using a simultaneous global communication that exchanges the content of the dummy cells for every block. A synchronization happens then naturally, preventing any process from beginning with the next generation before the other processes have finished. Figure 3.11 illustrates the mesh update mechanism. Initially, only the displacements $\mathbf{d}^f|_{\Gamma}$ at the interface are available (a). The necessary edges of the first generation blocks are then updated (b), followed by the sides and interior nodes (c). The displacements on the corresponding block-sides are transferred from first to second generation blocks (d). The necessary edges of the second generation blocks are updated (e), followed by the sides and interior nodes (f). The process continues with the third and fourth generation blocks in the same manner.

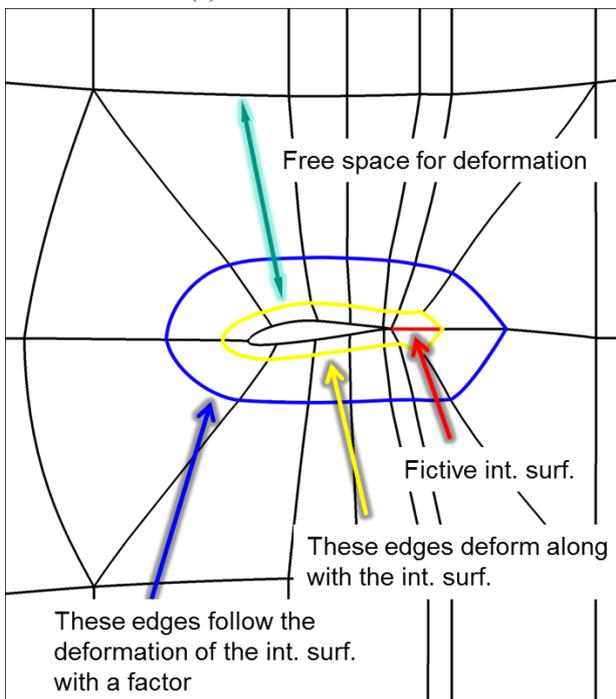
3.5 Interface treatment

As introduced in section 2.5.5, the interface treatment can be divided into a neighbour-search and an interpolation stage. These are most conveniently placed in the coupling manager, since data from both solvers is needed for that purpose. Additionally, with the condition that virtual works are to be conserved, both directions of data exchange are not independent and must be dealt with centrally. There is a parallelization potential in these interpolation algorithms which is, however, only of secondary importance since the computational cost is relatively small.



(a) Overview of H-Grid.

(b) Detail of O-Grid with block generations.



(c) Detail of block-boundaries motion.

Figure 3.9: Mesh deformation treatment for a complex blade topology

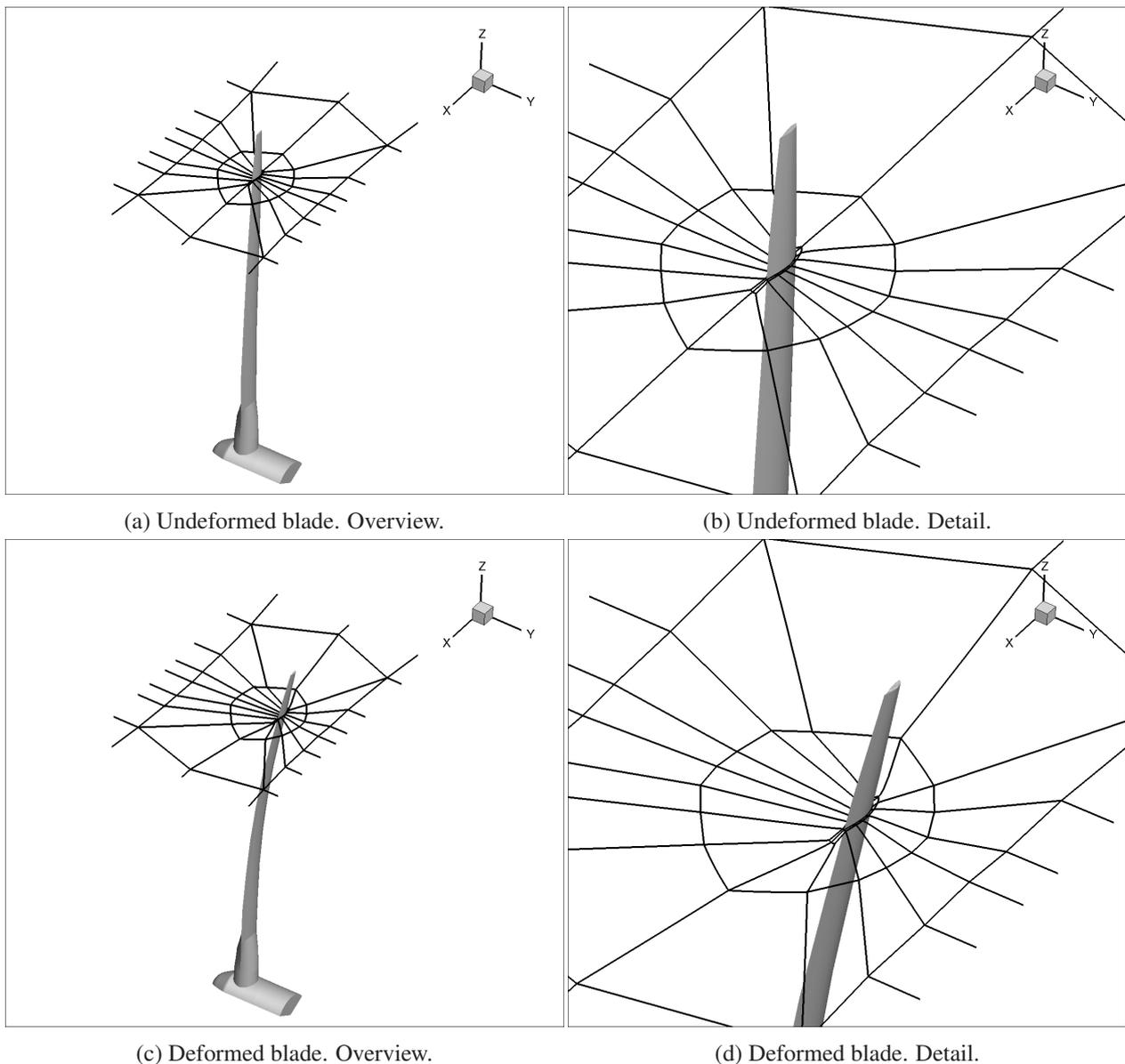


Figure 3.10: Mesh deformation treatment for a wind turbine blade.

3.5.1 Neighbour search

The neighbour search needs to be done only once at the beginning of the computation because, unlike for example a contact problem in structural mechanics, in FSI there is no relative motion between meshes. Because of this, the efficiency of this stage is not a crucial issue. The corresponding weight factors for the later interpolation are most conveniently obtained at this stage, since all necessary arrays are already present. For the calculation of the weight factors two geometries have been considered: triangular faces and quadrilateral faces.

In the present work, a local neighbour-search initially developed by Bardossy [117] is applied. Depending on how the interpolation is performed, different data sets relating the fluid and structural meshes are required. This gives rise to two search methods: one based on the fluid nodes, and one based on the structural element faces.

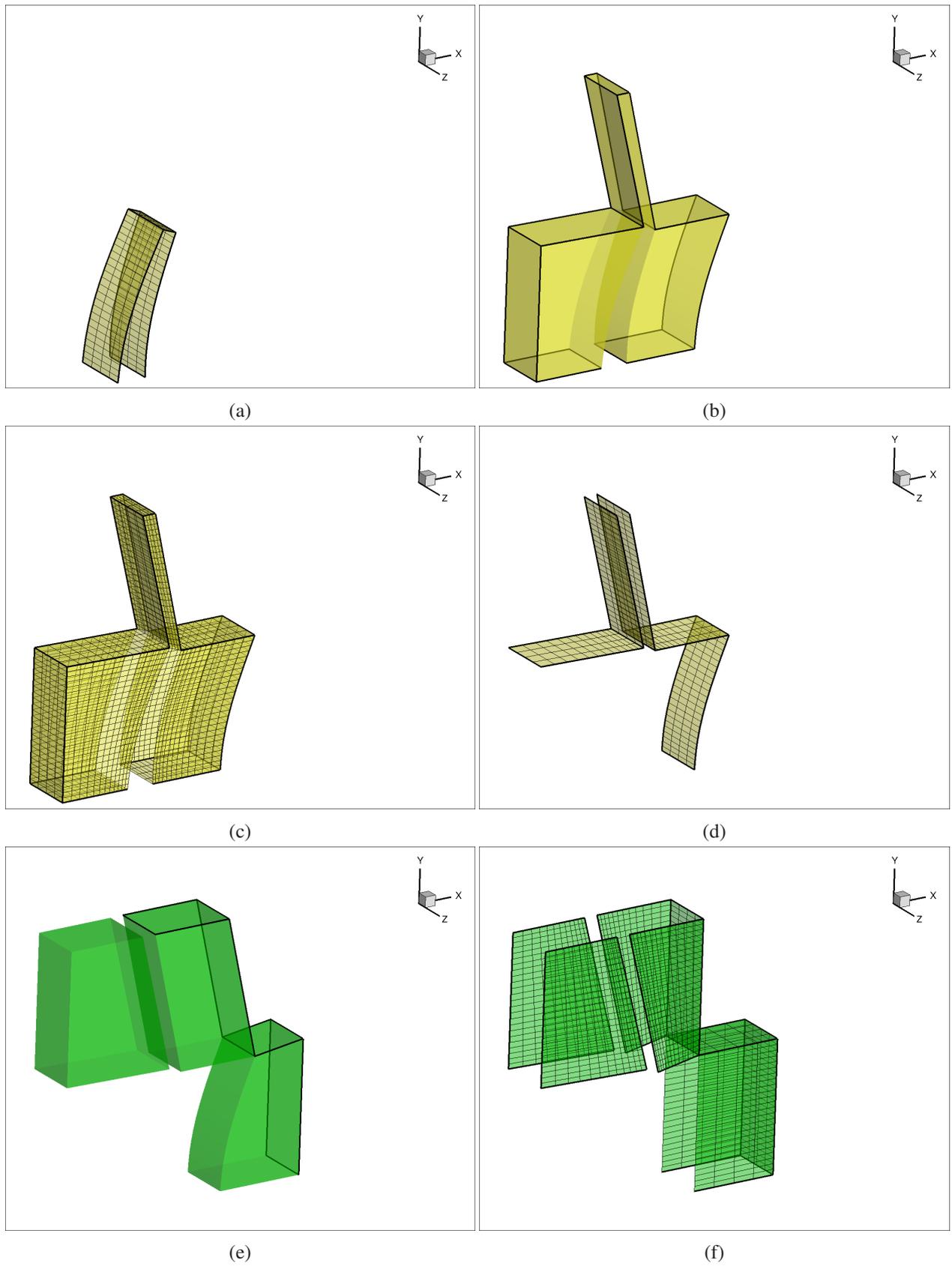


Figure 3.11: Propagation of interface displacements across the block structure.

Fluid-node-based search

For the displacement interpolation (see section 3.5.2) the same matching information is needed as for the conservative force interpolation. Figure 3.13a illustrates the algorithm. For each node in the CFD mesh, a structural element is to be found. For this purpose, a previously created data array is used, which contains the adjacent elements to every structural node in the corresponding Interaction Surface Pair. Then, the CFD node is projected onto the structural element-face using the element-face normal, and the weight factors w_i^s for the structural nodes i are obtained using linear or bilinear FE form functions, for triangular or quadrilateral element-faces respectively, as shown in figure 3.13b. The resulting association can be clearly visualized by lines linking the fluid nodes and the structural element-face midpoint, as shown in figure 3.15.

Structural-face-midpoint-based search

For the pressure interpolation (see section 3.5.2), the approach is to obtain a pressure to be applied to the midpoint of a structural element-face. The weight factors w_j^f for the fluid nodes j are obtained using bilinear quadrilateral FE form functions, as shown in figure 3.14. The resulting association can be clearly visualized by lines linking the structural element-face midpoint and the corresponding fluid cell-face midpoint, as shown in figure 3.16.

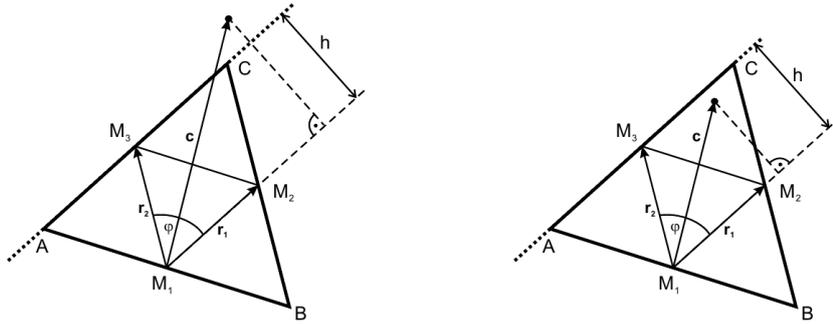


Figure 3.12: Algorithm to find a point in a triangle. Taken from [4]

Geometrical search

A central component of the neighbour-search is the algorithm to determine if a point is contained in a triangle, based on determinant comparison. The algorithm can be better understood with help of figure 3.12, and consists of the following steps:

1. Determine the mid-side points of the triangle. Choose one of them.
2. Create \mathbf{r}_1 and \mathbf{r}_2 from the mid-side point to the other two, and to point c .
3. Calculate $\Delta_1 = \mathbf{r}_1 \times \mathbf{r}_2$, which defines the area of a parallelogram.
4. Calculate $\Delta_2 = \mathbf{r}_1 \times \mathbf{c}$.
5. If $\Delta_2 \leq \Delta_1$, the point lies at most $h = |\mathbf{r}_2| \cdot \sin(\varphi)$ away from \mathbf{r}_1 . Otherwise, c is outside the triangle.

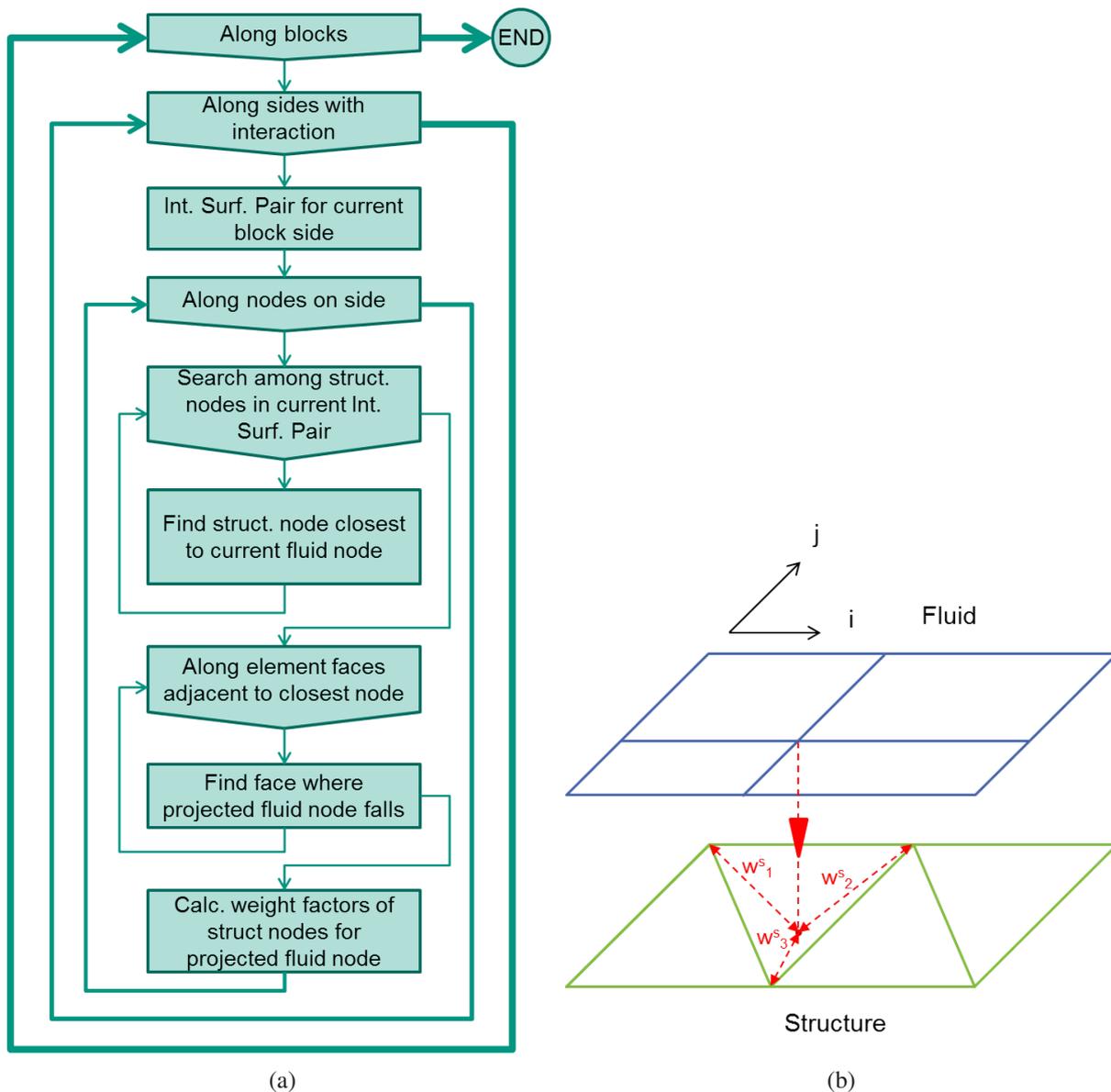


Figure 3.13: Fluid-node-based neighbour search.

6. If the point is not outside, the same procedure must be repeated with the other mid-points.

The procedure has proven very robust and can be easily extended to cope with any geometry by triangulation of the mesh.

3.5.2 Interpolation

Once the mesh objects (nodes and cell- and element-faces) that participate in the interpolation have been matched, the interpolation of variables from one mesh to the other can be dealt with. Three kinds of interpolation are implemented in *FSiM*:

- Conservative force interpolation
- Pressure interpolation
- Displacement interpolation

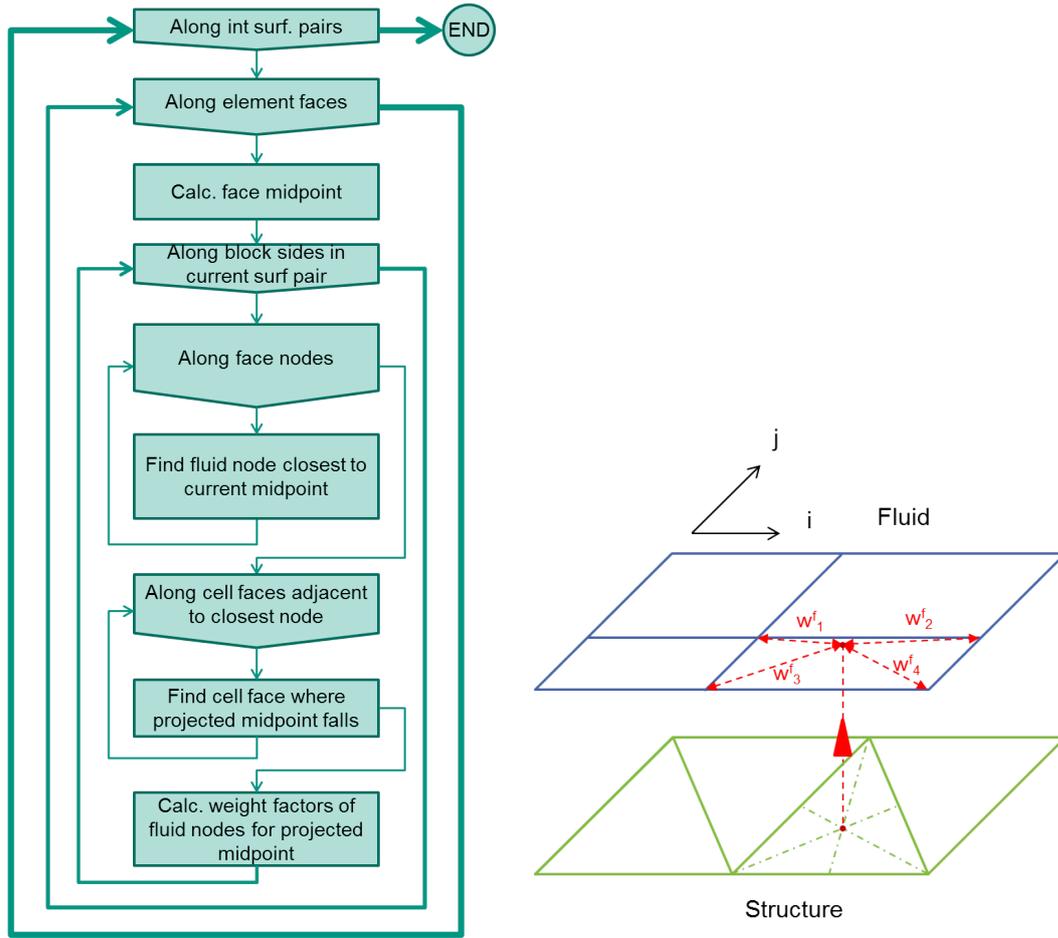


Figure 3.14: Structural-face-midpoint-based neighbour search

Conservative force interpolation

The first kind of load interpolation which is implemented in *FSiM* is the Conservative Force Interpolation introduced in section 2.5.5, whose algorithm is shown in figure 3.17. The first step, done within *SPARC*, is to determine the fluid stress tensor σ_m on every cell-face midpoint on the interaction surface. For that purpose, the static pressure is obtained by averaging the pressure in the last cell and the dummy cell, and the shear stresses by computing the appropriate velocity gradients. Then, the force in the midpoint of the cell-face is obtained as:

$$\mathbf{f}_m^f = \sigma_m \cdot \mathbf{S}_m \quad (3.3)$$

where \mathbf{S}_m is the area vector of the fluid cell-face. This force is then distributed into the four fluid corner nodes of the cell as:

$$\mathbf{f}_c^f = \frac{\mathbf{f}_m^f}{4} \quad (3.4)$$

Finally, the forces in the structural nodes are computed using the already determined weight factors:

$$\mathbf{f}_i^s = w_i^s \mathbf{f}_c^f \quad (3.5)$$

where $i=[1 \text{ to number of face nodes}]$.

Pressure interpolation

The conservative force interpolation presents some problems when the fluid cells are larger than the structural elements, as happens when coarser meshes are used with the multigrid method. In that case, not every structural element would receive forces, leading to an unphysical load distribution. For that reason, a pressure interpolation was developed, shown in figure 3.18.

For simplicity, a constant pressure distribution acting on the whole structural element-face is assumed. The stress tensor in the element-face midpoint is used as a representative value for the face. This can be interpreted as using a single Gauss point for the integration of the load distribution on the FE face. More Gauss points could be used to achieve a higher accuracy. The process then begins by projecting the structural face midpoint onto the fluid mesh. The fluid stress tensor in the projected point P is interpolated from the stress tensors in the surrounding fluid nodes as:

$$\sigma_p = \sum_j w_j^f \sigma_j \quad (3.6)$$

With the representative constant tensor σ_p and the area vector of the structural element-face \mathbf{S}_p , the resulting force acting on the element-face is calculated as:

$$\mathbf{f}_m^s = \sigma_p \cdot \mathbf{S}_p \quad (3.7)$$

Due to the large-displacements assumption, the change in geometry cannot, in general, be neglected. This affects the calculation of the area, because of what the current and not the original coordinates must be used for this.

The last step is to compute the finite element consistent forces in the face nodes. An accurate computation takes into account the distortion of the element face. For simplicity, it is assumed that the element faces are undistorted. In that case the nodal forces result [33]:

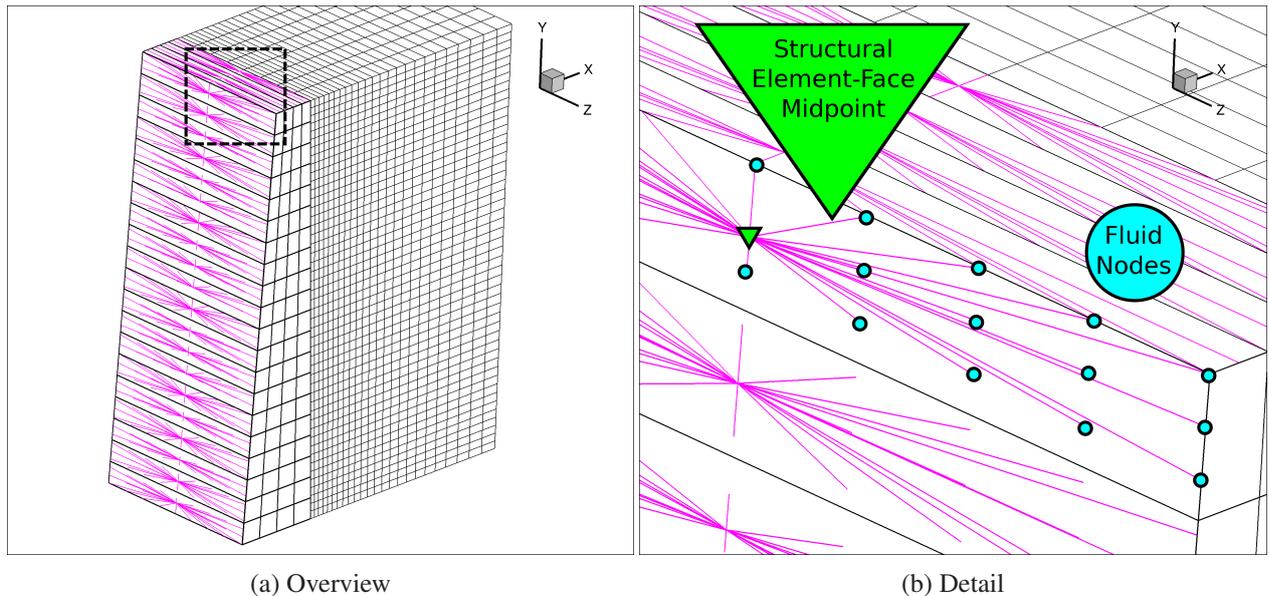


Figure 3.15: Neighbour associations. Linking lines between fluid nodes and structural element-face mid-point.

For triangular linear elements:

$$\mathbf{f}_m^s/3 \quad \text{for all nodes}$$

For quadrilateral linear elements:

$$\mathbf{f}_m^s/4 \quad \text{for all nodes}$$

For triangular parabolic elements:

$$\begin{aligned} 0 & \quad \text{for corner nodes} \\ \mathbf{f}_m^s/3 & \quad \text{for mid-side nodes} \end{aligned}$$

For quadrilateral parabolic elements:

$$\begin{aligned} -\mathbf{f}_m^s/12 & \quad \text{for corner nodes} \\ \mathbf{f}_m^s/3 & \quad \text{for mid-side nodes} \end{aligned}$$

3.5.3 Displacement interpolation

Lastly, the algorithm for displacement interpolation is presented in figure 3.19. The displacement \mathbf{d}^f of one fluid node is interpolated from the displacements of the neighbouring structural nodes \mathbf{d}_i^s by means of the weighing factors w_i^s :

$$\mathbf{d}^f = \sum_i w_i^s \mathbf{d}_i^s \quad (3.8)$$

3.6 Rotating models

The simulation of Fluid-Structure Interaction on rotating models comprises several techniques that must be combined in a physically meaningful way. Depending on the degree of accuracy of the intended investigations, simple or complex models can be applied. Table 3.2 summarizes the physics that can be resolved

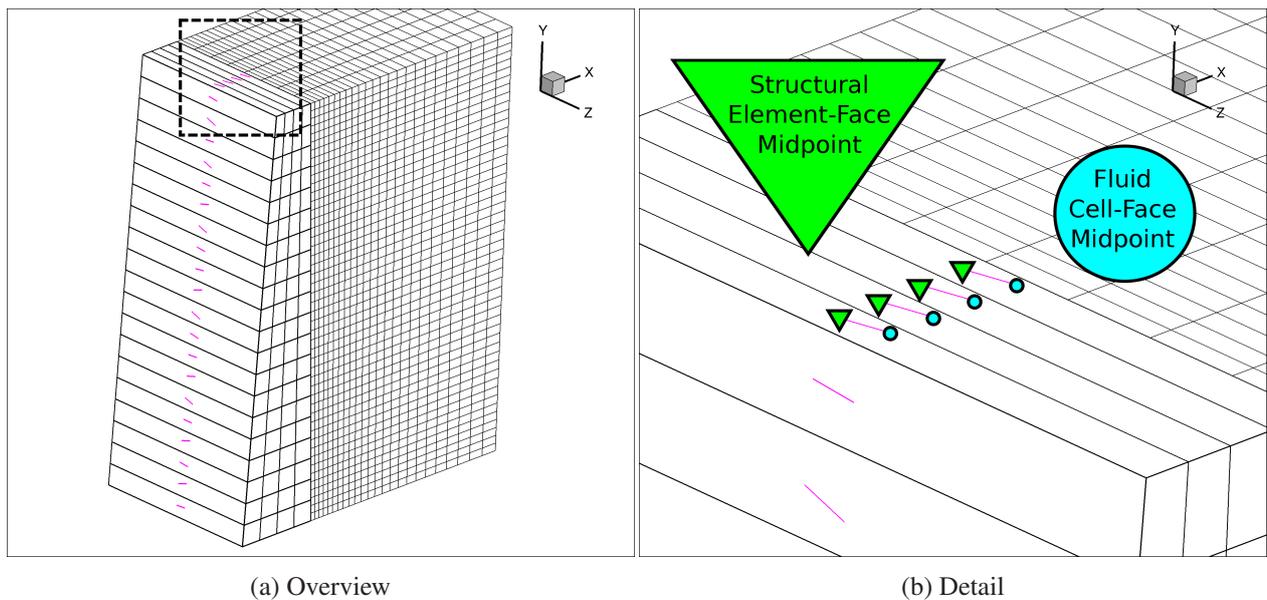


Figure 3.16: Structural-face-midpoint-based neighbour search. Linking lines between structural element-face midpoint and fluid cell-face midpoint.

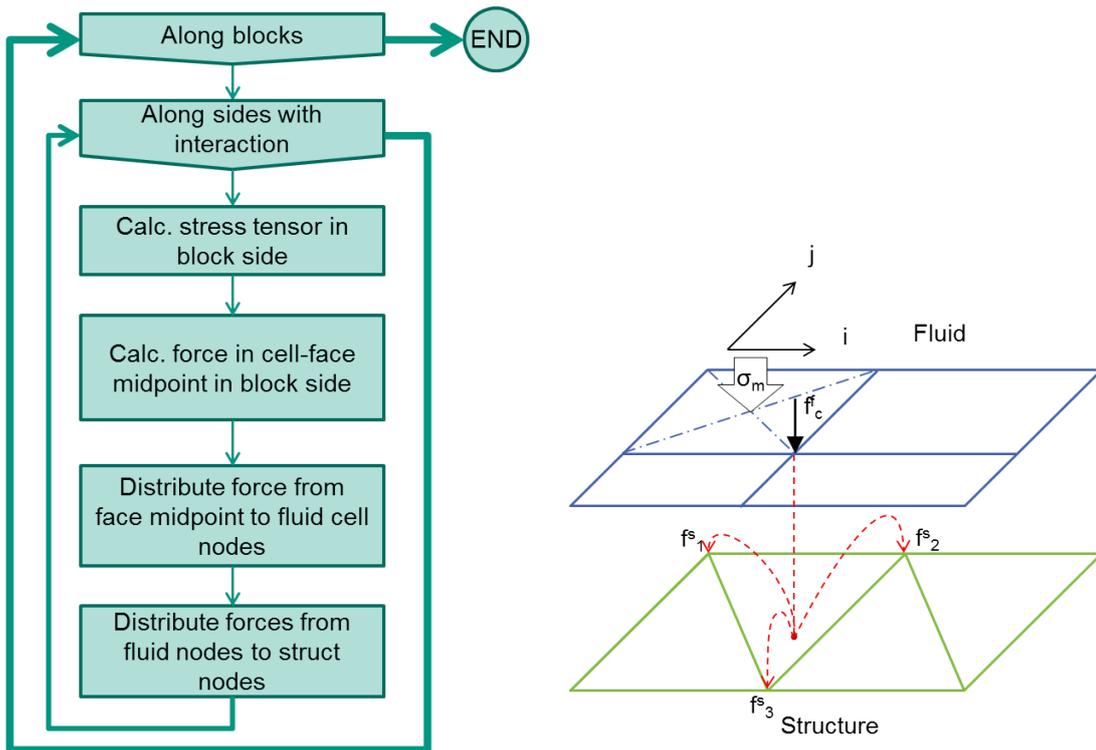


Figure 3.17: Conservative force interpolation.

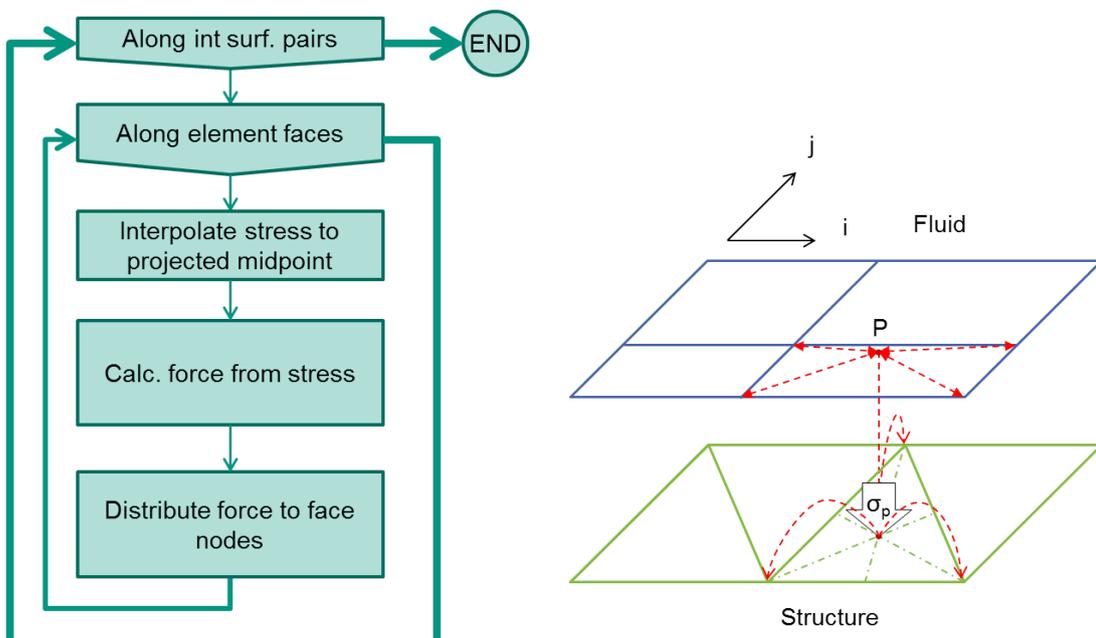


Figure 3.18: Pressure interpolation.

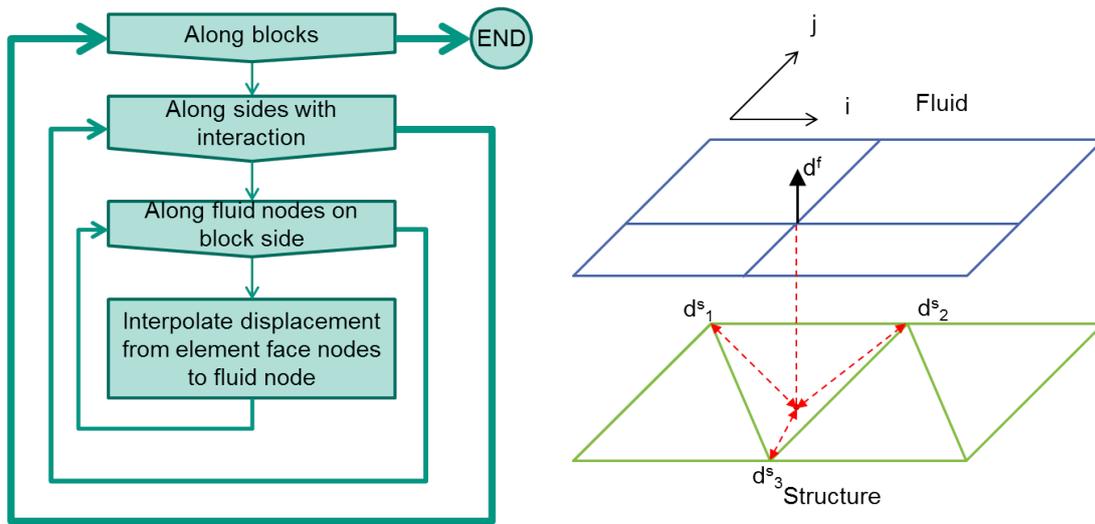


Figure 3.19: Displacement interpolation.

Model	Resolved Physics	
	Simplified	Complex
One-way	Structure with real loads	
Two-way		Change in geometry relevant for fluid
Steady-state fluid/static structure	Steady-state deformation and stresses. Axisymmetric body loads only (centrifugal)	
Unsteady fluid/dynamic structure		Fluid induced vibration. Unsymmetric body loads (gravitation)
Frozen rotor	Constant rotational velocity	
Rotating mesh		Rotational accelerations
Single rotor section	Periodic flow	
Full rotor		Asymmetric flow
Rotating domain only	Homogeneous inflow	
Rotating + stationary domain (sliding interface)		Inhomogeneous inflow
RANS Model	Average loads	
DES/LES		Effect of upwind turbulent structures

Table 3.2: Resolved physics for the different model complexity.

with the different models applicable to Fluid-Structure Interaction on rotating machinery. With the exception of rotational accelerations, all of these modeling capabilities are supported by the present implementation.

Once the part-solvers have been equipped with the appropriate formulations to handle rotations, it must be ensured that the motion of both fields is made compatible. Two kinds of rotating models can be considered for turbomachinery:

Constant rotating velocity In most investigations of a turbomachine, the motor or generator can be assumed to impose a constant rotating velocity. This information is given separately to each part-solver, no synchronization being needed between them. For an FSI model, this requires little additional implementations in the fluid solver in comparison with an uncoupled CFD computation. This is the approach used in the present work.

Free rotation A more accurate analysis requires an FSI simulation, whereby the balance between the fluid loads, the structural elasticity and inertia, and the effect of external forces (to account for the motor or generator) results in an instantaneous acceleration. Meaningful transient phenomena can be captured in this way. However, some technique must be devised in order to communicate the (non-constant) rotation to the fluid domain, resulting in a more complex implementation.

For the loads, the compatibility means to ensure they are applied to the structure with the correct direction. For the displacements, the compatibility means that for the adaption of the fluid mesh, the displacements are to be transferred without the rotational component. This is because in *SPARC*, the rotation is handled separate from the FSI displacements.

As explained in previous sections there are two ways to handle rotating models: a rotating mesh or a rotating frame of reference. These two alternatives can be combined resulting in four different cases:

Fluid: Rotating Frame of Reference / Structure: Rotating Frame of Reference (RFR/RFR)

This is the easiest variant. The only changes needed for FSI with respect to a non-rotating system is the addition of the rotational velocity to the velocity due to the FSI deformation of the interactions walls.

Fluid: Rotating Mesh / Structure: Rotating Frame of Reference (RM/RFR)

Here the structural deformations must be “rotated forward” to match the fluid configuration. Either the fluid stress tensor or the structural nodal forces must be transformed back to the non-rotated (but deformed) position.

Fluid: Rotating Frame of Reference / Structure: Rotating Mesh (RFR/RM)

In this case, the structure has to be “rotated back” to match the fluid configuration. Either the fluid stress tensor or the structural nodal forces must be transformed to the rotated (and deformed) position.

Fluid: Rotating Mesh / Structure: Rotating Mesh (RM/RM)

The rotation has to be subtracted from the structural displacements for the purpose of fluid mesh adaption and is added again to obtain the rotated and deformed fluid mesh.

For a better understanding, the case RM/RFR is elaborated upon with the example of a 2D-rotor, figure 3.20. For simplicity, only one section of the rotor is considered in figure 3.21. The structural displacements, figure 3.21b, are used to update the fluid mesh (c). Within *SPARC*, the deformed mesh is rotated (d) by

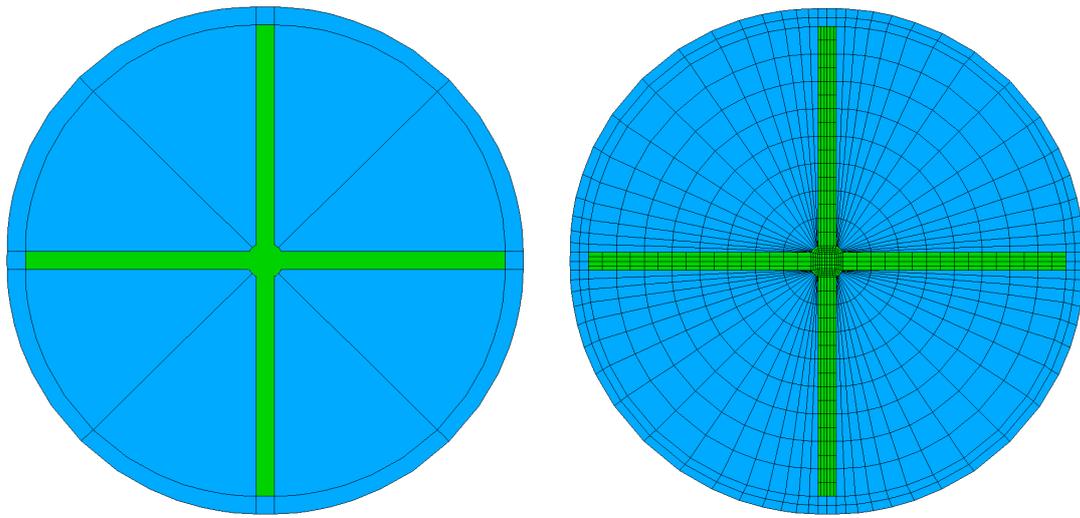


Figure 3.20: Block topology and meshes for a rotor

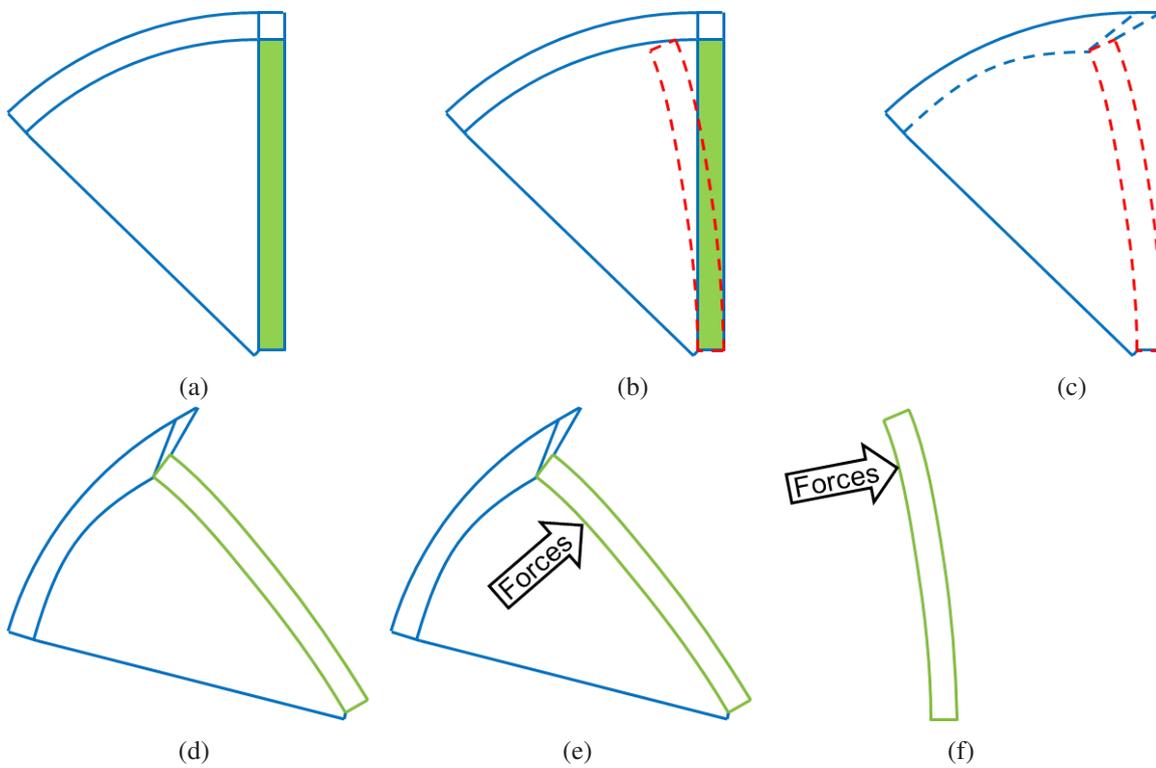


Figure 3.21: Sequence to handle rotating models

means of applying a rotation transformation to the deformed fluid mesh coordinates (assuming a rotation around the z-axis):

$${}^{t_n}\mathbf{x}_{rot}^f = \mathbf{R} \cdot {}^{t_n}\mathbf{x}^f \quad (3.9)$$

$$\text{where } \mathbf{R} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \phi = \omega t.$$

Then, one time step is solved and the forces are calculated (e). Finally, the forces are rotated back in *FSiM* to be applied to the structure (f), using a similar transformation:

$${}^{t_n}\mathbf{f}_s = \mathbf{R} \cdot {}^{t_n}\mathbf{f}_{s,rot} \quad (3.10)$$

with $\phi = -\omega t$.

The rotation of the stress tensor could also be used, but is more costly than the rotation of the force vector.

3.6.1 Measurement of moments

Torsional vibrations that affect mechanical components such as bearings and gears play a major role in the life-time of a turbomachine. From the standpoint of an FSI simulation, it is possible to predict the torsional loads with a high degree of accuracy. In a torsional system, the relevant load for the study of vibrations is the moment in the shaft of the turbomachine M_R , which the drive has to provide, as shown in figure 3.22a. In the drive, all mechanical elements and the motor or generator are found. For most applications, the drive can be considered to prescribe a constant rotating speed to the shaft of the turbomachine. The vibration displacements d are then measured with respect to a reference coordinate system η and ξ that rotates with the same constant velocity as the drive. The shaft moment is a result of the FSI process and therefore, the relationship between the exciting moment M_F and the reaction moment M_R on the shaft is not self-evident. A simplified torsional model with a single DOF, presented in figure 3.22b, can help to understand the behaviour of the real system. There, the physical rotor and shaft are modeled as a rigid part, a torsional spring and a damping element, so that the only remaining DOF is the torsional displacement θ . A moment balance can then be made:

$$M_D + M_F + M_S = \ddot{\theta}I \tag{3.11}$$

where

$$\begin{aligned} M_D &= -\dot{\theta}C \\ M_S &= -\theta K \end{aligned} \tag{3.12}$$

and C and K are the damping and stiffness coefficients respectively.

The fluid moment M_F is obtained by integration of the fluid loads within the fluid solver, or by integration of the loads applied to the structure, in the coupling manager. The difference between the two lies on the

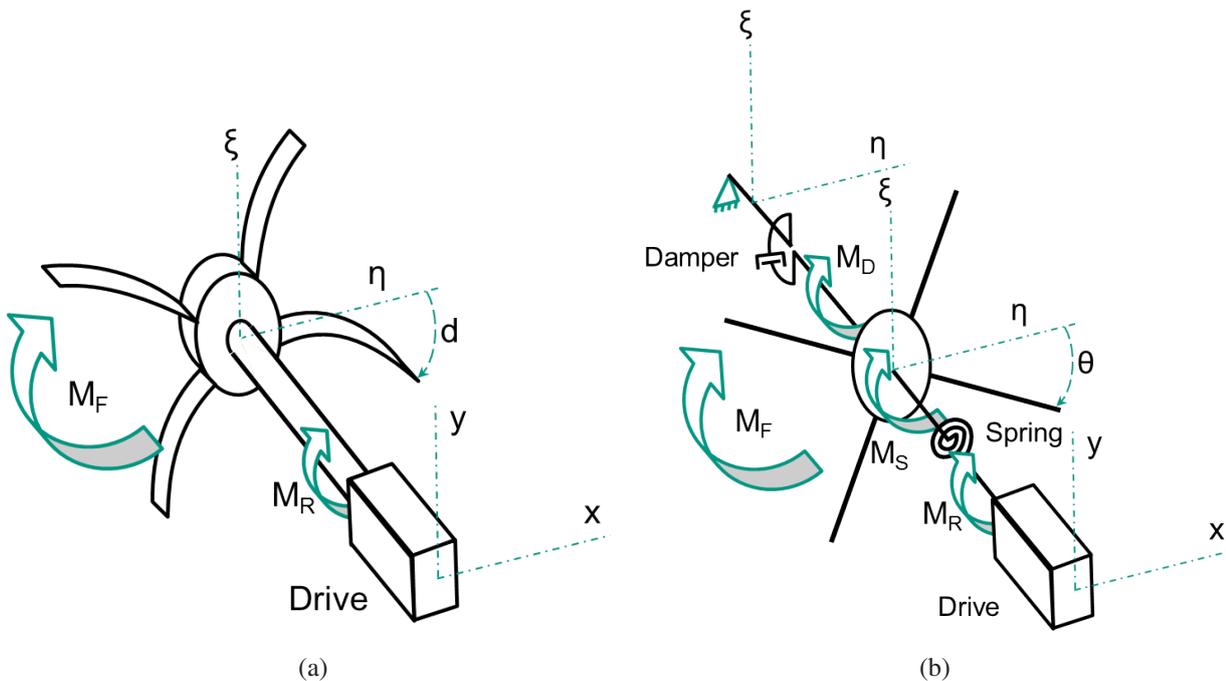


Figure 3.22: Torsional system for a turbomachine. a) Real system. b) Simplified equivalent system.

accuracy of the interface treatment.

In a simulation, the reaction moment M_R is directly obtained from integration of the reaction forces in the fixed DOF of the structural model, but as seen in figure 3.22b, it can also be related to the spring moment and therefore to the torsional displacement θ :

$$M_R = M_S = -\theta K \quad (3.13)$$

Therefore, in general, the reaction moment is a result not only of the acting fluid moment but also of the rotor dynamics, which determine the instantaneous value of θ . In the particular case of a rigid rotor, the terms M_D and M_S can be disregarded, and equations 3.11 and 3.13 then result in:

$$M_R = -M_F \quad (3.14)$$

That means, when rotor dynamics are neglected the shaft reaction can be directly obtained from the fluid solver output.

All of these effects can be captured by an unsteady, one-way FSI, since the structural dynamics is solved. A two-way FSI additionally accounts for the variations of M_F produced by the deformation of the rotor, as the fluid domain geometry is updated. This becomes important only when the deformation of the rotor is significant or when even small geometrical changes alter the flow significantly. This might happen, for example, when small gaps are present or when self-exciting configurations result, as in flutter.

4 Verification and performance of FSiM

4.1 Alternatives for benchmarking

In the course of development of a simulation tool, there are two instances to check for errors:

1. **Verification:** Verification means to ensure that the tool works as intended, that is to say, that there are no implementation errors.
2. **Validation:** In this second stage, it must be ensured that the used models do not depart too much from physics, in other words, to test the prediction capability of the tool.

A true validation can only be done by contrasting the implementation results against experimental data. Particularly in the context of Fluid-Structure Interaction, the design of an appropriate test case is challenging. From the numerical standpoint, two-dimensional test cases are desired but two-dimensional measurements are generally very difficult. Side-wall effects in experiments, such as those conducted by Schäfer et al. [121] and Gomez [131], are hardly numerically reproducible. Also, care must be taken as where the boundary conditions can be applied, and that material properties and behavior be accurately reproduced. Furthermore, for the particular case of a compressible flow solver like *SPARC*, too low velocities, below $Ma = 0.01$ (~4 m/s for air at standard conditions), are problematic. On this account, most cases conceived for incompressible flow are not applicable.

A single experimental test case was found in the literature, by Cerqueira et al. [132], dealing with compressible flows. Thereby, a wave propagation in a shock-tube with a flexible obstacle is reproduced with an FSI simulation. Although the experimental set-up meets the numerical requirements, it was not deemed sufficiently adequate for a verification. In the first place, the publication provides limited information to reproduce the case. Additionally, although the pressure in a reference point and some qualitative comparison of the shock-wave propagation show good agreement, the computed deflection of the obstacle panel departs considerably from the measurements. Finally, the short duration of the experiment does not allow to test a code implementation deeply enough.

An attempt was made at our institute by Mattern et al. [133] to produce appropriate experimental data, taking advantage of the PIV (Particle Image Velocimetry) technology available at FSM. In that case, the difficulties for validation are related with the modeling of the fixities between the plexiglas parts, and the cost of a three-dimensional computation required to appropriately reproduce the flow.

Due to the abovementioned obstacles, it was opted for a verification against numerical test cases. In this procedure, there is an implicit trust that the tool against which the comparison is done does not contain implementation errors itself. In spite of its limitations, this form of verification has the advantage that unmodeled physics can be excluded and a more accurate agreement between the two solutions can be expected. In this way, implementation errors become more obvious.

This strategy has been indeed followed in previous works of Fluid-Structure Interaction. The different authors choose test cases that match the particular requirements of their implementations. For that reason,

Panel Properties	
c	1m
h	2mm
ρ_s	2700 kg/m ³
E	71 GPa
ν	0,34

(a) Panel properties.

Air Properties	
P_0	101306 Pa
U	325.7 m/s
ρ_f	1.225 kg/m ³
T_0	228,15 K
μ	1.789 kg/m.s

(b) Air properties.

Table 4.1: Properties for the panel flutter problem.

many of the test cases found in the literature [43], [49], [51], [134] are also not suitable for the verification of the present development, either because they deal with very low speed flow, they require costly three-dimensional simulations, require a strongly coupled algorithm, or the structure exhibits only rigid body motion, which is not representative of a general flexible structure.

Under the previous considerations, the following two non-rotating test cases were chosen. Each is conceived to test the implemented tool in one of the two dominating mechanisms of fluid-induced vibrations introduced in section 2.2, namely, aeroelastic instabilities and forced oscillations.

Due to the nature of *SPARC*'s compressible formulation, the time-integration resolution is dominated by the fluid solution. As explained in section 2.5.3, in the CSS algorithm the same time-step size is used for both the fluid and structural solver. In all simulations performed, the time-step size was set so as to reach a convergence of the fluid solver within around 20 inner iterations.

4.2 Panel flutter problem

The panel flutter problem is a known case from the aeronautic industry, which can be addressed with a simple two-dimensional geometry. It is a high-speed test case and, therefore, well suited for a compressible flow solver. Figure 4.1 shows the geometry of the problem, resembling a fuselage side panel.

The flexible aluminum panel is pivoted at its ends and its properties are displayed in table 4.1a. One side of the panel is surrounded by air at aeronautic normal conditions at sea level, whose properties are detailed in table 4.1b. A constant pressure P_0 acts on the other side, so that the forces are initially balanced. A transonic, non-viscous flow of $Ma = 0.95$, with a velocity U , exists parallel to the panel, which is initially at rest. If a small perturbation is applied, a traveling wave appears in the panel after a short transient. A shock wave travels with the panel as shown in figure 4.4, indicating a close correlation between the two phenomena that can only be resolved with a two-way FSI. The fluid domain is discretized with approximately 40,000

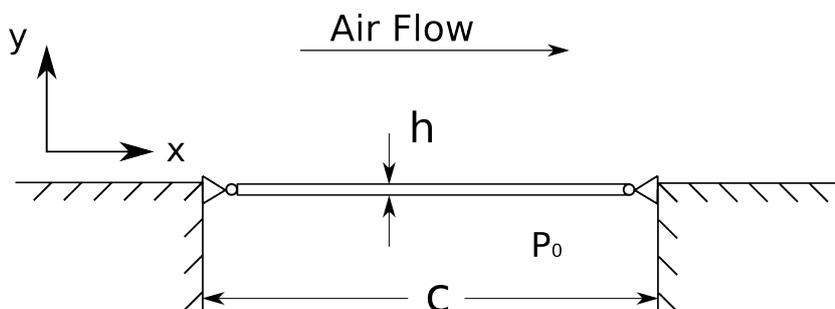


Figure 4.1: Geometry of the panel flutter test case.

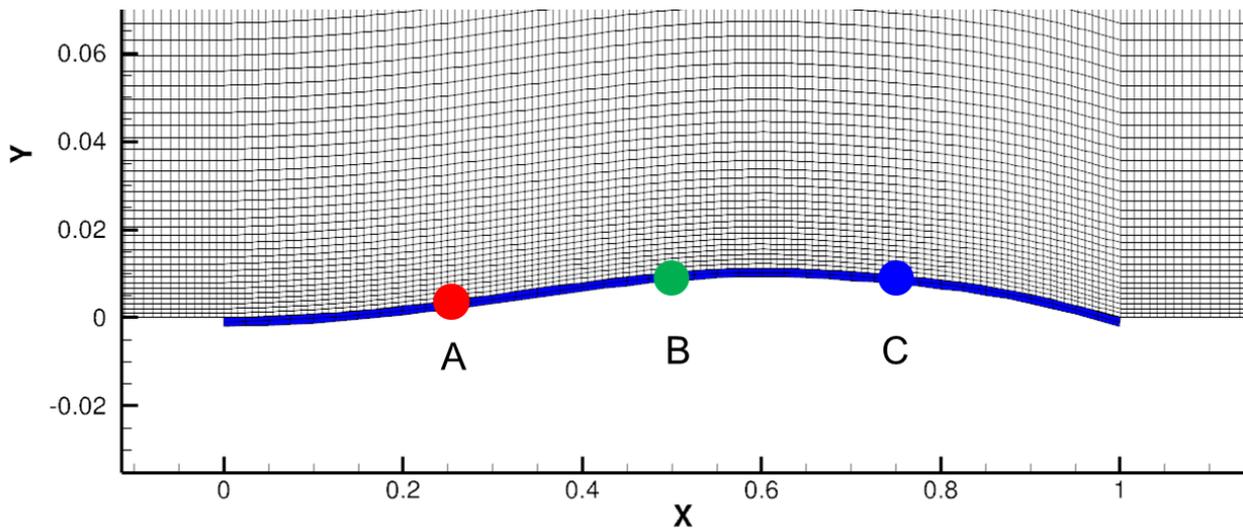


Figure 4.2: Data extraction location for panel deformation.

	Comm. from and to FSiM	FSiM	SPARC	CalculiX
Percentage	7%	2%	82%	9%

Table 4.2: Computational time fraction for each coupling component.

cells and extends 20 times the chord length in all three directions, namely, after, before and above the panel, to minimize the influence of wave reflection on the boundaries. The structure is resolved with 160 8-node solid bilinear elements. Two layers of elements were used in the thickness to resolve the bending mode better. The deflection is taken at three points along the panel, at one-quarter, one-half and three-quarters of the chord, as displayed in figure 4.2. The simulation with *FSiM* is compared with the reference computation by *Massjung et al.* [93], resulting in a 3% shorter period and 5% smaller amplitude. For a better comparison, the *FSiM* results are stretched 3% in time, as shown in figure 4.3, from where it can be concluded that both simulations are in very good agreement. *Massjung et al.* in turn, compared their own results with those of *Bendiksen et al.* [135], reporting a 6.4% longer period. Our results, therefore, lie between those of the cited sources.

In order to reach steady-state oscillations several periods need to be resolved, which demands 30,000 time steps of $2.5e-5$ s. That computation takes around 30 hours on 4 Intel *i5* @ 3.4 GHz CPUs, 2 for *SPARC*, 1 for *FSiM* and 1 for *CalculiX*. Table 4.2 shows the fraction of the total computation time needed for *SPARC*, *CalculiX*, *FSiM* and for the communication between *FSiM* and the two solvers. The time needed for MPI communication within *SPARC* is already included in its time fraction. The time analysis shows that overhead due to the Fluid-Structure Interaction is non-negligible but still remains in a tolerable level. Larger models tend to deliver better time ratios since the computing time grows with the model volume whereas the interaction operations grow with the model surface.

Finally, a comment is made regarding the use of this test case to identify implementation errors. During the code development, an error was introduced, due to which, the velocity at the wall was computed two times greater. This had produced a period deviation of around 25%. Since, as explained at the beginning of this chapter, a higher accuracy levels could be expected from the comparison with a numerical test case,

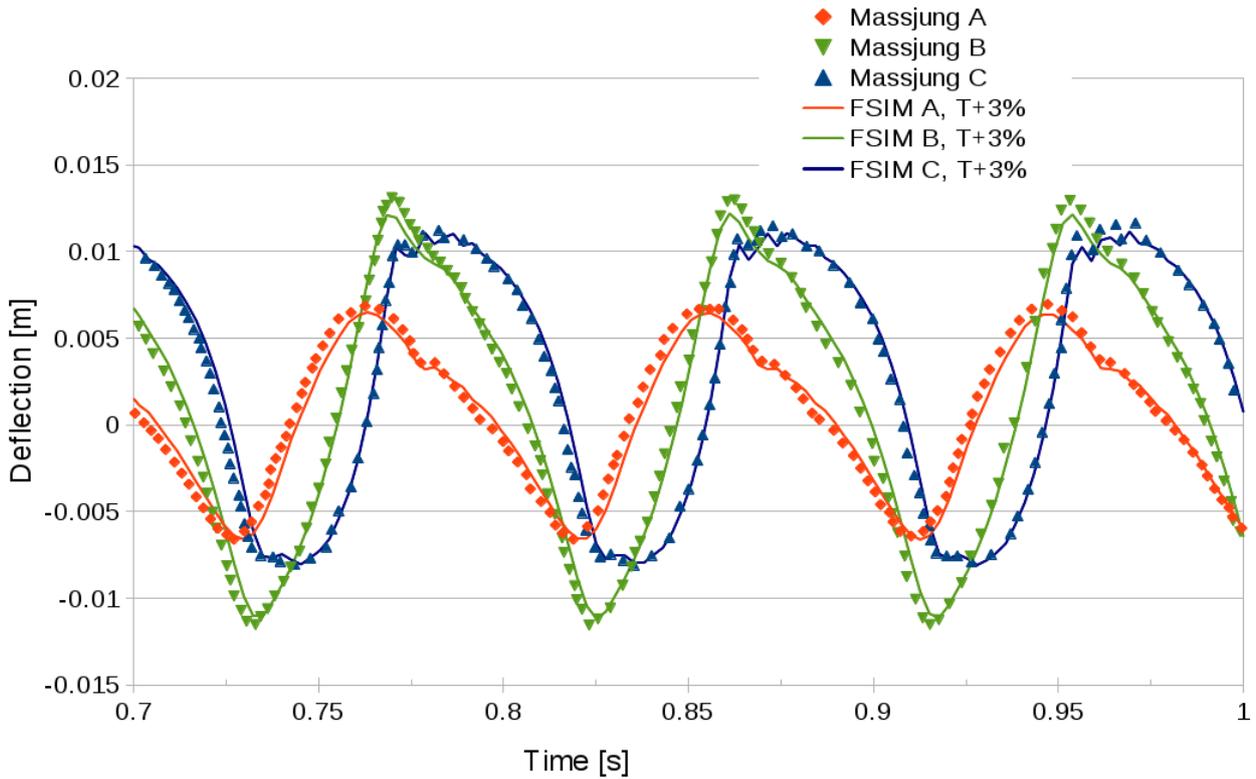


Figure 4.3: Comparison of results against the reference data.

that large deviation pointed out clearly a bug in the code.

4.3 Flexible obstacle

The second case is conceived to test *FSiM* in forced oscillations and to compare the performance with a commercial software package. The commercial solver *ADINA*[®] [77] has been used to generate the reference data. Unlike the previous case, viscous effects and a relatively low-Mach number are to be considered. The problem consists of a flexible obstacle in a half-open flow, whereby the tip deflection is examined. The corresponding geometry is presented in figure 4.5 whereas the properties of the beam and flow are summarized in Table 4.3. These properties have been chosen under following considerations:

1. A low-Mach number flow is desired to be able to compare against a common incompressible flow solver.
2. A laminar flow is convenient to exclude the influence of turbulence models in the comparison.
3. The deformation of the beam must be large enough to influence the flow but must be kept small enough to prevent excessive distortion of the mesh.
4. The density of the flow, ruling the acting forces, and the elasticity module, ruling the reacting forces, are to be chosen so as to fulfill the previous point.
5. The tapered form of the beam minimizes the contribution of higher order modes in the deflection.

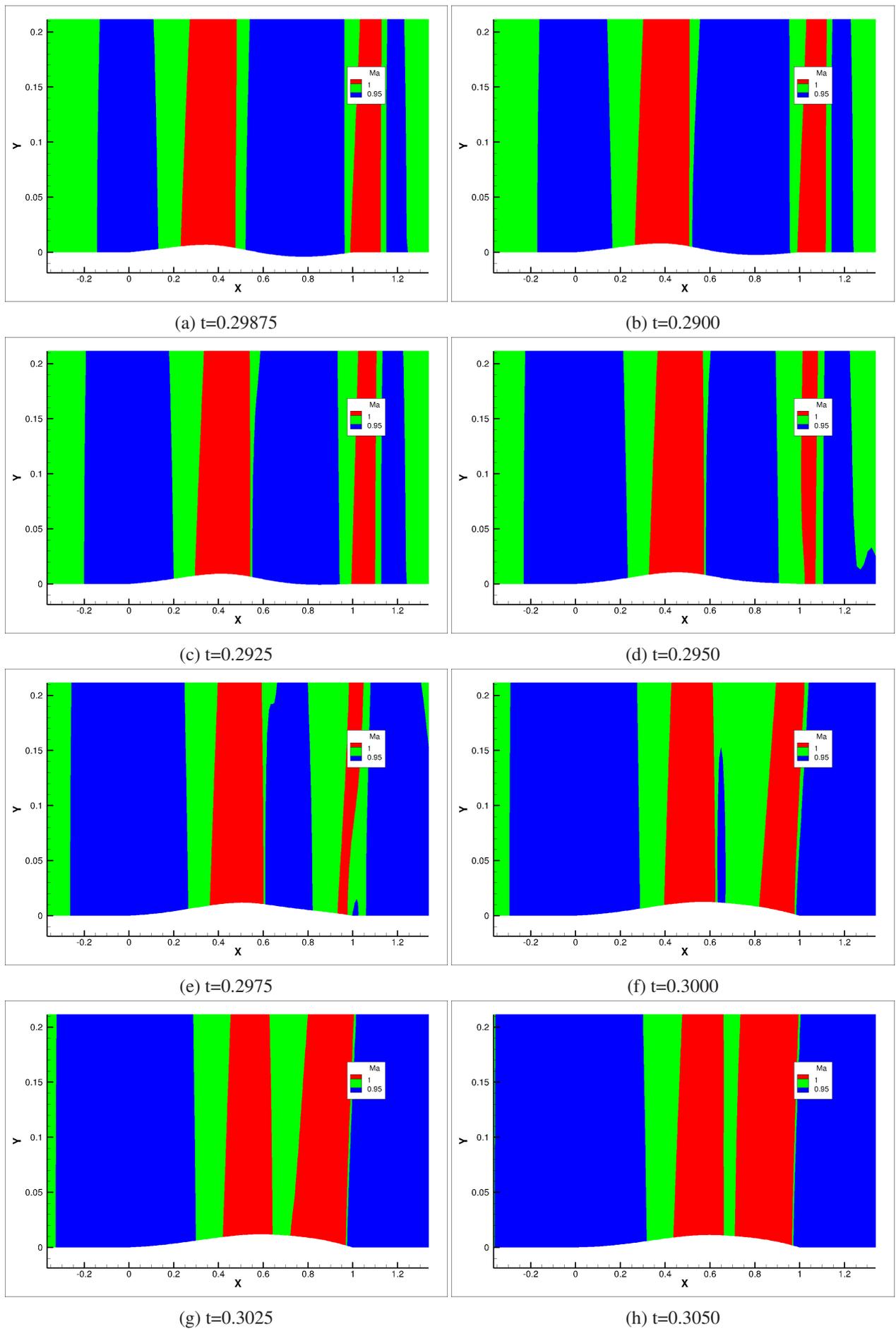


Figure 4.4: Mach number distribution, showing a traveling pressure shock.

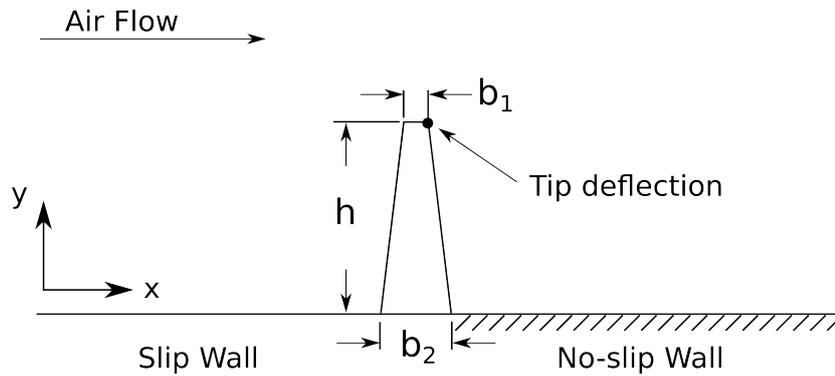


Figure 4.5: Geometry of the flexible obstacle test case.

Beam properties		Flow Properties	
h	5 mm	U	10 +/- 5m/s @ 120Hz
b ₁	0.5 mm	ρ _f	0.01 kg/m ³
b ₂	1 mm	μ	0.01 Pa.s
ρ _s	1000 kg/m ³	T ₀	292 K
E	1e5 Pa	P _{ref}	838.04 Pa
ν	0.3	Re _{h,max}	7.5
Natural frequency of 1 st mode	72.4 Hz		

Table 4.3: Properties of the beam obstacle.

- The slip-wall boundary condition before the obstacle allows to prescribe a block velocity profile. In this way it is easier to ensure that both solvers handle the inflow boundary condition in a similar manner.
- Due to the impossibility of introducing structural damping in *CalculiX*, a fictive, high viscosity fluid is required to reach a steady-state amplitude in a reasonable computing time.
- The frequency of the excitation must be chosen so as to achieve a moderate amplitude with the available damping, taking into account the characteristic of the amplification function, figure 2.3.

The two simulations were done using exactly the same geometry, mesh, boundary conditions and physical parameters. The only difference is that the incompressible formulation of *ADINA* was selected whereas *SPARC* uses a compressible formulation. Due to the large number of possible combinations, the selection of parameters was a very time consuming task. The boundaries are located 24 h above, 12 h before and 30 h after the obstacle, as can be appreciated in figure 4.6. Due to the high viscosity used, the domain should be chosen even larger, leading to a larger computing time. This is, however, not a problem for a numerical comparison, as long as the domain is equal for both solvers.

For the discretization of the obstacle, 80 8-node solid bilinear elements were used, whereas the fluid field was resolved with 20,000 finite-volume cells. Due to the strong diffusive fluxes a very small time step of 0.3125e-5 s was required.

A reference pressure is used for the calculation of the structural forces, which is equal to the static pressure at the outlet. To minimize reflections at the boundaries, the fluid domain is initialized with a steady-state solution of the pure CFD-model. The FSI computation begins with the beam at rest, and a pulsating block

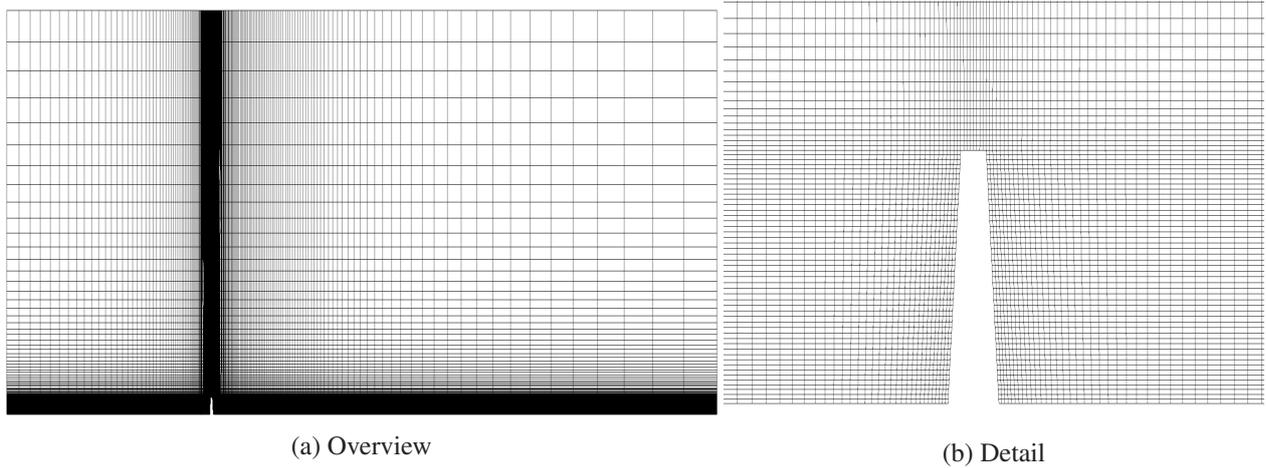


Figure 4.6: Simulation domain.

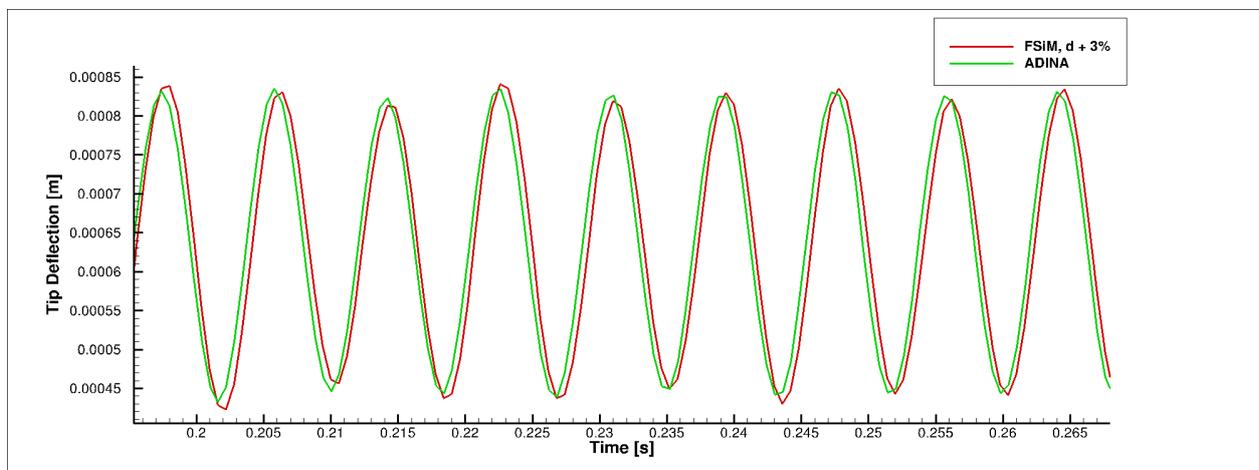


Figure 4.7: Comparison of tip deflection.

velocity profile is applied to the inlet. The beam then oscillates initially in the natural frequency passing through a transient phase and reaching limit-cycle-oscillations at the forced frequency after around 95,000 time steps. The computations with *FSiM* and *ADINA* show in general a very good agreement. Almost no deviation is observed in the amplitude, while the *FSiM* deflections have an offset of 3% below the *ADINA* deflections. Figure 4.7 shows the comparison with *ADINA*, whereby the deflection of *FSiM* is plotted with a 3% correction, in order to better compare the results. Between the two curves, a mean phase deviation of 17° is observed. This is attributed to slightly different overall resolution of the damping effects. Figure 4.8 shows the frequency spectrum of the tip deflection. The number of periods is not large enough for the Fourier transform to provide a good resolution of the peak. Still, the dominant frequency of 120 Hz can be clearly identified.

The deflection offset cannot be explained due to compressibility effects, since for the low Mach number (0.029) the variation of total pressure in comparison to the incompressible computation is negligible. The accuracy of the structural solver has been tested with similar cases and the deviation lies below 0.5%. Therefore the explanation must be searched in the fluid field. In order to compare the fluid fields of *SPARC* and *ADINA*, additional simulations were performed with a constant velocity of 10 m/s, so as to achieve a stationary solution. The velocity and pressure contours show a very good agreement, figures 4.9 to 4.12. For

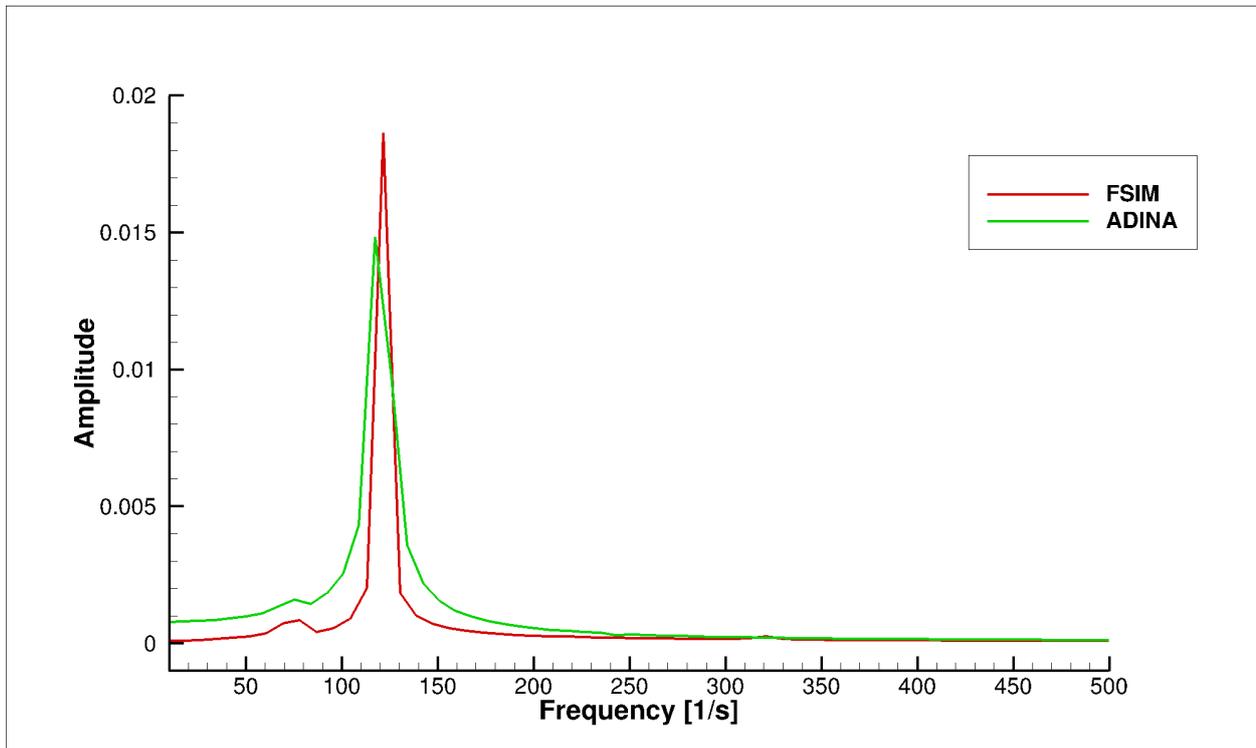


Figure 4.8: Frequency spectrum of tip deflection.

a more quantitative comparison, figures 4.13 and 4.14 show the horizontal velocity component and pressure profiles, extracted close before the beam along the line shown in figure 4.9. The velocity profile exhibits a maximal deviation of around 20% in the lowest area. Unlike the *ADINA* solution, where the density is constant, *SPARC*'s compressible formulation computes a density distribution. Despite the low mach number, the compressibility effects on the density are not negligible and the deviation from the constant value of $0.01\text{kg}/\text{m}^3$ amounts to 2% at the obstacle height. Since the velocity divergence is strongly related to the incompressibility, this deviation can partly explain the different velocity profiles. The remaining part can be attributed to numerical effects since the compressible formulation of *SPARC* is pushed to its limits. However, it is not the velocity deviation but rather the pressure that has a direct influence on the deflection. The maximal deviation in the pressure profile is around 6% in the outer region, where the influence on the deflection is less important. At the obstacle height the pressure of the *FSiM* solution is around 3% lower than that of *ADINA*. Since pressure is most directly related to the obstacle deflection, this deviation can adequately explain the deflection offset.

Finally, a time benchmark was done to show the overall performance in comparison with *ADINA*. Table 4.4 shows the time needed for both codes to compute 1000 time steps, having 8 cores available. In the case of *ADINA*, 8 threads were started, whereas for *SPARC*, 6 cores were assigned, leaving the other two for *FSiM* and *CalculiX*. It must be taken into account that the default working mode of most MPI implementations assumes no oversubscribed nodes. This means that the total number of processes is not higher than the number of available cores. Under this assumption, a waiting process still consumes 100 % of the core in order to detect the availability of the pending message as soon as possible. In some implementations it is possible to change the working mode so that a waiting process consumes very little core performance, thus allowing to assign 8 cores to *SPARC* in this example. However, since two extra cores do not improve the times of *SPARC* significantly, all runs have been done with the default working mode. The performance

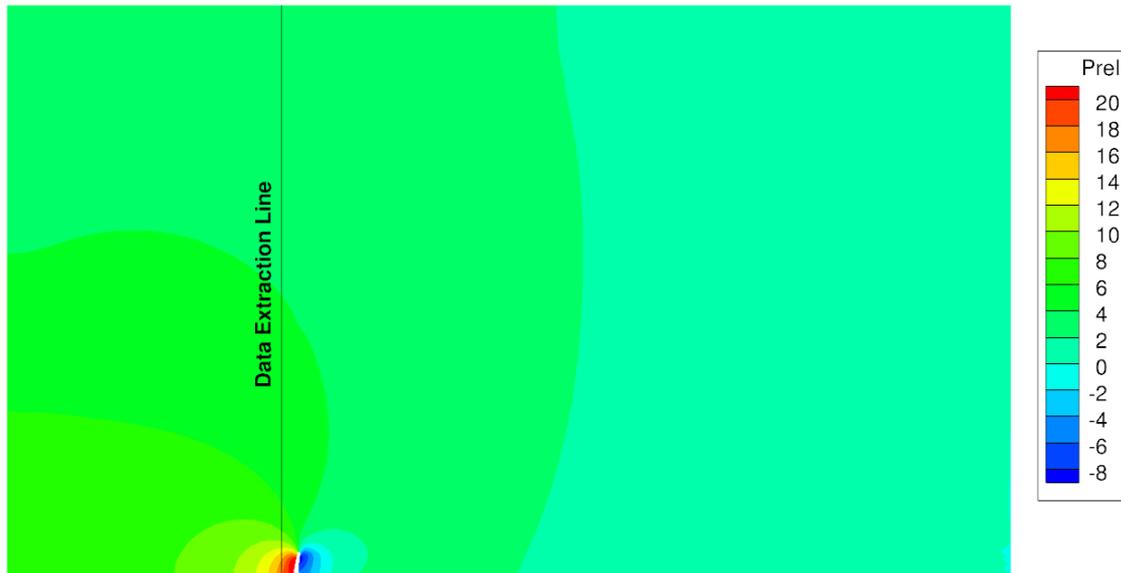


Figure 4.9: Pressure contour plot of FSiM.



Figure 4.10: Pressure contour plot of ADINA.

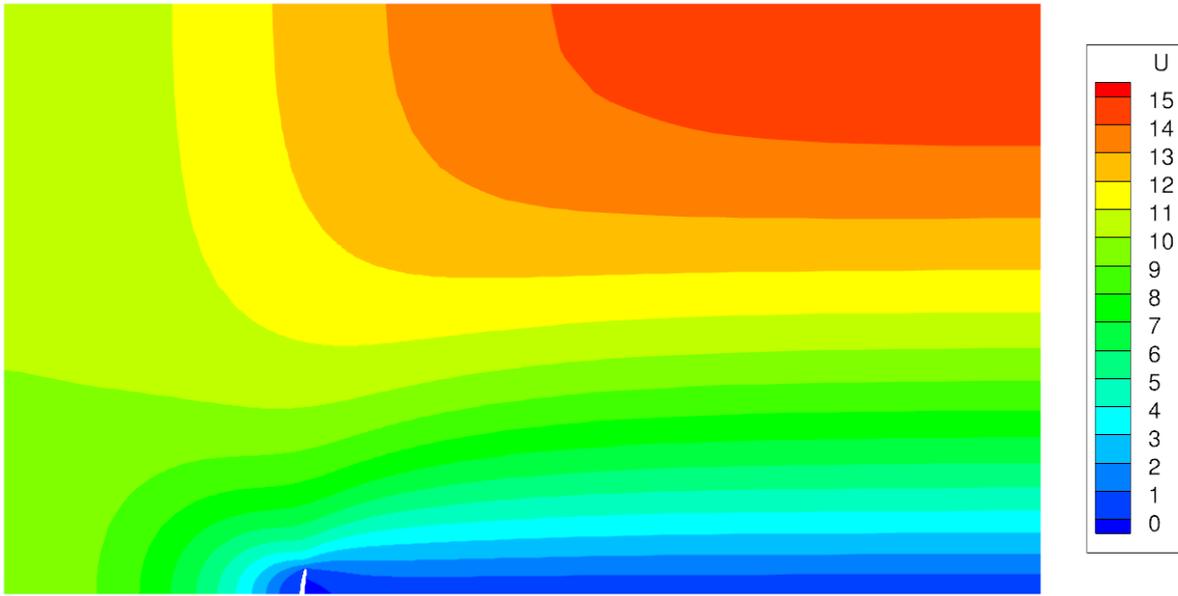


Figure 4.11: Velocity contour plot of FSiM.

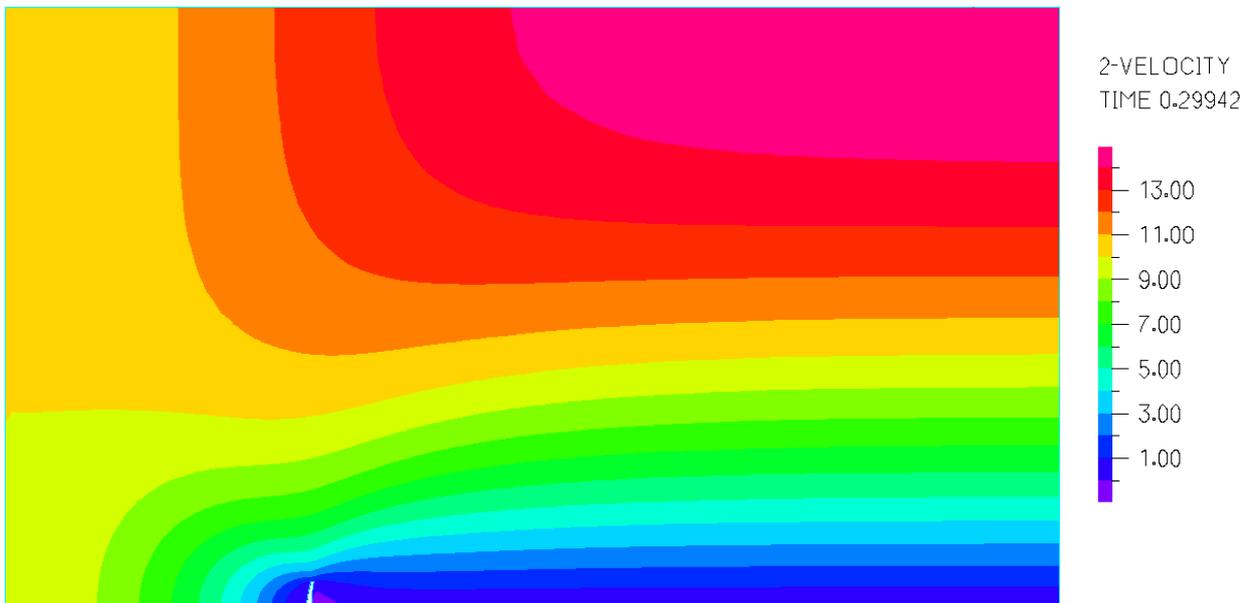


Figure 4.12: Velocity contour plot of ADINA.

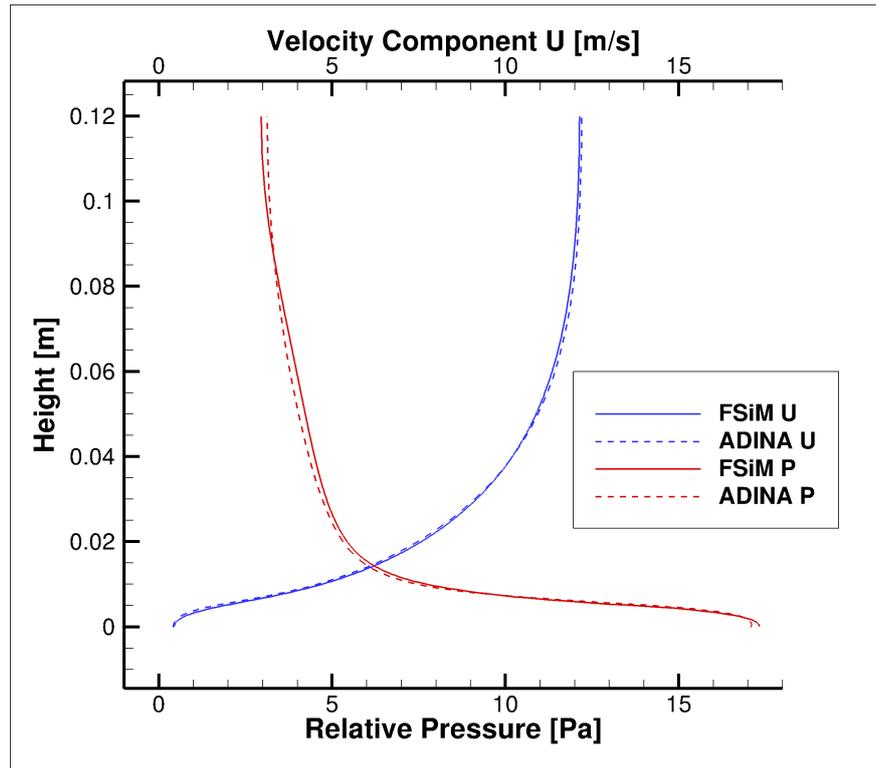


Figure 4.13: Comparison of pressure and velocity profiles between FSiM and ADINA.

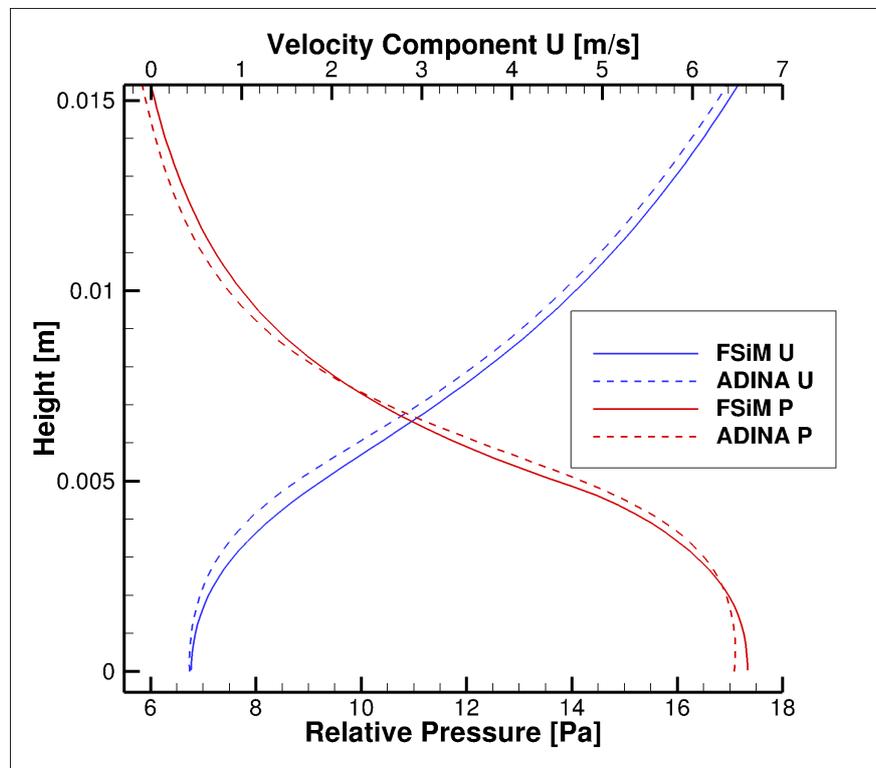


Figure 4.14: Comparison of pressure and velocity profiles between FSiM and ADINA. Detail.

of *ADINA* is slightly better (14 %) which can be expected since an incompressible formulation was used where the energy equation is not solved in comparison with *SPARC*'s compressible formulation. Still, the comparison points out the efficient implementation of the developed tool. Table 4.5 shows the time fractions needed by each component of the simulation. These are similar to those of the previous case, which can be anticipated since the model sizes are comparable.

Performance	
FSiM	ADINA
84 min/1000 it	74 min/1000 it

Table 4.4: Computational time performance.

	Comm. from and to FSiM	FSiM	SPARC	CalculiX
Percentage	6%	5%	80%	9%

Table 4.5: Computational time fraction for each coupling component.

5 Application to rotating machinery

As presented in the introduction, the final objective of the present development is the investigation of flow-induced vibrations in turbomachinery. To demonstrate the potential of the developed tool, an application example is now presented. This test case consists of a two-dimensional turbine inside a casing, resembling a flow-meter, as shown in figure 5.1. The relevant physical and geometrical parameters are detailed in table 5.1.

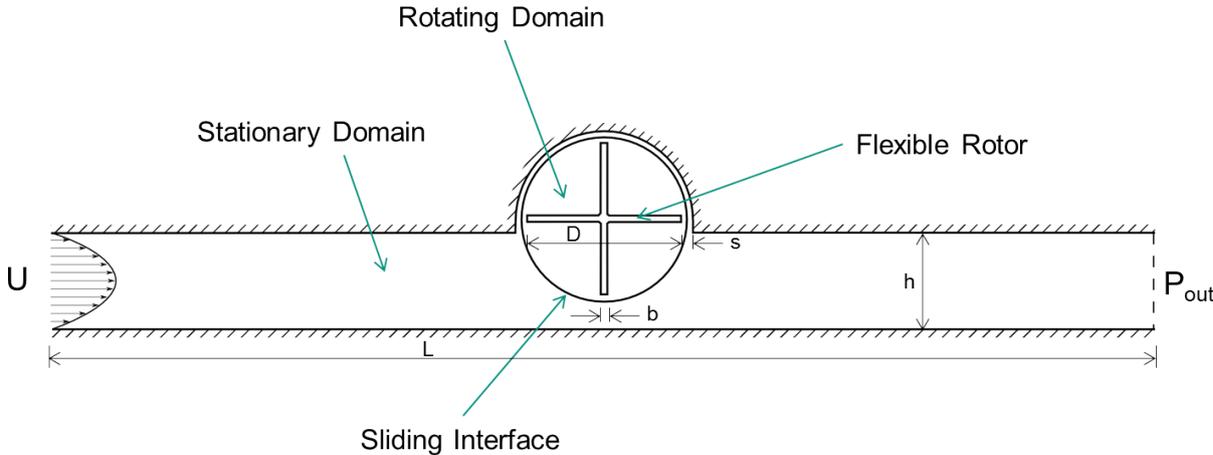


Figure 5.1: Geometry of the application example.

Rotor Properties		Flow Properties	
D	13 cm	U	30 m/s
b	0.5 cm	ρ_f	6 kg/m ³
ρ_s	1200 kg/m ³	μ	1.807e-5 Pa.s
E	3.2e9 Pa	T_0	292 K
ν	0.3	P_{ref}	502824 Pa
f	55 Hz	Casing Geometry	
λ	0.75	H	7.25 cm
		L	93 cm
		s	1 cm

Table 5.1: Properties of the application example.

It must be stressed that the purpose of this test case is to demonstrate what analysis could be relevant for the investigation of vibrations in turbomachinery, with a simplified but still intuitive model, rather than to reproduce an experiment. This is reflected in the choice and placement of boundary conditions.

The inflow boundary condition consists of a parabolic velocity profile while a constant static pressure equal to the reference pressure is prescribed at the outlet. To determine the rotor velocity, the experimental maximum for power extraction of a savonius rotor was used as a reference. This is given in form of the tip-speed ratio:

$$\lambda = \frac{U}{\pi f D} \quad (5.1)$$

On this basis, the rotor is set to rotate at a constant frequency f . A sliding interface is used to connect the rotating and stationary domains, which are discretized with a total of 140,000 finite-volume cells. The values of velocity, flow density and elasticity module have been chosen so as to achieve a moderate deformation of the rotor, since a too large deformation would degrade the mesh quality. The flow is in full-turbulent regime ($Re_H = 730,000$) and resolved with the Spalart-Allmaras turbulence model. Around 130,000 time steps of $2e-5$ s were needed to overcome the initial transient phase and reach a periodic solution.

The rotor is discretized with 904 8-node solid bilinear elements. In accordance with the rotating frame of reference approach, the only influence that the rotation has on the structural field is the addition of centrifugal forces. These contribute with a stiffening effect on the rotor blades. The rotor is supported on eight nodes in the shaft zone, as shown in figure 5.2, where all DOF are set to zero. This is equivalent to a rigid shaft. The shaft reaction moment results from the integration of the reaction forces in these nodes.

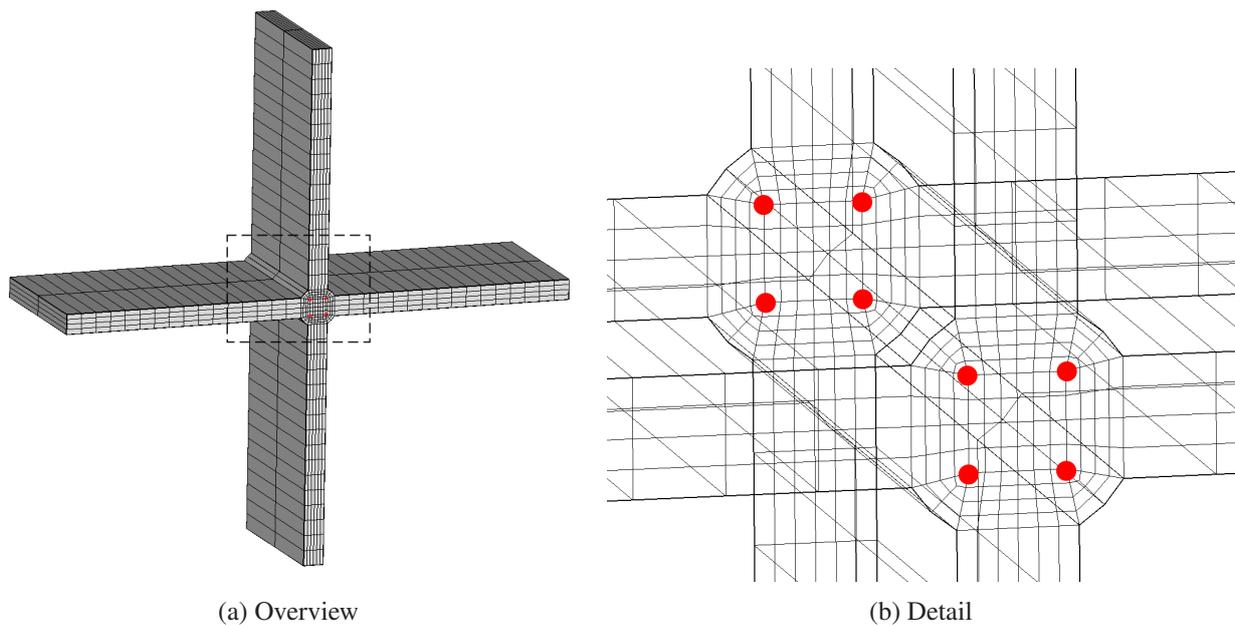


Figure 5.2: Fixed nodes to model the rotor shaft.

A first impression of the FSI computation is given in figure 5.4, from where the connection between the pressure field and the resulting force vectors acting on the structure can be inferred. Consistent with the rotating frame of reference approach, the structure is always depicted in the unrotated position, since the structural field is actually computed so. This representation is also more convenient to visualize the blade deformation. As a reference, the magenta mark points out the upper blade on the structure plot.

Three investigations are deemed relevant for industrial purposes:

- Influence of rotor flexibility on fluid moment
- Influence of rotor flexibility on shaft reaction moment
- Time history of excitation and displacement for a blade

For the first and second investigations, an additional CFD computation with the same parameters was done to represent the case of a stiff rotor. Figure 5.5 shows the fluid moment for the rigid and flexible rotor that results from the integration of the forces acting on all four blades. As shown in Figure 5.6, the dominating frequency is 220 Hz, the four-fold of the rotation frequency, as can be obviously anticipated from the number of blades. A slight reduction of the fluid moment can be appreciated due to the flexibility of the rotor.

Rotor natural frequencies	
1 st Mode	136 Hz
2 nd Mode	342 Hz
3 rd Mode	346 Hz
4 th Mode	347 Hz
5 th Mode	1465 Hz

Table 5.2: Natural frequencies of the rotor.

The second aspect requires more insight into the rotor dynamics. For that purpose, a modal analysis was performed with *CalculiX*. The frequencies and forms of the first five eigenmodes are presented in table 5.2 and figure 5.9 respectively. From direct observation, it can be anticipated that the asymmetric modes 1 and 5 might have an influence in the shaft reaction torque, whereas the symmetric modes 2,3 and 4 are balanced with respect to the rotation axis. The fluid moment load is nearly harmonic with a frequency of 220 Hz but a complex spatial distribution. A modal decomposition of this distribution leads to an excitation of all modes, including mode 1. The excitation frequency is 1.65 times the natural frequency of the first mode and therefore it can be anticipated, from figure 2.3, that the first mode will be triggered with an amplification factor lower than one. Since the reaction moment is directly related to the amplitude of motion, equation 3.13, a reduction of the shaft reaction is expected with respect to the stiff rotor. Analogously, a phase difference of around π is expected. These deductions can be confirmed in figure 5.7, where the reaction moments of the two simulations are compared. The reaction moment for the rigid rotor was obtained from the CFD computation by means of equation 3.14.

The third issue is addressed in figure 5.8, where the fluid moment of a single blade is plotted. The amplitude of the blade fluid moment is greater than that for the whole rotor, because the single blade moments partially cancel each other and are therefore not fully transmitted to the shaft. The blade fluid moment shows a clear peak with the rotation frequency of 55 Hz, but contains other components of higher frequency as well, as can be observed in a spectrum plot, figure 5.3. These loads trigger a complex tip displacement outlined in the same figure. The blade tip oscillates with a dominating frequency of 330 Hz which can, of course, also be found in the spectrum plot. The frequency of 330 Hz is the one in the spectrum that lies closest to the natural frequencies of modes 2, 3 and 4, and therefore its contribution is amplified much more than that of the other frequencies. The amplitude of the blade tip deflection oscillates with low frequencies, among others, the rotational frequency of 55 Hz, which can be clearly recognized.

All of these information show the consistency of the results delivered by the developed tool and can be used, for example, as the starting point for a fatigue analysis of the blade.

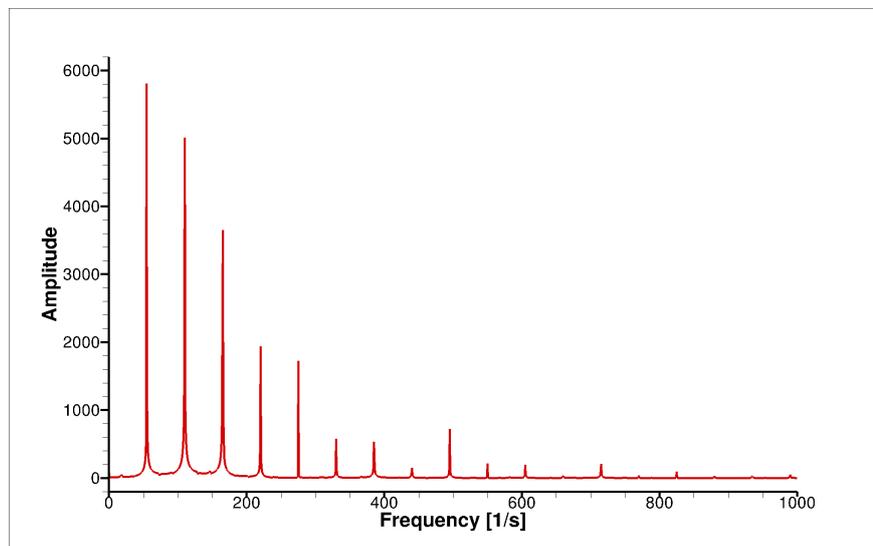
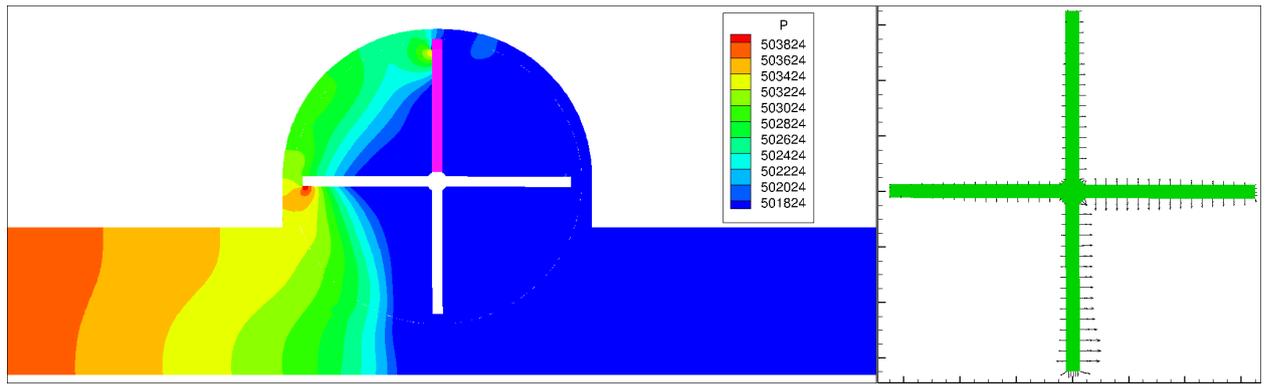
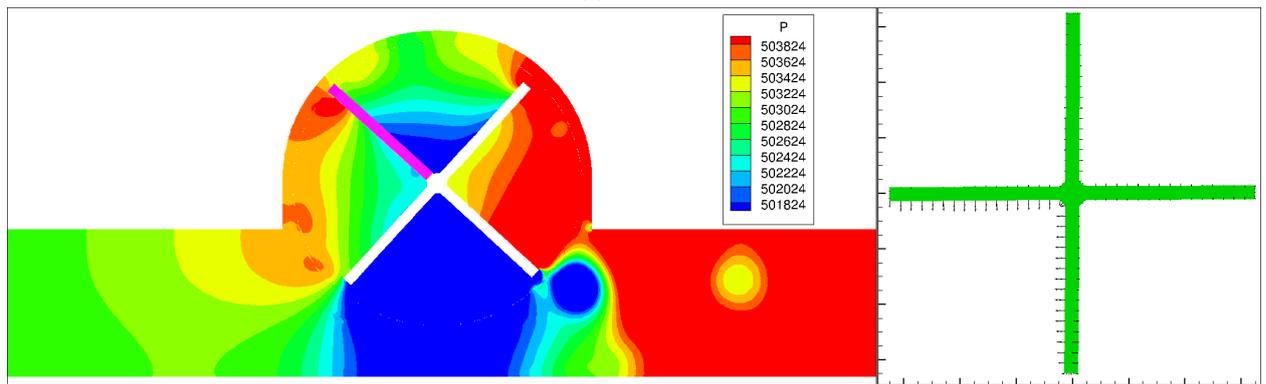


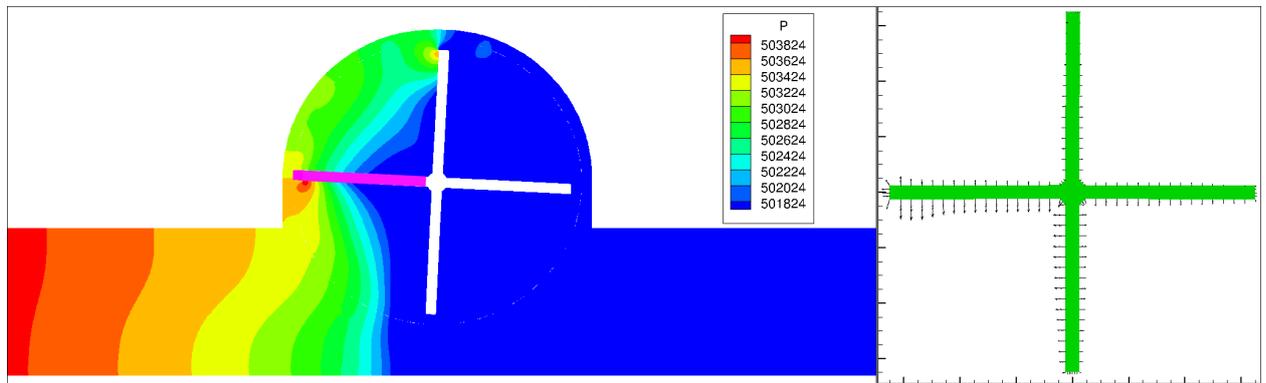
Figure 5.3: Frequency spectrum of the blade moment.



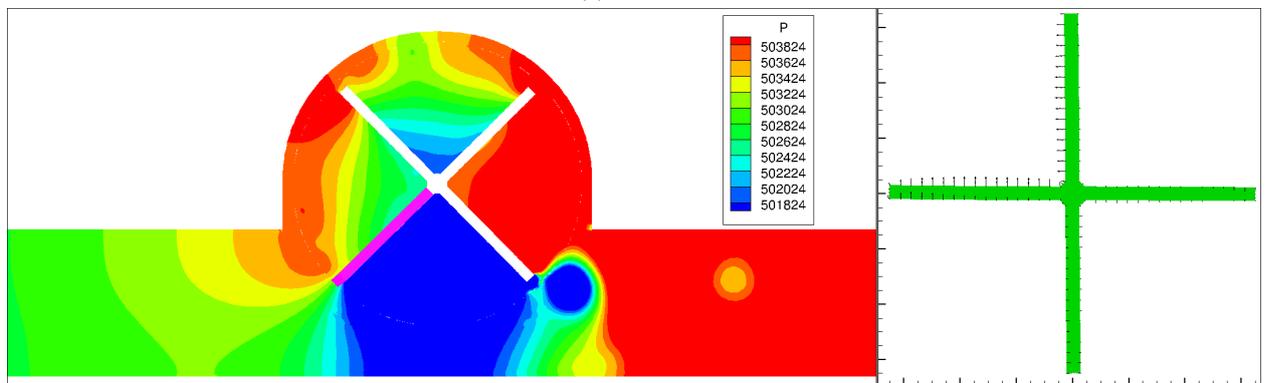
(a) 2.400 s



(b) 2.402 s



(c) 2.404 s



(d) 2.407 s

Figure 5.4: Computation results at different times.

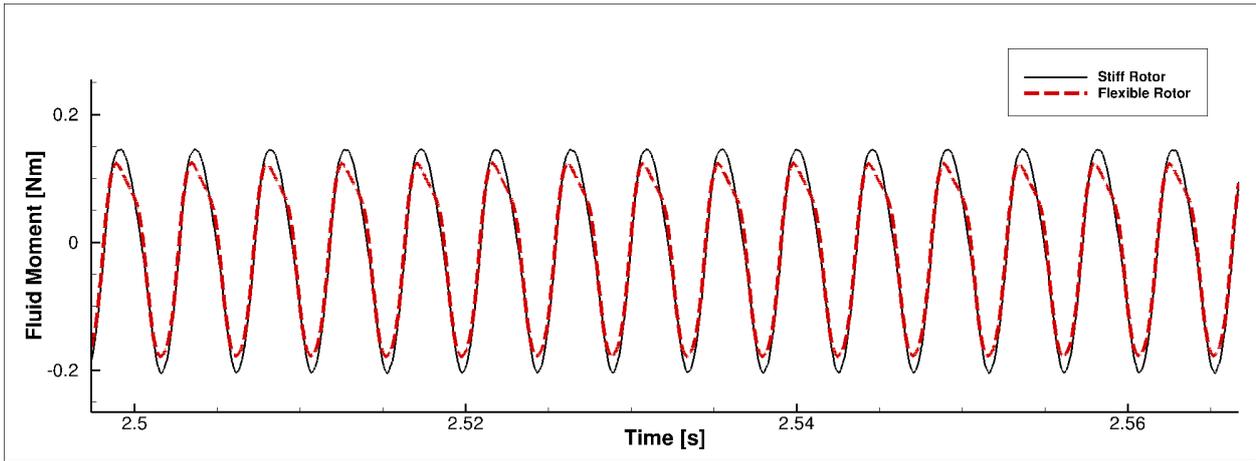


Figure 5.5: Influence of rotor flexibility on fluid moment.

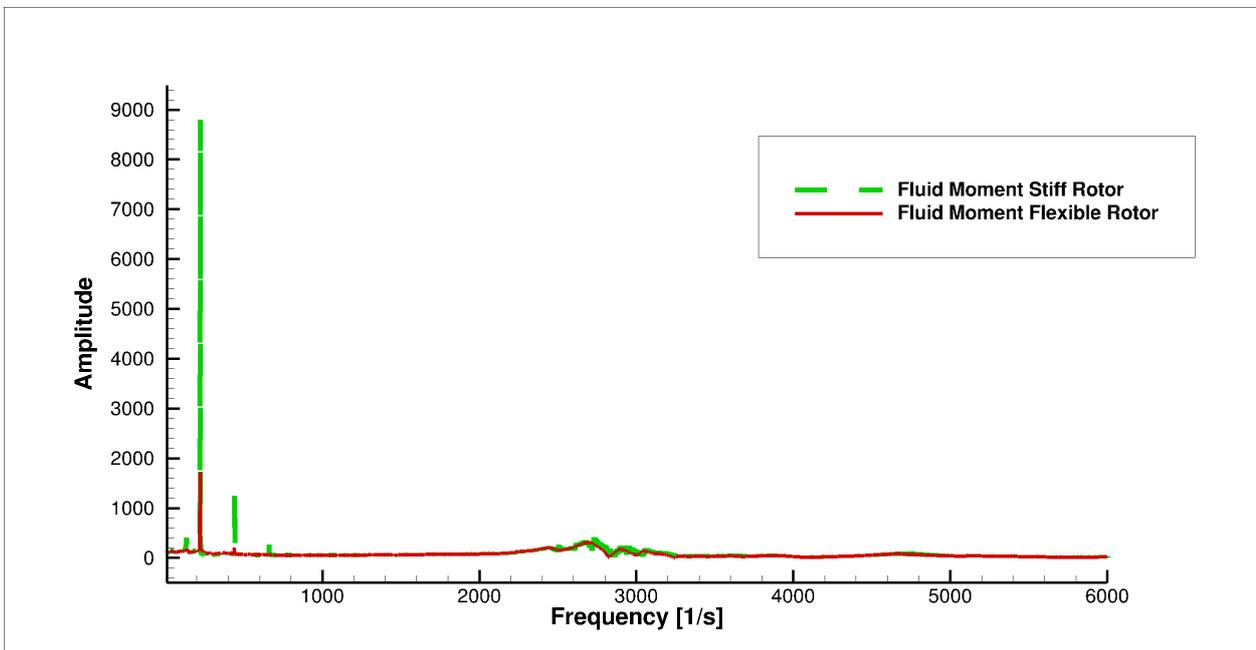


Figure 5.6: Frequency spectrum of fluid moment.

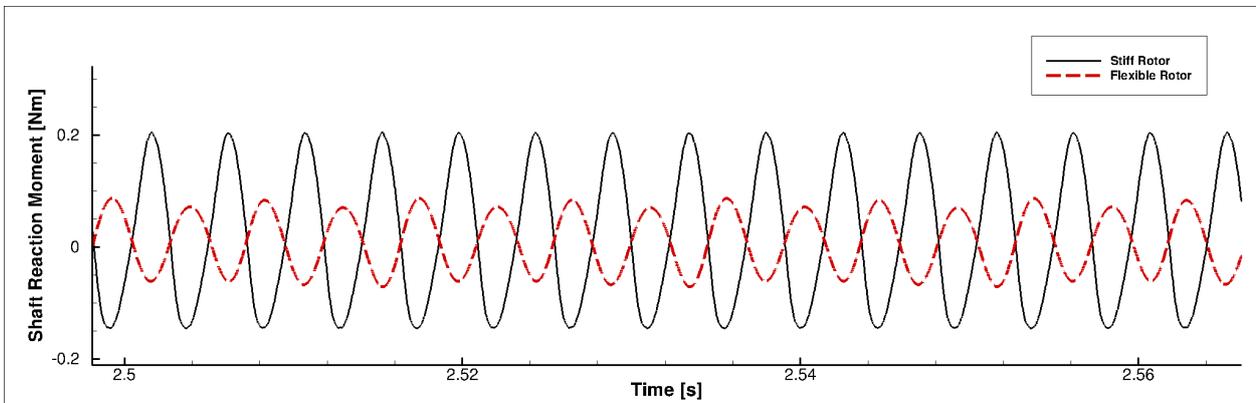


Figure 5.7: Influence of rotor flexibility on shaft reaction moment.

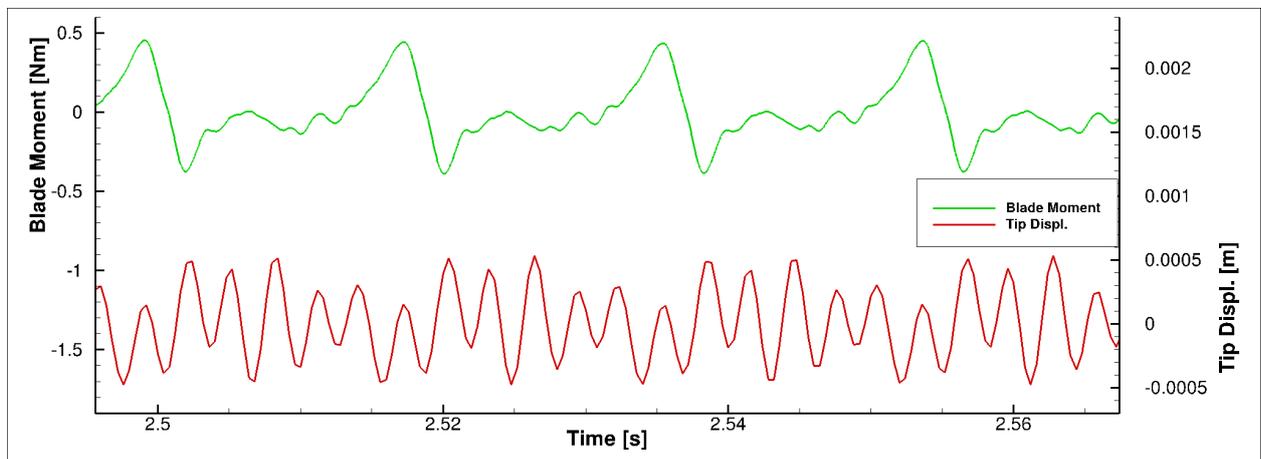
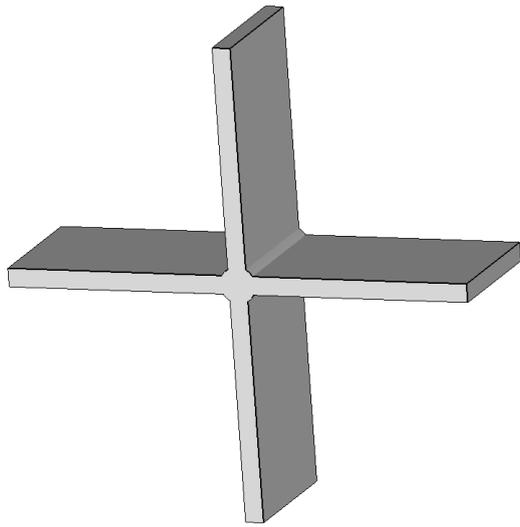
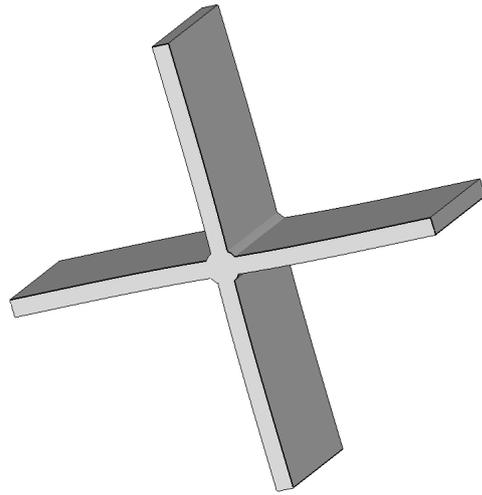


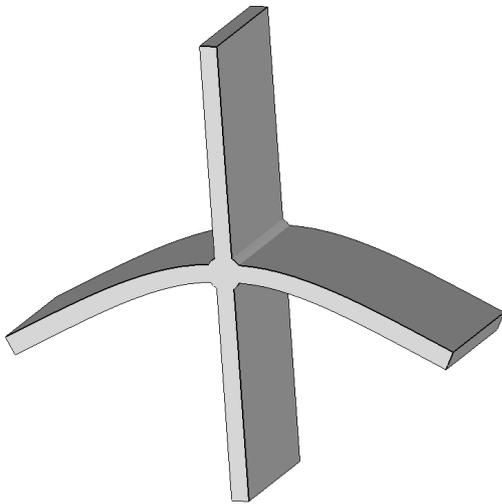
Figure 5.8: Time history of excitation and displacement for a blade.



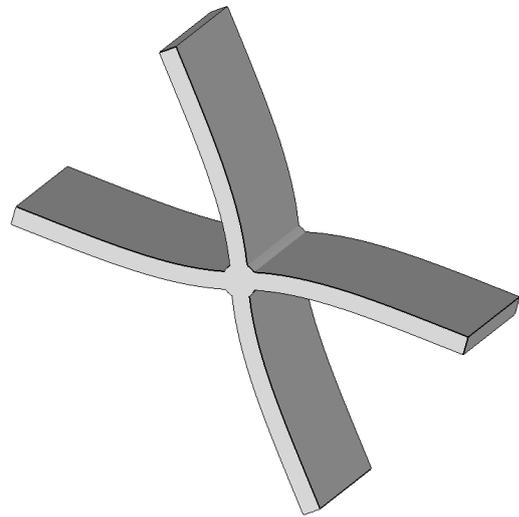
(a) Undeformed



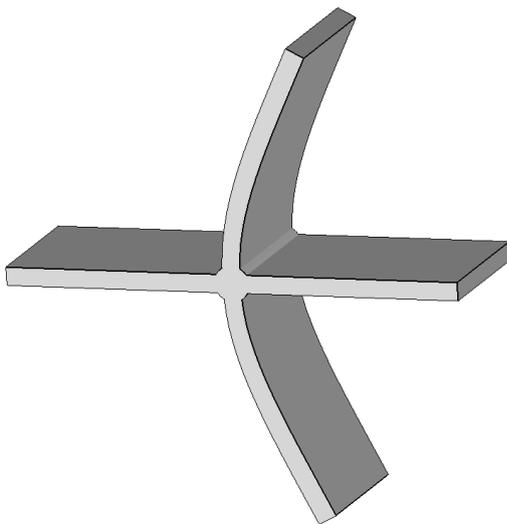
(b) 1st mode



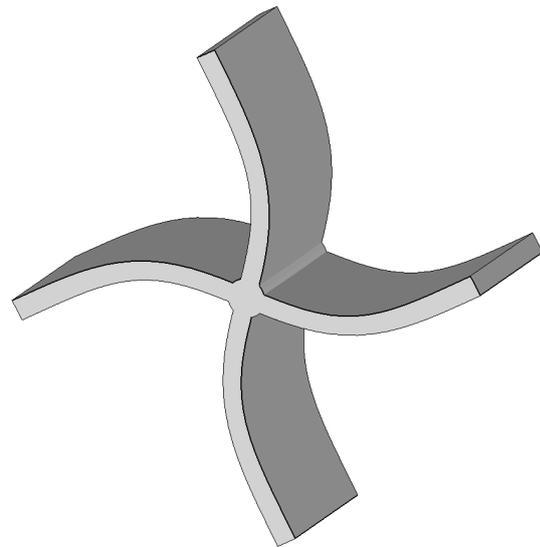
(c) 2nd mode



(d) 3rd mode



(e) 4th mode



(f) 5th mode

Figure 5.9: Vibration modes of the rotor.

6 Conclusions and outlook

6.1 Conclusions

Throughout this thesis, the development of a computational tool for the simulation of Fluid-Structure Interaction in rotating machinery has been presented. Three-dimensional, high resolution coupled solvers are required to address challenging design issues in different turbomachinery types. The FSM's CFD solver *SPARC* matches those requirements with some advantages in comparison with commercial tools. To exploit these advantages for coupled simulations, the open-source software package *CalculiX* was chosen to solve the structural dynamics. The use of existing solvers exhibits some limitations as regards the error handling and the text output, due to the fact that the source code of these existing solvers had not been conceived for coupled use.

To deal with the communication and synchronization of the two part-solvers, the "Fluid Structure Interaction Simulation Manager" *FSiM* was created, based on the Dynamic Process Management of MPI. The complete resulting coupled tool (*SPARC* + *CalculiX* + *FSiM*) has been designed to take advantage of the benefits of block-structured meshes. The multigrid strategy, particularly efficient in block-structured meshes, can be exploited to overcome meaningless transients with a faster, coarser mesh. The use of the algebraic Transfinite Mapping method, only applicable to block-structured meshes, allows for a fast mesh adaption in comparison with the pseudo-elastic approaches, usually employed in unstructured meshes. The block structure provides a straightforward basis to control the mesh adaption, in order to avoid excessive mesh distortion. In comparison with general-purpose coupling tools, *FSiM* incorporates the data structures of both part solvers to avoid costly additional mappings between volume and surface meshes in every coupling iteration. Particularly for block-structured meshes, the mapping from volume to surface mesh is very efficient.

To handle non-matching meshes, appropriate algorithms have been developed to transfer the displacements and loads across the interface. To this purpose, the interaction surfaces can be segmented to account for geometrical discontinuities. The developed tool is quite general and can be easily extended to other surface-coupled problems, such as heat transfer.

Flow-induced vibrations appear with two dominating mechanism: aeroelastic instabilities and forced oscillations. Two suitable test cases addressing each of them were carefully selected in order to verify the implementation. The first mechanism was verified with results from the literature for the panel flutter problem, while for the second, the commercial software package *ADINA* was used to generate reference data. In both cases, a very good agreement of around 3% was achieved. This kind of numerical verification allows a more accurate comparison against reference data than experimental validations. For FSI in particular, the design of an experimental setup for validation purposes is still challenging. No suitable validation test was found in the literature that met the additional requirement of a moderate Mach number for *SPARC*'s compressible formulation.

Finally, to address rotating systems, the compatibility of displacements and loads must be ensured taking into account the different ways there exist to handle rotational effects. The alternatives of rotating frame of

reference and rotating mesh can be combined for the fluid and structural solver, to obtain four possibilities that are to be handled differently. In this work, we concentrated on the combination of rotating mesh treatment for the flow solution, and the rotating frame of reference approach for the structural solution. The different configurations are made compatible with rotational transformations for displacements and forces. These transformations must be taken into account for the fluid mesh adaptation. To show the capabilities of the new development for the investigations of flow-induced vibrations in turbomachinery, the example of a two-dimensional turbine has been discussed.

6.2 Outlook

A series of possible extensions have been identified throughout the work, that would improve the computational performance and accuracy. These have not been implemented due to time restrictions. They include:

Mesh update Currently, the algebraic Transfinite Mapping method is used. An elliptic method or the Delauney mapping could provide better mesh quality. To obtain more flexibility, a different mesh update procedure could be defined for each block.

ALE formulation The algorithm presented by Geuzaine et al. [57] promises a second-order accuracy in time in moving meshes, and should be considered for high efficient computations.

Coupling scheme For aeroelastic applications, the Generalized Sequential Staggered (GSS) algorithm promises a faster convergence.

Interface treatment Interpolation methods based on Radial Basis Functions [136], [112] have shown an improvement in the flexibility and accuracy of the interface treatment.

Free rotations The computation of models with free rotations would allow for more insight into transient phenomena. *FSiM* would prove very advantageous for that since, for example, external forces coming from the generator or motor can be easily introduced in the simulation.

Multiple blade domains An interesting possibility for turbomachinery can be achieved if a full rotor simulation is desired. In some cases the hub can be modeled as rigid, allowing an easy domain decomposition whereby each blade/vane is solved in parallel with independent *CalculiX* processes. Some additional implementations are then needed to handle several independent structural models simultaneously.

CalculiX spawning scheme Due to some limitations in many MPI implementations it is not convenient to spawn a program too many times. Modifying the spawning scheme of *CalculiX*, to have it run only once or twice throughout the computation, would slightly improve the efficiency and allow for a more elegant and stable implementation.

Bibliography

- [1] Dewey H. Hodges and G. Alvin Pierce. *Introduction to structural dynamics and aeroelasticity*. New York : Cambridge University Press, 2011.
- [2] Helmut Jäger, Roland Mastel, and Manfred Knaebel. *Technische Schwingungslehre*. Springer Vieweg, 2013.
- [3] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier, 2001.
- [4] Gergerly Bárdossy. Weiterentwicklung des in SPARC integrierten FSI-Moduls. Technical report, Fachgebiet Strömungsmaschinen, KIT, 2008.
- [5] Yuwei Li, Kwang-Jun Paik, Tao Xinga, and Pablo M. Carrica. Dynamic overset CFD simulations of wind turbine aerodynamics. *Renewable Energy*, 37:285–298, 2012.
- [6] Markus May, Yann Mauffrey, and Frédéric Sicot. Numerical flutter analysis of turbomachinery bladings based on time-linearized, time-spectral and time-accurate simulations. *IFASD 2011 - 15th International Forum on Aeroelasticity and Structural Dynamics*, Paris, 27 - 30 June 2011.
- [7] Felix Lippold. *Zur Simulation von Fluid-Struktur-Wechselwirkungen mit flexiblen Kopplungsverfahren*. PhD thesis, Institut für Strömungsmechanik und Hydraulische Strömungsmaschinen. Universität Stuttgart, 2009.
- [8] B. Hübner, U. Seidel, and S. Roth. Application of fluid-structure coupling to predict the dynamic behavior of turbine components. *25th IAHR Symposium on Hydraulic Machinery and Systems*, 2010.
- [9] Gunjit Bir and Jason Jonkman. Aeroelastic Instabilities of Large Offshore and Onshore Wind Turbines. *Journal of Physics*, Conference Series 75, 2007.
- [10] Paul Veers, Gunjit Bir, and Donald Lobitz. Aeroelastic Tailoring in Wind-Turbine Blade Applications. *Windpower '98*, 1998.
- [11] Y. Bazilevs, M.-C. Hsu, I. Akkerman, S. Wright, K. Takizawa, B. Henicke, T. Spielman, and T. E. Tezduyar. 3D simulation of wind turbine rotors at full scale. Part I: Geometry modeling and aerodynamics. *Int. J. Numer. Meth. Fluids*, 65:207–235, 2011.
- [12] M. O. L. Hansen, J. N. Sørensen, S. Voutsinas, N. Sørensen, and H.A. Madsen. State of the art in wind turbine aerodynamics and aeroelasticity. *Progress in Aerospace Sciences*, 42:285–330, 2006.

- [13] Don W. Lobitz. Aeroelastic Stability Predictions for a MW-sized Blade. *Wind Energy*, 7:211–224, 2004.
- [14] C. A. Baxevanou, P. K. Chaviaropoulos, S. G. Voutsinas, and N. S. Vlachos. Evaluation study of a Navier - Stokes CFD aeroelastic model of wind turbine airfoils in classical flutter. *Journal of Wind Engineering and Industrial Aerodynamics*, 96:1425–1443, 2008.
- [15] P. K. Chaviaropoulos, N. N. Sørensen, M. O. L. Hansen, I. G. Nikolaou, K. A. Aggelis, J. Johansen, Mac Gauna, T. Hambraus, Heiko Frhr. von Geyr, Ch. Hirsch, Kang Shun, S. G. Voutsinas, G. Tzabiras, Y. Perivolaris, and S. Z. Dyrmoose. Viscous and Aeroelastic Effects on Wind Turbine Blades. The VISCEL Project. Part II: Aeroelastic Stability Investigations. *Wind Energy*, 6:387–403, 2003.
- [16] S. Streiner, E. Krämer, A. Eulitz, and P. Armbruster. Aeroelastic analysis of wind turbines applying 3D CFD computational results. *Journal of Physics*, Conference Series 75, 2007.
- [17] F. Carta. Coupled Blade-Disk-Shroud Flutter Instabilities in Turbojet Engine. *Journal of Engineering for Power*, pages 419–426, 1967.
- [18] Joury Temis and Ilya Fedorov. Simulation of Turbomachine Blade Bending-Torsion Flutter using a Pretwisted Beam Finite Element. *ENOC 2008*, June, 30-July, 4 2008.
- [19] T. J. Larsen, A. M. Hansen, and T. Buhl. *Aeroelastic effects of large blade deflections for wind turbines*, pages 238–246. Delft University of Technology, 2004.
- [20] A. Ahlstrøm. *Aeroelastic simulation of wind turbine dynamics*. PhD thesis, KTH, Sweden, 2005.
- [21] V. A. Riziotis, S. G. Voutsinas, E. S. Politis, and P. K. Chaviaropoulos. Aeroelastic Stability of Wind Turbines: the Problem, the Methods and the Issues. *Wind Energy*, 7:373–392, 2004.
- [22] E. S. Politis, I. G. Nikolaou, P. K. Chaviaropoulos, F. Bertagnolio, N. N. Sørensen, and J. Johansen. KNOW-BLADE Task-4 report; Navier-Stokes Aeroelasticity. Technical report, Risø National Laboratory, 2005.
- [23] J.F. Manwell, J.G.McGowan, and A.L. Rogers. *Wind Energy Explained. Second Edition*. Wiley, 2009.
- [24] Hirofumi Doi. *Fluid/Structure Coupled Aeroelastic Computations for Transonic Flows in Turbomachinery*. PhD thesis, Stanford University, August 2002.
- [25] F. Magagnato. Sparc-user manual.
- [26] F. Magagnato. Kappa - karlsruhe parallel program for aerodynamics. in: Task quarterly 2 (1998), s. 215-270.
- [27] T. Gallinger, A. Kupzok, U. Israel, K.-U. Bletzinger, and R. Wüchner. A computational environment for Membrane-Wind Interaction. *Fluid-Structure Interaction. Theory, Numerics and Applications.*, Herrsching am Ammersee. 29.9-1.10, 2008.

- [28] H.-J. Bungartz, J. Benk, B. Gatzhammer, M. Mehl, and T. Neckel. Partitioned Simulation of Fluid-Structure Interaction on Cartesian Grids. In *Fluid Structure Interaction II*. Springer Berlin Heidelberg, 2010.
- [29] M. H. Hansen. Improved Modal Dynamics of Wind Turbines to Avoid Stall-induced Vibrations. *Wind Energy*, 6:179–195, 2003.
- [30] S. Sarkar and H. Bijl. Nonlinear aeroelastic behavior of an oscillating airfoil during stall-induced vibration. *Journal of Fluids and Structures*, 24:757–777, 2007.
- [31] Derek E. Shipley, Mark S. Miller, and Michael C. Robinson. Dynamic Stall Occurrence on a Horizontal Axis Wind Turbine Blade. Technical report, National Renewable Energy Laboratory, 1995.
- [32] Jesper Winther Larsen. *Nonlinear Dynamics of Wind Turbine Wings*. PhD thesis, Department of Civil Engineering, Aalborg University, 2005.
- [33] Robert D. Cook, David S. Malkus, Michael E., and Plesha Robert J. Witt. *Concepts and Applications of Finite Element Analysis*. Wiley, 2002.
- [34] E. N. Dvorkin and M. B. Goldschmit. *Nonlinear Continua*. Springer, 2006.
- [35] R. Löhner, J.D. Baum, E.L. Mestreau, D. Sharov, Ch. Charman, and D. Pelessone. Adaptive Embedded Unstructured Grid Methods. *AIAA-03-1116*, 2003.
- [36] B. Jastrow and F. Magagnato. An Immersed Boundary Method for Simulating Compressible Viscous Flow in Complex Geometries. *9th International Conference on Advances in Fluid Mechanics, Split, Croatia, 26.06.-28.06.2012*, 2012.
- [37] J.T. Batina. Unsteady Euler airfoil solutions using unstructured dynamic meshes. *AIAA Paper no. 89-0115, AIAA 27th Aerospace Sciences Meeting & Exhibit*, 9-12 January 1989, Reno, NV.
- [38] O. A. Kandil and H. A. Chuang. Unsteady vortex-dominated flows around maneuvering wings over a wide range of Mach numbers. *AIAA 26th Aerospace Sciences Meeting, Reno, Nevada, 11-14 January 1988*, AIAA Paper No. 88-0317.
- [39] C. Farhat and T. Y. Lin. Transient aeroelastic computations using multiple moving frames of reference. *AIAA Eighth Applied Aerodynamics Conference, Portland, Oregon, 20-22 August 1990*, AIAA Paper No. 90-3053.
- [40] Tayfun Tezduyar, Sunil Sathe, Keith Stein, and Luca Aureli. Modeling of Fluid-Structure Interactions with the Space-Time Techniques. 53:50–81, 2006. 10.1007/3-540-34596-5/3.
- [41] J. Donea. An arbitrary Lagrangian-Eulerian finite element method for transient fluid-structure interactions. *Comput Meths Appl Mech Engng*, 33:689–723, 1982.
- [42] M. Lesoinne and C. Farhat. Geometric conservation laws for flow problems with moving boundaries and deformable meshes, and their impact on aeroelastic computations. *Computer Methods in Applied Mechanics and Engineering*, 134:71–90, 1996.

- [43] Christiane Förster. *Robust methods for fluid-structure interaction with stabilized finite elements*. PhD thesis, Institut für Baustatik und Baudynamik der Universität Stuttgart, 2007.
- [44] R. van Loon, P. D. Anderson, F. N. van de Vosse, and S. J. Sherwin. Comparison of various fluid-structure interaction methods for deformable bodies. *Computers and Structures*, 85:833–843, 2007.
- [45] A. van Dam and P. A. Zegeling. Mesh speed versus solution interpolation - making moving mesh results sharper. *Proceedings of the ICFD 2007*.
- [46] Sunil Sathe and Tayfun E. Tezduyar. Modeling of fluid-structure interactions with the space-time finite elements: contact problems. *Computational Mechanics*, 43:51–60, 2008.
- [47] Erwin Stein, René de Borst, and Thomas J. R. Hughes, editors. *Encyclopedia of Computational Mechanics, Volume 1: Fundamentals*. John Wiley & Sons, Ltd., 2004. ISBN: 0-470-84699-2.
- [48] J. G. Thompson, Z. U. A. Warsi, and C. Wayne Mastin. *Numerical Grid Generation*. 1985.
- [49] Malte von Scheven. *Effiziente Algorithmen für die Fluid-Struktur-Wechselwirkung*. PhD thesis, Institut für Baustatik und Baudynamik der Universität Stuttgart, 2009.
- [50] J. F. Thompson, B. K. Soni, and N. P. Weatherill. *Handbook of Grid Generation*. 1999.
- [51] M. Glück. *Ein Beitrag zur numerischen Simulation von Fluid-Struktur Interaktionen - Grundlagenuntersuchungen und Anwendung auf Membrantragwerke*. Dissertation, Technische Fakultät der Universität Erlangen-Nürnberg, 2002.
- [52] P. D. Thomas and C. K. Lombard. Geometric conservation law and its application to flow computations on moving grids. *AIAA Journal*, 17:1030–1037, 1979.
- [53] J. G. Trulio and K. R. Trigger. Numerical solutions of the one-dimensional hydrodynamic equations in an arbitrary time-dependent coordinate system. *Univ. of California Lawrence Radiation Lab. Rep.*, UCLR-6522, 1961.
- [54] Ch. Förster, W. A. Wall, and E. Ramm. On the geometric conservation law in transient flow calculations on deforming domains. *Int. J. Numer. Meth. Fluids*, 50:1369–1379, 2006.
- [55] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer Berlin, 2002.
- [56] Y. G. Lai and A. J. Przekwas. A Finite-Volume Method For Fluid Flow Simulations With Moving Boundaries. *International Journal of Computational Fluid Dynamics*, 2:1061–8562, 1994.
- [57] Philippe Geuzaine, Céline Grandmont, and Charbel Farhat. Design and analysis of ALE schemes with provable second-order time-accuracy for inviscid and viscous flow simulations. *Journal of Computational Physics*, 191(1):206 – 227, 2003.

- [58] Charbel Farhat, Philippe Geuzaine, and Céline Grandmont. The Discrete Geometric Conservation Law and the Nonlinear Stability of ALE Schemes for the Solution of Flow Problems on Moving Grids. *Journal of Computational Physics*, 174(2):669 – 694, 2001.
- [59] A. Jameson, W. Schmidt, and E. Turkel. Numerical Solutions of the Euler Equations by Finite Volume Methods Using Runge-Kutta Time-Stepping Schemes. *AIAA Paper 81-1259*, 1981.
- [60] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer-Verlag, Heidelberg, 1985.
- [61] A. Brandt. Multi-Level Adaptive Solutions to Boundary-Value Problems. *Math. Comp.*, 31:333–390, 1977.
- [62] A. Jameson. An Assessment of Dual-Time Stepping, Time Spectral and Artificial Compressibility Based Numerical Algorithms for Unsteady Flow with Applications to Flapping Wings. *AIAA Paper 2009-4273*, 2009.
- [63] B. Pritz, R. Werner, and M. Gabi. A New Sliding Interface Method for Fluid Machinery Calculations. *Proceedings of the 14th of International Symposium on Transport Phenomena and Dynamics of Rotating Machinery (ISROMAC-14), Honolulu, USA*, 2012.
- [64] Z. Liu and D. L. Hill. Issues Surrounding Multiple Frames of Reference Models for Turbo Compressor Applications. *International Compressor Engineering Conference*, Paper 1369, 2000.
- [65] Cleve Ashcraft, Daniel Pierce, David K. Wah, and Jason Wu. The Reference Manual for SPOOLES, Release 2.2: An Object Oriented Software Library for Solving Sparse Linear Systems of Equations, 1999.
- [66] Guido Dhondt. *The Finite Element Method for Three-Dimensional Thermomechanical Applications*. John Wiley & Sons, 2004.
- [67] Adnan Ibrahimbegovic. *Nonlinear Solid Mechanics. Theoretical Formulations and Finite Element Solution Methods*. Springer, 2009.
- [68] M. Hojjat, E. Stavropoulou, T. Gallinger, U. Israel, R. Wuchner, , and K.-U. Bletzinger. Fluid-Structure Interaction in the Context of Shape Optimization and Computational Wind Engineering. In *Fluid Structure Interaction II*. Springer Berlin Heidelberg, 2010.
- [69] S. Turek, J. Hron, M. Madlik, M. Razzaq, H. Wobker, and J. Acker. Numerical simulation and benchmarking of a monolithic multigrid solver for fluid–structure interaction problems with application to hemodynamics. In H. Bungartz, M. Mehl, and M. Schäfer, editors, *Fluid-Structure Interaction II: Modelling, Simulation, Optimisation*. Springer, 2010. doi 10.1007/978-3-642-14206-2.
- [70] Philippe Destuynder and Jocelyne Vétillard. Modelling and control of noise in a flexible flow duct. *Computer Methods in Applied Mechanics and Engineering*, 197(19-20):1801 – 1812, 2008. Computational Methods in Fluid-Structure Interaction.

- [71] C. A. Felippa, K. C. Park, and M. R. Ross. A Classification of Interface Treatments for FSI. In Hans-Joachim Bungartz, Miriam Mehl, and Michael Schäfer, editors, *Fluid Structure Interaction II*, volume 73 of *Lecture Notes in Computational Science and Engineering*, pages 27–51. Springer Berlin Heidelberg, 2010.
- [72] K. C. Park and C. A. Felippa. Partitioned analysis of coupled systems. *Chapter 3 in Computational Methods for Transient Analysis*, T. Belytschko and T. J. R. Hughes, eds., North-Holland, Amsterdam-New York, 157-219, 1983.
- [73] Wolfgang Wall, Axel Gerstenberger, Peter Gamnitzer, Christiane Förster, and Ekkehard Ramm. Large Deformation Fluid-Structure Interaction - Advances in ALE Methods and New Fixed Grid Approaches. In Hans-Joachim Bungartz and Michael Schäfer, editors, *Fluid-Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 195–232. Springer Berlin Heidelberg, 2006. 10.1007/3-540-34596-5-9.
- [74] Jaroslav Hron and Stefan Turek. A Monolithic FEM/Multigrid Solver for an ALE Formulation of Fluid-Structure Interaction with Applications in Biomechanics. 53:146–170, 2006. 10.1007/3-540-34596-5/7.
- [75] Thomas Dunne and Rolf Rannacher. Adaptive Finite Element Approximation of Fluid-Structure Interaction Based on an Eulerian Variational Formulation. 53:110–145, 2006.
- [76] Matthias Heil and Andrew Hazel. oomph-lib - An Object-Oriented Multi-Physics Finite-Element Library. In Timothy J. Barth, Michael Griebel, David E. Keyes, Risto M. Nieminen, Dirk Roose, Tamar Schlick, Hans-Joachim Bungartz, and Michael Schäfer, editors, *Fluid-Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 19–49. Springer Berlin Heidelberg, 2006. 10.1007/3-540-34596-5_2.
- [77] ADINA R & D. ADINA v8.6.1. 2009.
- [78] C. Farhat, K. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Comput Methods Appl Mech Eng*, 195:1973–2001, 2006.
- [79] Matthias Heil, Andrew L. Hazel, and Jonathan Boyle. Solvers for large-displacement fluid-structure interaction problems: segregated versus monolithic approaches. *Computational Mechanics*, 2007.
- [80] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157(1-2):95 – 114, 1998.
- [81] Jan Steindorf. *Partitionierte Verfahren für Probleme der Fluid-Struktur Wechselwirkung*. PhD thesis, Fachbereich für Mathematik und Informatik der Technischen Universität Braunschweig, 2002.

- [82] M. Lesoinne and C. Farhat. Stability analysis of dynamic meshes for transient aeroelastic computations. *AIAA Paper no. 93-3325, 11th AIAA Computational Fluid Dynamics Conference*, 6-9 July 1993, Orlando, FL.
- [83] D. Filsinger, J. Szwedowicz, and O. Schäfer. Approach to Unidirectional Coupled CFD-FEM Analysis of Axial Turbocharger Turbine Blades. *Journal of Turbomachinery*, 124, 2002.
- [84] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method, Band I, II & III*. Elsevier Butterworth-heinemann, Oxford, 2005.
- [85] Ulrich Küttler and Wolfgang Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43:61–72, 2008. 10.1007/s00466-008-0255-5.
- [86] D. P. Mok. *Partitionierte Lösungsansätze in der Strukturdynamik und der Fluid-Struktur-Interaktion*. PhD thesis, Institute of Structural Mechanics, University of Stuttgart, 2001.
- [87] Rainald Löhner, Juan Cebral, Chi Yang, Joseph Baum, Eric Mestreau, and Orlando Soto. Extending the Range and Applicability of the Loose Coupling Approach for FSI Simulations. In Hans-Joachim Bungartz and Michael Schäfer, editors, *Fluid-Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 82–100. Springer Berlin Heidelberg, 2006. 10.1007/3-540-34596-5_4.
- [88] S. R. Idelsohn, F. Del Pin, R. Rossi, and E. Oñate. Fluid-structure interaction problems with strong added-mass effect. *Int. J. Numer. Meth. Engng*, 80:1261–1294, 2009.
- [89] Christiane Förster, Wolfgang A. Wall, and Ekkehard Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1278 – 1293, 2007.
- [90] P. Causin, J. F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194(42-44):4506–4527, 2005.
- [91] C. Farhat. Parallel and distributed solution of coupled nonlinear dynamic aeroelastic response problems. In *Parallel Solution Methods in Computational Mechanics*, Chichester, John Wiley & Sons. Papadrakakis, M., 1997.
- [92] S. Piperno, C. Farhat, and B. Larrouturou. Partitioned procedures for the transient solution of coupled aeroelastic problems. *Computational Methods Appl. Mech. Eng.*, 124:79–112, 1995.
- [93] Ralf Massjung, Jörg Hurka, Wolfgang Dahmen, and Josef Ballmann. On well-Posedness and Modelling for Nonlinear Aeroelasticity. In J. Ballmann, editor, *Flow Modulation and Fluid-Structure Interaction at Airplane Wings*, volume 84 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, chapter 4. Springer, 2003.

- [94] C. Farhat and M. Lesoinne. Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems. *Computer Methods in Applied Mechanics and Engineering*, 182(3-4):499 – 515, 2000.
- [95] S. Piperno. Explicit/implicit fluid/structure staggered procedures with a structural predictor and fluid subcycling for 2D inviscid aeroelastic simulations. *Int. J. Num. Meth. Fluids*, 25:1207–1226, 1997.
- [96] S. Piperno and C. Farhat. Design and evaluation of staggered partitioned procedures for fluid-structure interaction simulations. Technical report, INRIA, Sophia Antipolis Cedex, September 1997.
- [97] P. Le Tallec and J. Mouro. Fluid-structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190:3039–3067, 2001.
- [98] Qun Zhang and Toshiaki Hisada. Studies of the strong coupling and weak coupling methods in FSI analysis. *International Journal for Numerical Methods in Engineering*, 60, 2004.
- [99] C. Grandmont. *Analyse mathématique et numérique de quelques problèmes d'interaction fluide-structure*. PhD thesis, Université de Paris, 1998.
- [100] Santiago Badia and Ramon Codina. On some fluid-structure iterative algorithms using pressure segregation methods. Application to aeroelasticity. *Int. J. Numer. Meth. Engng*, 72:46–71, 2007.
- [101] M. A. Fernández, J.-F. Gerbeau, and C. Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *Int. J. Numer. Meth. Engng*, 69:794–821, 2007.
- [102] P. LeTallec and S. Mani. Numerical analysis of a linearised fluid-structure interaction problem. *Numerische Mathematik*, 87(2):317–354, 2001.
- [103] J. Mouro. Numerical simulation of nonlinear fluid structure interactions problems and application to hydraulic shock-absorbers. *Proc. Third World Conf. Appl. Computational Fluid Dynamics (May 19-23, 1996) Basel World User Days CFD*, 1996.
- [104] M. Schäfer, D. C. Sternel, G. Becker, and P. Pironkov. Efficient Numerical Simulation and Optimization of Fluid-Structure Interaction. In Hans-Joachim Bungartz, Miriam Mehl, and Michael Schäfer, editors, *Fluid Structure Interaction II*, volume 73 of *Lecture Notes in Computational Science and Engineering*, pages 131–158. Springer Berlin Heidelberg, 2010.
- [105] C. Wood, A. J. Gil, O. Hassan, and J. Bonet. A partitioned coupling approach for dynamic fluid-structure interaction with applications to biological membranes. *Int. J. Numer. Meth. Fluids*, 57:555–581, 2008.
- [106] Manuel Barcelos and Kurt Maute. Aeroelastic design optimization for laminar and turbulent flows. *Computer Methods in Applied Mechanics and Engineering*, 197(19-20):1813 – 1832, 2008. Computational Methods in Fluid-Structure Interaction.

-
- [107] R. Glowinski, T.-W. Pan, and J. Periaux. A fictitious domain method for external incompressible viscous flow modeled by Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 112(1-4):133 – 148, 1994.
- [108] Wulf Dettmer and Djordje Perić. On the coupling between fluid flow and mesh motion in the modelling of fluid-structure interaction. *Computational Mechanics*, 43:81–90, 2008. 10.1007/s00466-008-0254-6.
- [109] Tayfun E. Tezduyar. Finite element methods for flow problems with moving boundaries and interfaces. *Arch. Comput. Method Eng.*, 8:83–130, 2001.
- [110] T. E. Tezduyar, S. Sathe, and K. Stein. Solution techniques for the fully discretized equations in computation of fluid-structure interactions with the space-time formulations. *Comput Meth Appl Mech Eng*, 195:5743–5753, 2006.
- [111] J. Vierendeels, J. Degroote, S. Annerel, and R. Haelterman. Stability Issues in Partitioned FSI Calculations. In Hans-Joachim Bungartz, Miriam Mehl, and Michael Schäfer, editors, *Fluid Structure Interaction II*, volume 73 of *Lecture Notes in Computational Science and Engineering*. Springer Berlin Heidelberg, 2010.
- [112] R. Ahrem, A. Beckert, and H. Wendland. A Meshless Spatial Coupling Scheme for Large-scale Fluid-Structure-Interaction Problems. *CMES*, 12, no.2:121–136, 2006.
- [113] J. R. Cebal and R. Löhner. Conservative load projection and tracking for fluid-structure interaction problems. *AIAA Journal*, 35, no4:687–692, 1997.
- [114] R. Löhner. Robust, vectorized search algorithms for interpolation on unstructured grids. *J. Comput. Phys.*, 118:380–387, 1995.
- [115] N. Maman and C. Farhat. Matching fluid and structure meshes for aeroelastic computations: a parallel approach. *Comput. Struct.*, 54:779–785, 1995.
- [116] Fraunhofer Institute for Algorithms and Scientific Computing SCAI. *MpCCI Technical Reference, Version 3.0.4*, June 2005.
- [117] Gergely Bárdossy. Studie zur Fluid-Struktur-Kopplung im Hinblick auf eine zukünftige Fluid-Akustik-Struktur-Kopplung. Master’s thesis, Fachgebiet Strömungsmaschinen, Universität Karlsruhe, 2006.
- [118] C. Bernardi, Y. Maday, and T. Patera. A new non-conforming approach to domain decomposition; the mortar element method. *Collège de France Seminar, Nonlinear Partial Differential Equations and their Applications*, 1990.
- [119] C. und Y. Maday Lacour. Two different approaches for matching non- conforming grids: the mortar element method and the feti method. *BIT*, 37(3):720–738, 1997.
- [120] D. C. Stempel, M. Schäfer, M. Heck, and S. Yigit. Efficiency and accuracy of fluid-structure interaction simulations using an implicit partitioned approach. *Computational Mechanics*, 43:103–113, 2008.

- [121] Michael Schäfer, Marcus Heck, and Saim Yigit. An Implicit Partitioned Method for the Numerical Simulation of Fluid-Structure Interaction. 53:171–194, 2006.
- [122] Serge Piperno and Charbel Farhat. Partitioned procedures for the transient solution of coupled aeroelastic problems - Part II: energy transfer analysis and three-dimensional applications. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3147 – 3170, 2001.
- [123] Iris Pantle, G. Bárdossy, and Martin Gabi. A Hybrid Approach for Investigating Fluid-Structure Interactions Combining Classical CFD with FEM Methods. In *Proceedings 9th International Symposium on Experimental and Computational Aerothermodynamics of Internal Flows/9th ISAIF, GyeongJu, Korea, 2009*.
- [124] Dominik Jürgens. Advanced Software Engineering for FSI Applications. *Fluid-Structure Interaction. Theory, Numerics and Applications.*, pages 171– 178, Herrsching am Ammersee, 29.9.-1.10.2008.
- [125] MPI: A Message-Passing Interface Standard, Version 2.2.
- [126] William Gropp, Ewing Lusk, and Rajeev Thakur. *Using MPI-2. Advanced Features of the Message-Passing Interface*. The MIT Press, 1999.
- [127] *Abaqus/CAE 6.12-2*. ©Dassault Systemès, 2012.
- [128] Copyright ©1998 Guido Dhondt and Klaus Wittig. www.calculix.de.
- [129] ANSYS, ANSYS Workbench, AUTODYN, CFX, FLUENT and any and all ANSYS, Inc. brand, product, service and feature names, logos and slogans are registered trademarks or trademarks of ANSYS, Inc. or its subsidiaries in the United States or other countries. [ICEM CFD is a trademark used by ANSYS, Inc. under license.] All other brand, product, service and feature names or trademarks are the property of their respective owners.
- [130] Pablo Mosquera Michaelsen, Balázs Pritz, and Martin Gabi. A fluid-structure-interaction tool by coupling of existing codes. In *Proceedings of the 20th European MPI Users' Group Meeting, EuroMPI '13*, pages 157–162, New York, NY, USA, 2013. ACM.
- [131] Jorge Pereira Gomes and Hermann Lienhart. Experimental Study on a Fluid-Structure Interaction Reference Test Case. In *Fluid Structure Interaction*. Springer Berlin Heidelberg, 2006.
- [132] S. Cerqueira, G. Chaineray, B. Courbet, M. Errera, F. Feyel, and F. Vuillot. Validation of Fluid-Structure Coupling computations for solid propellant rocket motors applications. *3rd European Conference for Aerospace Sciences*, 2009.
- [133] Philipp Mattern, Friedrich Fröhlich, Iris Pantle, and Martin Gabi. Experimentelle PIV-Messung eines umströmten Zylinders mit elastischer Struktur im Nachlauf. *Fachtagung "Lasermethoden in der Strömungsmesstechnik"*, 7- 9th September 2010, Cottbus.

- [134] Stefan Turek and Jaroslav Hron. Proposal for Numerical Benchmarking of Fluid-Structure Interaction between an Elastic Object and Laminar Incompressible Flow. In Timothy J. Barth, Michael Griebel, David E. Keyes, Risto M. Nieminen, Dirk Roose, Tamar Schlick, Hans-Joachim Bungartz, and Michael Schäfer, editors, *Fluid-Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 371–385. Springer Berlin Heidelberg, 2006. 10.1007/3-540-34596-5_15.
- [135] O. O. Bendiksen and G. A. Davis. Nonlinear Traveling Wave Flutter of Panels in Transonic Flow. *AIAA*, 95-1486, 1995.
- [136] Armin Beckert and Holger Wendland. Multivariate interpolation for fluid-structure-interaction problems using radial basis functions. *Aerospace Science and Technology*, 5(2):125 – 134, 2001.

Curriculum Vitae

Name: Pablo Mosquera Michaelsen (Ing.)
Date of birth: July 10, 1983
Place of birth: San Miguel, Province of Bs. As., Argentina
Nationality: Argentinian

1997-2002 Secondary school.
Instituto Tecnológico Philips Argentina
Buenos Aires, Argentina.

2003-2008 Studies of Mechanical Engineering.
Instituto Tecnológico de Buenos Aires
Buenos Aires, Argentina.

2008 Ingeniero Mecánico (Mechanical Engineer)
Graduated with honors.

2008-2009 Software developer at SIM&TEC S.A.
Buenos Aires, Argentina.

2009-2013 Doctoral scholarship holder of the German
Academic Exchange Service (DAAD).

2009-2014 Research fellow at the Institute of Fluid Machinery,
Karlsruhe Institute of Technology
Karlsruhe, Germany.

Since 2014 Design applications developer for Secondary Air
System technologies, Large Gas Turbines division
SIEMENS AG.
Mülheim an der Ruhr, Germany