UPC CTTC

Developing Numerical Methods for Fully–Coupled Nonlinear Fluid–Structure Interaction Problems

Heat and Mass Technological Centre
Departament de Màquines i Motors Tèrmics
Universitat Politècnica de Catalunya

Alireza Naseri
Doctoral Thesis

Developing Numerical Methods for Fully–Coupled Nonlinear Fluid–Structure Interaction Problems

Alireza Naseri

PHD THESIS

submitted to the

Departament de Màquines i Motors Tèrmics
Escola Superior d'Enginyeries Industrial, Aeroespacial i Audiovisual de Terrassa
Universitat Politècnica de Catalunya

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Thermal Engineering

Terrassa (Barcelona), June 2019

Developing Numerical Methods for Fully–Coupled Nonlinear Fluid–Structure Interaction Problems

Alireza Naseri

Thesis Supervisor

Prof. Carlos David Pérez-Segarra

Thesis Tutor

Prof. Assensi Oliva

Committee Board

Prof. Antonio Pascau
Universidad de Zaragoza, Spain

Prof. Jesús Castro
Universitat Politècnica de Catalunya, Spain

Prof. Ricardo Vinuesa
KTH Royal Institute of Technology, Sweden

External Referees

Prof. Miriam Mehl
University of Stuttgart, Germany

Prof. Peyman Givi
University of Pittsburgh, USA

Eritis sicut Deus scientes bonum et malum.

Acknowledgements

I would like to thank my supervisors for their guidance and support. I would also like to thank the members of the committee and the external reviewers for their time and consideration in reviewing this thesis.
I thank my friends and my family for their support.
Finally, I would like to thank AGAUR of Generalitat de Catalunya, for funding my studies through a FI scholarship.

Abstract

This thesis is dedicated to developing numerical methods to solve fluid–structure interaction (FSI) problems. FSI features in a vast range of physical systems and has a wide application in engineering. In the first chapter of this document we provide a literature review on different numerical methods for FSI problems. The work of this thesis is focused on the partitioned methods, mostly due to their modularity, robustness and reliability. In a partitioned approach, separate solvers are used for the fluid and structural sub–problem domains and a coupling method is devised to account for their mutual interaction. Partitioned approach is preferred in this work because it allows using the most adapted numerical methods for each sub–problem domain, and also the use of previously–developed and optimized solver codes. Moreover, the thesis is focused on FSI problems with strong added–mass effect, which are more challenging to solve numerically. For such FSI problems, normally an implicit partitioned method is used which enforces the coupling conditions on the interface through coupling iterations between the fluid and structural solvers. However, these methods are computationally expensive due to several coupling iterations which require solving the fluid and structural problems several times per time step. In this work we follow a semi–implicit approach to develop stable, efficient and accurate numerical methods for FSI problems. In a semi–implicit method, the fluid pressure term is segregated and strongly coupled to the structure via coupling iterations. However, the remaining fluid terms and the geometrical nonlinearities are treated explicitly. Strong coupling of the fluid pressure term provides for the stability of the method in FSI problems with strong added–mass effect, while loose coupling of the remaining terms reduces the computational cost of the simulations.

The work of this thesis could be divided into three major parts. In the first part, we have developed a simple, efficient and robust semi–implicit coupling method for FSI problems with strong added–mass effect. The proposed method is simple and modular, as it is developed with the least possible mathematical and numerical complications. An extensive set of numerical tests were carried out and the results were compared both to literature data (numerical and experimental), as well as domestic results obtained by using a fully–implicit coupling method. Results showed that the proposed method considerably reduces the computational cost of the simulations without degrading the stability or accuracy of the solution. Moreover, the robustness of the method is demonstrated through numerical tests, as the method was shown to remain stable, accurate and efficient for a wide range of FSI problems including ones with very large deformations. Furthermore, in this part of the work, we have tried to further analyze the semi–implicit methods in order to gain a better understanding of several unaddressed issues concerning different aspects of these methods.

The second major part of this thesis is focused on the temporal accuracy of the

semi–implicit coupling methods for FSI problems. The semi–implicit methods in the literature appear to be only first–order in time. Most semi–implicit methods rely on using a projection method for the fluid equations (in order to segregate the fluid pressure term), while extending the temporal accuracy of the projection methods is not straightforward. Moreover, mesh–conforming FSI solution methods require solving the ALE form of the Navier–Stokes equations on a moving mesh, which does not necessarily preserve the order of accuracy of the method on a fixed grid. Furthermore, if the FSI coupling technique is not properly designed, the second-order accuracy for the coupled problem is not guaranteed, even though each sub–problem possessed such accuracy. In this work, we have proposed a second–order time accurate semi–implicit method for FSI problems and demonstrated its second–order accuracy through rigorous numerical tests. For this purpose we have taken four steps; i) presented an incremental projection method and discretization in time that actually yields second–order accuracy for fluid pressure and velocity, ii) derived specific projection–consistent boundary conditions for all fluid boundaries including the interface with the solid, iii) developed an ALE scheme on a moving grid and evaluated the arisen geometrical terms with second–order accuracy and, iv) properly coupled the fluid and structural solvers in order to retain the second–order accuracy for a coupled nonlinear FSI problem. The resulting method was shown to deliver second–order accuracy for all the variables of interest in realistic FSI test cases.

The last major part of this thesis is concerned with computational efficiency and parallel scalability of the developed methods for numerical solution of complex FSI problems on massively–parallel machines. We have presented a scalable parallel framework for partitioned solution of FSI problems through multi–code coupling. Following a partitioned approach, we have used preexisting solver codes for each sub–problem and efficiently coupled them on the common interface. Two instances of our in–house software is used to solve the fluid and structural sub–problems. The communication between the single–physics solvers are carried out using an external coupling library. The semi–implicit methods developed in the previous parts of this thesis are used in the proposed framework. The parallel implementation of the coupled framework, different levels of communication (intra-solver and inter-code), and load balancing between the solvers are presented and analyzed. Both fluid and structural solvers use distributed–memory parallelism and all the communications are point–to–point and non–blocking. Inside each single–physics solver, the load is balanced by dividing the computational domain into fairly equal blocks for each process. In addition, a load balancing model is used in the inter–code level to minimize the overall idle time of the processes. Practical test cases in the context of biological flow (blood flow inside deformable vessels) are solved and the parallel efficiency and scalability of the coupled framework are demonstrated.

Contents

List of Figures

List of Tables

Nomenclature

Latin symbols

$\mathbf{u}$	fluid velocity
$\mathbf{w}$	domain velocity
$\mathbf{c}$	fluid ALE convective velocity
p	fluid pressure
$\mathbf{I}$	the unity tensor
$\mathbf{d}$	structural displacement
$\mathbf{v}$	structural velocity
$\mathbf{B}$	left Cauchy-Green deformation tensor
$\mathbf{P}$	first Piola-Kirchhoff tensor
$\mathbf{S}_s$	second Piola-Kirchhoff tensor
$\mathbf{F}$	material deformation tensor
J	determinant of material deformation tensor
E	Young modulus
$\mathbf{n}$	surface normal vector
$\mathbf{A}$	area vector
A	surface area
v	volume
s	surface
h	solid wall thickness
I	second moment of area

Greek symbols

Ω	computational domain
Γ	fluid–structure interface
$\boldsymbol{\sigma}_f$	fluid stress tensor
$\boldsymbol{\gamma}$	fluid strain rate tensor
ρ_f	fluid density
μ_f	fluid dynamic viscosity
ϕ	projector in numerical methods for fluid equations
φ	radial basis function
$\boldsymbol{\sigma}_s$	structural Cauchy stress tensor

ρ_s	structural density
μ_s	Lamé's parameter
λ_s	Lamé's parameter
ν	Poisson ratio
$\boldsymbol{\tau}$	surface tangential vector

Subscripts

f	related to fluid
s	related to structure
Γ	related to fluid–structure interface
b	related to boundary
∞	far field condition

Superscripts

n	time step indicator
k	coupling iteration indicator
p	predicted value
$*$	intermediate value
T	transpose

1

Introduction

This thesis is aimed at developing novel numerical methods to solve systems including the interaction of a fluid flow and an structure. In this chapter we present an introduction to fluid-structure interaction (FSI) problems and the numerical method to solve them. The chapter includes an extensive literature review on different numerical methods for FSI problems. The scope, objectives and the outline of the thesis are presented at the end of this chapter.

1.1 Fluid-structure interaction problems

Fluid-structure interaction refers to problems with bilateral interaction between a fluid flow and a moving or deforming structure. Fluid flow exerts surface forces on the structure which causes its movement or deformation. The movement of the solid boundary, in return, alters the flow field. Therefore, there is a mutual interaction between the fluid and the structure which couples them on their common interface.

A very wide range of applications is cited for FSI, ranging from civil engineering to biomechanics. An interesting example of FSI application in biomedical engineering is simulation of blood flow inside deformable vessels in human arterial system (e.g. [1–7], see figure 1.1(Left)). The simulations may help improving the quality of artificial blood vessels and predicting the rupture of aneurysms during specific medical treatments or surgeries. A thoracic aortic aneurysm, as an example, is a weakened and dilated area in the upper part of the aorta above the diaphragm. It grows by the patient's age and its rupture could pose a life risk to the patient. The condition is defined as the dilatation of the aorta by more than 50% [8]. The aneurysm is normally treated by surgery or endovascular repair, depending on the level of the rupture risk [9]. Therefore, understanding the process of formation, growth and rupture of the aneurysms has a great clinical value. Considering the flexibility of the vessels and the coupled FSI problem is necessary to accurately analyze the aneurysm and predict its rupture risk. The FSI problem in aneurysm is studied, among others,

in [2–4, 6].

Another interesting example of FSI application in biomedical engineering is simulation of blood flow passing through the heart valves (e.g. [10–17], see figure 1.1(Right)). The movement of the valve leaflets depend on the blood flow and the pressure gradient over the leaflets, creating a coupled FSI problem. There are many diseases that could affect the proper functioning of the heart valves (both aortic and mitral valves). In severe cases, the heart valve might be completely replaced by an artificial prostheses valve. Various mechanical heart valve designs have been developed in the past year. Evaluating the performance and reliability of such artificial valves requires a proper understanding of the complex flow field and the interaction of the leaflets with the flow.

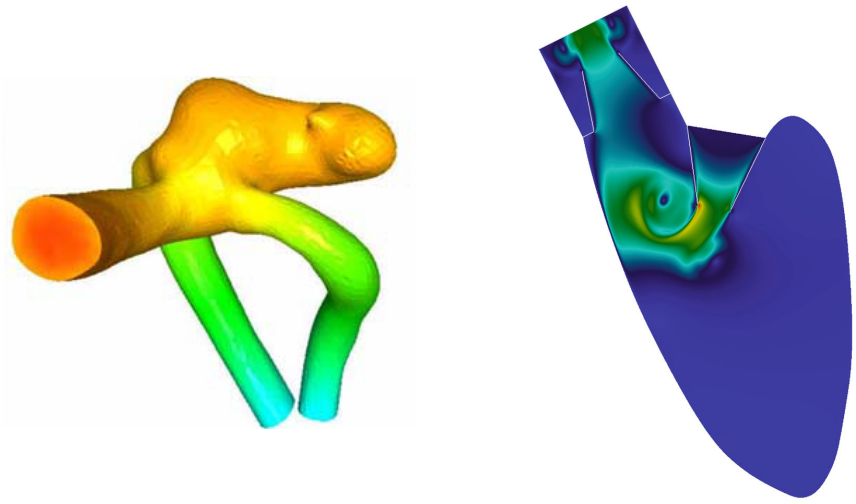

Figure 1.1: Applications of FSI in biomedical engineering. Left: blood flow inside a cerebral aneurysm (figure from [1]); Right: blood flow passing through the heart valve (figure from [13]).

Application of FSI in civil and marine engineering includes predicting the vortex-induced vibration (VIV) of elastically mounted rigid bodies (e.g. [18–26], see figure 1.2). VIV is an important phenomena in submerged structures in offshore engineering (e.g. pipes, risers, mooring lines, etc.). Structures subject to wind may also experience VIV (e.g. chimneys, bridge suspension cables, power transmission lines, etc.). Evaluating the life-time of such structures and the fatigue damage requires understanding the undergoing VIV phenomenon. VIV effects must be considered in the design of many offshore structures to assure their proper functioning.

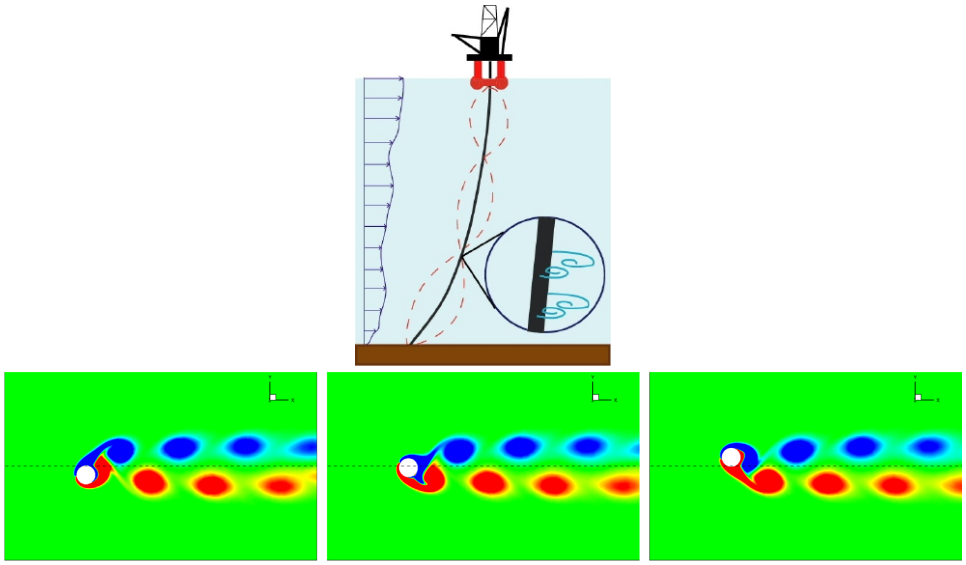

Figure 1.2: Vortex–induced vibration in submerged structures. Top: an schematic offshore structure; Bottom: VIV of an elastically mounted cylinder (figure from [25]).

The vast application of FSI in different scientific and engineering disciplines demonstrates the importance of this class of problems. A great effort has been made by the research community during the last decades to better understand the underlying phenomena in FSI problems and to be able to appropriately predict the behavior of the systems involving such phenomena. Numerical studies account for a significant part of the works aimed to understanding FSI problems, along with experimental and analytical studies. In the following, a review of different numerical methods used to solve FSI problems is presented.

1.2 Numerical methods for FSI problems

Numerical simulation is a strong tool to study physical systems and analyze and improve engineering designs. They are often preferred to experimental analyses due to their lower cost, simpler conduct and extensive testing possibilities. Numerical simulations have become even more popular thanks to the rapid growth in computational power that permits studying increasingly larger and more complex systems.

Accurate mathematical models and reliable numerical methods are required in order to simulate complex physical systems on a computer. The accuracy of the

simulations would depend on the accuracy of the mathematical model used for modeling the physical phenomena and the precision of the numerical methods used to solve the mathematical equations. Moreover, numerical stability is another issue that must be considered, as the accumulation of the computer round-off errors might become unbounded and cause the instability of the solution. With all in mind, developing stable and accurate numerical methods to solve complex physical systems, e.g. FSI problems, is a challenging task which has been receiving an extensive attention from the research community.

In an FSI problem, the physical laws governing the fluid flow and structural displacement remain in force in their respective domains, along with new conditions which are physical consistency and equilibrium on the interface. Therefore, any FSI problem could be seen as a three-part problem, a fluid sub-problem domain where the fluid equations govern, a solid sub-problem domain with the structural governing equations, and a common boundary or interface where the equilibrium conditions hold.

Broadly, there are two approaches to solve FSI problems which are called monolithic and partitioned methods. In a monolithic approach the fluid and structural equations are solved simultaneously as a single system of equations. Therefore, the equilibrium conditions on the interface and the interaction of the domains are inherently taken into account. Examples of monolithic methods could be found in [2,27,28]. Partitioned methods, on the other hand, use separate solvers for fluid and structural equations and adopt a coupling scheme to account for the interaction of the domains. The coupling scheme determines the order and frequency in which the fluid and structural equations should be solved. It also determines the manner of communication and information exchange between the two solvers which is essentially restricted to the fluid-structure interface. Examples of partitioned methods could found in [29–32]. Recent review articles on partitioned methods could be consulted in [33,34].

The main advantage of the monolithic approach is the elimination of the need for any further coupling technique at the fluid-structure interface, which reduces the complexity of the problem. In contrast, coupling the fluid and structural solvers in a partitioned approach introduces a new and challenging aspect to the problem with many issues to consider. In some applications, this fact may give a great superiority to the monolithic methods. On the other hand, partitioned approach allows using the most-adapted and well-validated numerical numerical methods for each sub-problem. These methods are previously tested and verified on diverse cases which greatly increases the reliability of the FSI simulations. Conversely, the monolithic approach normally requires using the same numerical methods to discretize and solve the fluid and structural equations, while they possess different mathematical properties and have their own specific considerations. This may cause monolithic methods to be less reliable or less optimized in some applications [33,34]. Another

important advantage of the partitioned approach is its ability to use the *heritage codes*, i.e. previously developed fluid and structural solvers, which saves a large software development effort and increases modularity of the code. On the contrary, monolithic approach generally requires a large software development effort and usually results in a less modular solver [33,34]. Based on the above discussion, it is evident that both monolithic and partitioned methods have their own merits and drawbacks. In this work we will focus on the partitioned methods as we would like to use specific (and optimized) numerical methods for each sub-problem and we are eager to utilize our previously-developed in-house solvers.

As explained above, the partitioned approach treats the FSI problem as two sub-problems (the fluid and structural problems) which are coupled on the common interface through equilibrium conditions (the kinematic and dynamic equilibrium). In a partitioned method, the equilibrium conditions on the interface are applied as boundary conditions on each sub-problem, through a decomposition method. Dirichlet-Neumann decomposition is a classical and widely used method in partitioned solution of FSI problems (see e.g. [35–37]). The name of the method indicates that a Dirichlet boundary condition (coming from kinematic equilibrium) is used for the fluid equations and a Neumann boundary condition (coming from dynamic equilibrium) for the structure. Therefore, the fluid equations are solved for a known displacement of the solid, while the structural equations are solved for a known stress on the interface. This is a simple decomposition and it is consistent with the most common numerical methods for fluid and structural equations. There is a more recent class of Robin-based decomposition methods that use a Robin boundary condition for the fluid and either a Robin or Neumann boundary condition for the structure [38–40].

Partitioned methods could be further divided into explicit (or loosely coupled) and implicit (or strongly coupled) schemes. In an explicit coupling method, the fluid and structural equations are solved in sequence and only once at every time step. Consequently, explicit methods do not satisfy the exact coupling condition at the fluid-structure interface. The most basic explicit scheme is the conventional serial staggered method [41]. Implicit methods, in contrast, enforce the equilibrium condition at the interface by means of coupling iterations between the fluid and structural solvers at each time step. Fixed-point (Gauss-Seidel or Jacobi) iterations [30, 33] and Newton-based methods [36, 42–45] are the most commonly used techniques to carry out the FSI coupling iterations. Vector extrapolation methods have also been used for this purpose [46]. The execution of the fluid and structural solvers could be staggered (sequential) or parallel (simultaneous). Staggered methods are very common, although they have a very limited computational efficiency when used on a multi–node cluster, due to a severe load imbalance [32, 47]. This is a very serous drawback, considering that practical FSI problems are computationally demanding and often need to be run on massively parallel machines. Parallel execution with

fixed-point iterations normally does not show a good performance, while parallel execution together with a quasi-Newton method result in a very efficient and robust method [32,45,47].

Explicit methods work well for aeroelastic simulations and problems involving compressible flows [29,48]. However, used along with a classical Dirichlet-Neumann decomposition, explicit methods are unstable for a wide range of problems, especially ones with an incompressible flow, slender domain and low solid/fluid density ratios (similar density of fluid and solid). The instability happens regardless of the time step size or specific discretization of the equations at each sub-problem domain. It is inherent to the coupling method and is often called *added-mass effect* or *added-mass instability*. The instability rises due to the fact that fluid forces in the explicit coupling depend upon a predicted displacement of the structure, rather than the correct one. As the structure moves, it has to accelerate the bulk of the fluid around it as well. Thus, part of the fluid acts as an extra mass in the structural dynamics system, which has given rise to the name added-mass effect. This effect is particularly strong when densities of the fluid and the structure are similar. For any loosely coupled method there is a limiting density ratio, beyond which the method would suffer from instability issues [49,50]. While the added-mass effect causes instability in the loosely coupled schemes, it deteriorates convergence of the strongly coupled methods. Therefore, a FSI problem with strong added-mass effect is also challenging for implicit methods, as it requires many coupling iterations to converge at each time step [49,50].

Implicit methods provide stable solution for FSI problems with strong added-mass effect, of which explicit methods are incapable. However, their computational cost is generally high due to the repetitive solution of the governing equations at each time step [33,34]. The applicability of strongly-coupled methods are often limited by their high cost. There has been many attempts to stabilize explicit coupling methods [51,52]. However, these stabilized explicit methods normally require solving a separate (and new) set of equations on the interface which reduces the modularity and robustness of the method and deteriorates the accuracy of the solution (particularly in the vicinity of the common interface). Robin-based methods are gaining popularity as they allow a loosely-coupled and yet added-mass-free FSI coupling. The structural inertia term is usually included in the Robin boundary condition for fluid equations which eliminates the added-mass instability (see e.g. [53,54]). However, they require using an especial fluid solver capable of handling such a Robin boundary condition. Therefore, they might not be readily usable with some of the most common fluid solvers.

A very interesting approach to tackle the problem of unstable-explicit or costly-implicit coupling methods was proposed by Fernández et al. [55], and called *semi-implicit FSI coupling*. In this technique, a fractional-step (projection) method was used for fluid equations and only the projection step was strongly coupled to the structure.

This way the fluid pressure term and the structural deformation were strongly coupled while fluid convective and diffusive terms as well as the geometrical nonlinearities (the moving mesh) were treated explicitly [55]. Therefore, the main aspect of a semi-implicit coupling method is that the fluid pressure term is segregated and strongly coupled to the structure, while the remaining fluid terms are only loosely coupled. It is argued that the fluid pressure term is the main contributor to the added-mass effect and its explicit coupling would cause instability issues [49]. Strong coupling of the fluid pressure term in a semi-implicit method avoids added-mass instability while loose coupling of the remaining fluid terms avoids excessive computational cost [55]. Thus, semi-implicit methods maintain the favorable stability of the implicit methods while reducing the computational cost. Segregation of the pressure term could be naturally achieved by using a classical Chorin-Temam projection method [56] or potentially by any other pressure splitting scheme.

A similar semi-implicit method was proposed by Breuer et al. [57,58] where the geometrical nonlinearities are also coupled implicitly. Astorino et al. [59] improved the stability of the method in [55] by using a specific Robin treatment of the explicit part of the coupling derived from Nitsche's method. An analogous idea is present in the hybrid monolithic-partitioned method of Grétarsson et al. [60] for FSI problems with compressible flow. It strongly couples the fluid pressure and solid velocity by solving them implicitly in a monolithic manner, while the remaining terms are loosely coupled in a partitioned manner [60]. Other similar semi-implicit methods are presented in [61–63] which use a characteristic-based split (CBS) scheme instead of Chorin-Temam projection method. Similar explicit-implicit splitting of the coupling is proposed in [64,65] using algebraic splitting methods instead of differential ones. These methods are based on inexact factorization of the matrix obtained after discretization of the equations in space and time. Methods in [64, 65] permit using accurate algebraic splitting methods that do not have differential counterparts (e.g. [66, 67]).

It should be noted that these semi-implicit methods are somehow different from the methods in [68–70]–which are also sometimes called semi-implicit. In the methods in [68–70], the location of the interface is treated explicitly (extrapolated in time) and the fluid mesh is moved once in a time step, however, the fluid and structural equations are solved completely at each coupling iteration (i.e. only the geometrical nonlinearities are treated explicitly). In fact, these methods could be called strongly coupled techniques on a frozen mesh.

1.3 Objectives of the thesis

Semi-implicit FSI coupling approach seems to be a promising technique for accurate and cost-optimized solution of complex FSI problems. Despite receiving attention from the research community, semi-implicit coupling techniques are far from perfect.

Many of the reported methods in the literature lack modularity and simplicity. Moreover most of the reported methods are only tested in a specific type of FSI problems and their robustness in dealing with different types of FSI problems is not evaluated. Besides, there are many unaddressed questions concerning different aspects of the semi-implicit coupling methods that require more work and attention. Moreover, the semi-implicit methods in the literature all seem to be only first-order accurate in time. Extending these methods to higher order accuracy seems a significant improvement that was not accomplished. Furthermore, these methods have mostly been used previously to solve small benchmark test cases, using very few processors or even in serial configuration. Scalable and efficiently parallel methods for FSI problems using semi–implicit coupling approach are missing from the literature.

In this Thesis, we focus on the semi–implicit partitioned approach to develop accurate, robust and efficient techniques for numerical solution of FSI problems with strong added-mass effect. We also try to address some of the open questions concerning semi-implicit methods. The main aspects in which the present work improves the existing methods are as following:

- Simplicity: The present methods are developed with the least possible mathematical and numerical complications. This results in techniques which are easy to follow and simple to implement.
- Modularity: Different components of the present methods could be flexibly separated and recombined, offering a great degree of modularity.
- Accuracy: The present methods improve the accuracy of the existing semi–implicit techniques. Particularly, we have extended the temporal accuracy of the semi–implicit methods to second–order. Moreover, we have evaluated the effect of different alterations on the overall accuracy of the semi–implicit methods.
- Robustness: We present methods that remain stable, accurate and efficient for a wide range of FSI problems, including ones with very large deformations and ones with very strong added–mass effect.
- Computational efficiency: The proposed coupling techniques along with the choices for numerical schemes and discretization, result in fast and efficient overall methodologies.
- Parallel scalability: We present a highly efficient and scalable framework to solve complex FSI problems on massively–parallel machines.

We believe this thesis significantly contributes to the development of an important class of numerical methods for fluid–structure interaction problems. It also helps to better understand different numerical aspects of these methods and their effect

on the accuracy and efficiency of the overall techniques. Moreover, we demonstrate the capability of the proposed methods to solve real–life complex FSI systems on massively–parallel machines.

1.4 Outline of the thesis

This thesis is organized in six chapters. The first chapter provides an introduction and an extensive literature review on numerical methods to solve FSI problems. The main contributions of this research work are presented in three main chapters (chapters 2-4). These chapters are self–contained and could be read on their own, without the need to read the complete document. Each chapter contains a short introduction that aims to highlight the contribution of the chapter and put it in the right context by reviewing related works in the literature. The content of each main chapter is published as original research articles in international journals and presented in different scientific conferences. The last two chapters provide, respectively, concluding remarks and possible future works.

Chapters 2 to 4 present the main contributions of this thesis. Chapter 2 represents a simple, efficient and robust semi-implicit method for FSI problems with strong added-mass effect. A through numerical analysis is carried out to verify the accuracy of the method by comparing its results against experimental data and other numerical results from the literature. Numerical test cases are chosen to be widely distinct in order to evaluate the robustness of the method. The accuracy and computational cost of the method is also compared to a fully-implicit method. Moreover, the effect of implicit or explicit treatment of the geometrical nonlinearities on the overall performance and accuracy of the developed method is studied. Besides, both fixed-point and Newton methods are used to solve the coupled interface problem and a comparison is made on their performance.

Chapter 3 represents a semi–implicit FSI method with second–order accuracy in time. It includes presentation of an incremental projection method and temporal discretization that actually yields second-order accuracy for fluid pressure, as well as velocity. Specific projection–consistent boundary conditions are derived for fluid boundaries including the interface with solid. An ALE scheme is developed on a moving domain and the associated geometrical terms are evaluated with second–order accuracy. The fluid and structural solvers are properly coupled in order to retain the second–order accuracy for a coupled nonlinear FSI problem. The second–order accuracy of the method for realistic FSI problems is demonstrated through rigorous numerical tests.

Chapter 4 represents a scalable framework for partitioned solution of FSI problems on massively–parallel machines. The framework is developed through multi–code coupling, taking advantage of the previously–developed optimized solver codes

for the single–physics domains. Semi–implicit methods developed in the previous chapters are used for the proposed framework. The communication between the single–physics solvers are carried out using an external coupling library. The parallel implementation of the coupled framework, different levels of communication, and load balancing between the solvers are described. Practical FSI test cases are studied and the scalability of the coupled framework is evaluated.

Chapter 5 summarizes and concludes the thesis, while chapter 6 contains suggestions for future works.

1.5 Background of the research group

This work was conducted in the Heat and Mass Transfer Technological Center (CTTC), at the Universitat Politècnica de Catalunya. CTTC was created in the 1990s, with research focus on mathematical modeling and numerical solution of fluid dynamics and heat transfer phenomena, complemented with experimental research for their validation. The main objective of the research work at CTTC is developing efficient fluid and thermal systems and equipments.

In the early 2000s, the simulation tools developed within the group were joined and generalized to create the CFD software package called *TermoFluids* [71]. TermoFluids is a robust general–purpose software for fluid and heat transfer problems, using state-of-the-art methods for turbulent flow [72,73], multiphase flow [74,75] and complex thermal systems [76,77], with a high computational efficiency and parallel scalability [78–80].

The methods and tools developed in this thesis have been incorporated into the TermoFluids software. Equally, many tools developed previously inside TermoFluids, have been used for this work.

1.6 Related publications

The material of this thesis have appeared in the following publications:

- **A. Naseri**, I. Gonzalez, A. Amani, C. D. Pèrez-Segarra and A. Oliva, "A second-order time accurate semi-implicit method for fluid-structure interaction problems," *Journal of Fluids and Structures*, Vol. 86 (2019), 135–155.
- **A. Naseri**, O. Lehmkuhl, I. Gonzalez, E. Bartrons, C. D. Pèrez-Segarra and A. Oliva, "A semi-implicit coupling technique for fluid-structure interaction problems with strong added-mass effect," *Journal of Fluids and Structures*, Vol. 80 (2018), 94–112.

- **A. Naseri**, I. Gonzalez, A. Amani and C. D. Pèrez-Segarra, "Second-Order Semi-Implicit Partitioned Method for Fluid-Structure Interaction Problems," In *7th European Conference on Computational Fluid Dynamics*, Glasgow, UK, June 2018.

- A. Amani, **A. Naseri**, C. D. Pèrez-Segarra and A. Oliva, "A Method for Fluid-Structure Interaction Problems with Non-Newtonian Fluid," In *7th European Conference on Computational Fluid Dynamics*, Glasgow, UK, June 2018.

- I. Gonzalez, **A. Naseri**, J. Chiva, J. Rigola and C. D. Pèrez-Segarra, "An enhanced finite volume based solver for thermoelastic materials in fluid-structure coupled problems," In *6th European Conference on Computational Mechanics*, Glasgow, UK, June 2018.

- I. Gonzalez, **A. Naseri**, J. Rigola, C. D. Pèrez-Segarra and A. Oliva, "A fluid-structure interaction solver for the fluid flow through reed type valves," In *IOP Conference Series: Materials Science and Engineering*, vol. 232, no. 1, p. 012032. IOP Publishing, 2017.

- I. Gonzalez, O. Lehmkuhl, **A. Naseri,** J. Rigola, and A. Oliva, "Fluid-structure interaction of a reed type valve," In *23rd International Compressor Engineering Conference at Purdue*, West Lafayette, USA, July 2016.

- **A. Naseri,** O. Lehmkuhl, I. Gonzalez and A. Oliva, "Partitioned semi-implicit methods for simulation of biomechanical fluid–structure interaction problems," In *Journal of Physics: Conference Series*, vol. 745, no. 3 (2016), p. 032020.

References

[1] J Gerbeau, M Vidrascu, and P Frey. Fluid-structure interaction in blood flows on geometries based on medical imaging. *Computers and Structures*, 83:155–165, 2005.

[2] Y. Bazilevs, V. M. Calo, Y. Zhang, and T. J R Hughes. Isogeometric fluid-structure interaction analysis with applications to arterial blood flow. *Computational Mechanics*, 38(4-5):310–322, 2006.

[3] Alessandro Borghi, Nigel B Wood, Raad H Mohiaddin, and X Yun Xu. Fluid-solid interaction simulation of flow and stress pattern in thoracoabdominal aneurysms: a patient-specific study. *Journal of Fluids and Structures*, 24(2):270–280, 2008.

[4] Tayfun E Tezduyar, Sunil Sathe, Matthew Schwaab, and Brian S Conklin. Arterial fluid mechanics modeling with the stabilized space–time fluid–structure interaction technique. *International Journal for Numerical Methods in Fluids*, 57:601–629, 2008.

[5] Esko Järvinen, Peter Råback, Mikko Lyly, and Juha Pekka Salenius. A method for partitioned fluid-structure interaction computation of flow in arteries. *Medical Engineering and Physics*, 30(7):917–923, 2008.

[6] Hiroshi Suito, Kenji Takizawa, Viet QH Huynh, Daniel Sze, and Takuya Ueda. Fsi analysis of the blood flow and geometrical characteristics in the thoracic aorta. *Computational Mechanics*, 54(4):1035–1045, 2014.

[7] Jingliang Dong, Zhonghua Sun, Kiao Inthavong, and Jiyuan Tu. Fluid–structure interaction analysis of the left coronary artery with variable angulation. *Computer methods in biomechanics and biomedical engineering*, 18(14):1500–1508, 2015.

[8] Eric M Isselbacher. Thoracic and abdominal aortic aneurysms. *Circulation*, 111(6):816–828, 2005.

[9] John A Elefteriades. Natural history of thoracic aortic aneurysms: indications for surgery, and surgical versus nonsurgical risks. *The Annals of thoracic surgery*, 74(5):S1877–S1880, 2002.

[10] Fotis Sotiropoulos and Iman Borazjani. A review of state-of-the-art numerical methods for simulating flow through mechanical heart valves. *Medical & biological engineering & computing*, 47(3):245–256, 2009.

[11] Sebastiaan Annerel, Joris Degroote, Tom Claessens, Patrick Segers, Pascal Verdonck, and Jan Vierendeels. The upstream boundary condition influences the

leaflet opening dynamics in the numerical FSI simulation of an aortic BMHV. *International Journal for Numerical Methods in Biomedical Engineering*, 28:745–760, 2012.

[12] Sebastiaan Annerel, Joris Degroote, Tom Claessens, Sigrid K. Dahl, Bjørn Skallerud, Leif Rune Hellevik, Peter Van Ransbeeck, Patrick Segers, Pascal Verdonck, and Jan Vierendeels. A fast strong coupling algorithm for the partitioned fluid–structure interaction simulation of BMHVs. *Computer Methods in Biomechanics and Biomedical Engineering*, pages 1–32, 2012.

[13] Kenji Takizawa, Tayfun E Tezduyar, Austin Buscher, and Shohei Asada. Space–time interface-tracking with topology change (ST-TC). *Computational Mechanics*, 54(4):955–971, 2014.

[14] Kenji Takizawa, Tayfun E Tezduyar, Austin Buscher, and Shohei Asada. Space–time fluid mechanics computation of heart valve models. *Computational Mechanics*, 54(4):973–986, 2014.

[15] S. Annerel, T. Claessens, J. Degroote, P. Segers, and J. Vierendeels. Validation of a numerical FSI simulation of an aortic BMHV by in vitro PIV experiments. *Medical Engineering and Physics*, 36(8):1014–1023, 2014.

[16] Kenji Takizawa, Tayfun E Tezduyar, Takuya Terahara, and Takafumi Sasaki. Heart valve flow computation with the integrated space–time vms, slip interface, topology change and isogeometric discretization methods. *Computers & Fluids*, 158:176–188, 2017.

[17] Mohammadali Hedayat and Iman Borazjani. Comparison of platelet activation through hinge vs bulk flow in bileaflet mechanical heart valves. *Journal of biomechanics*, 83:280–290, 2019.

[18] P Anagnostopoulos and PW Bearman. Response characteristics of a vortex-excited cylinder at low reynolds numbers. *Journal of Fluids and Structures*, 6(1):39–50, 1992.

[19] P Anagnostopoulos. Numerical investigation of response and wake characteristics of a vortex-excited cylinder in a uniform stream. *Journal of Fluids and Structures*, 8(4):367–390, 1994.

[20] R Govardhan and CHK Williamson. Modes of vortex formation and frequency response of a freely vibrating cylinder. *Journal of Fluid Mechanics*, 420:85–130, 2000.

[21] D Shiels, A Leonard, and A Roshko. Flow-induced vibration of a circular cylinder at limiting structural parameters. *Journal of Fluids and Structures*, 15(1):3–21, 2001.

[22] Matteo Luca Facchinetti, Emmanuel De Langre, and Francis Biolley. Coupling of structure and wake oscillators in vortex-induced vibrations. *Journal of Fluids and structures*, 19(2):123–140, 2004.

[23] PW Bearman. Circular cylinder wakes and vortex-induced vibrations. *Journal of Fluids and Structures*, 27(5):648–658, 2011.

[24] Jisheng Zhao, Justin S Leontini, David Lo Jacono, and John Sheridan. Fluid–structure interaction of a square cylinder at different angles of attack. *Journal of Fluid Mechanics*, 747:688–721, 2014.

[25] Ravi Chaithanya Mysa, Abouzar Kaboudian, and Rajeev Kumar Jaiman. On the origin of wake-induced vibration in two tandem circular cylinders at low Reynolds number. *Journal of Fluids and Structures*, 61:76–98, 2016.

[26] Tristan Leclercq and Emmanuel de Langre. Vortex-induced vibrations of cylinders bent by the flow. *Journal of Fluids and Structures*, 80:77–93, 2018.

[27] K. J. Bathe and Hou Zhang. Finite element developments for general fluid flows with structural interactions. *International Journal for Numerical Methods in Engineering*, 60(1):213–232, 2004.

[28] Kenji Takizawa, Yuri Bazilevs, and Tayfun E Tezduyar. Space–time and ALE–VMS techniques for patient-specific cardiovascular fluid–structure interaction modeling. *Archives of Computational Methods in Engineering*, 19(2):171–225, 2012.

[29] Charbel Farhat, Kristoffer G. van der Zee, and Philippe Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195:1973–2001, 2006.

[30] U. Küttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43:61–72, 2008.

[31] Miriam Mehl, Markus Brenk, Hans-Joachim Bungartz, Klaus Daubner, Ioan Lucian Muntean, and Tobias Neckel. An Eulerian approach for partitioned fluid–structure simulations on Cartesian grids. *Computational Mechanics*, 43(1):115–124, 2008.

[32] Miriam Mehl, Benjamin Uekermann, Hester Bijl, David Blom, Bernhard Gatzhammer, and Alexander Van Zuijlen. Parallel coupling numerics for partitioned fluid–structure interaction simulations. *Computers & Mathematics with Applications*, 71(4):869–891, 2016.

[33] Joris Degroote. Partitioned simulation of fluid-structure interaction. *Archives of Computational Methods in Engineering*, 20:185–238, 2013.

[34] Gene Hou, Jin Wang, and Anita Layton. Numerical methods for fluid-structure interaction - A review. *Communications in Computational Physics*, 12(2):337–377, 2012.

[35] Patrick Le Tallec and Jean Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3039–3067, 2001.

[36] Jean Frédéric Gerbeau and Marina Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:631–647, 2003.

[37] E Kuhl, S Hulshoff, and R De Borst. An arbitrary lagrangian eulerian finite-element approach for fluid–structure interaction phenomena. *International Journal for Numerical Methods in Engineering*, 57(1):117–142, 2003.

[38] Fabio Nobile and Christian Vergara. An effective fluid-structure interaction formulation for vascular dynamics by generalized robin conditions. *SIAM Journal on Scientific Computing*, 30(2):731–763, 2008.

[39] Santiago Badia, Fabio Nobile, and Christian Vergara. Fluid-structure partitioned procedures based on Robin transmission conditions. *Journal of Computational Physics*, 227(14):7027–7051, 2008.

[40] Miguel A Fernández, Jimmy Mullaert, and Marina Vidrascu. Explicit robin–neumann schemes for the coupling of incompressible fluids with thin-walled structures. *Computer Methods in Applied Mechanics and Engineering*, 267:566–593, 2013.

[41] M Lesoinne and C Farhat. Higher-order subiteration-free staggered algorithm for nonlinear transient aeroelastic problems. *AIAA Journal*, 36(9):1754–1757, 1998.

[42] C Michler, E H Van Brummelen, and R De Borst. An interface Newton-Krylov solver for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 47(10-11):1189–1195, 2005.

[43] Miguel Angel Fernández and Marwan Moubachir. A Newton method using exact Jacobians for solving fluid–structure coupling. *Computers and Structures*, 83:127–142, 2005.

[44] R Haelterman, Alfred EJ Bogaers, K Scheufele, B Uekermann, and M Mehl. Improving the performance of the partitioned QN-ILS procedure for fluid–structure interaction problems: Filtering. *Computers & Structures*, 171:9–17, 2016.

[45] Klaudius Scheufele and Miriam Mehl. Robust multisecant quasi-Newton variants for parallel fluid-structure simulations–and other multiphysics applications. *SIAM Journal on Scientific Computing*, 39(5):S404–S433, 2017.

[46] U. KuÌttler and W. A. Wall. Vector extrapolation for strong coupling fluid-structure interaction solvers. *Journal of Applied Mechanics*, 76(2):021205, 2009.

[47] Hans-Joachim Bungartz, Florian Lindner, Miriam Mehl, and Benjamin Uekermann. A plug-and-play coupling approach for parallel multi-field simulations. *Computational Mechanics*, 55(6):1119–1129, 2015.

[48] EH Van Brummelen. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *Journal of Applied mechanics*, 76(2):021206, 2009.

[49] 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*, 194:4506–4527, 2005.

[50] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196:1278–1293, 2007.

[51] Erik Burman and Miguel A. Fernández. Stabilization of explicit coupling in fluid–structure interaction involving fluid incompressibility. *Computer Methods in Applied Mechanics and Engineering*, 198:766–784, 2009.

[52] Erik Burman and Miguel A Fernández. Explicit strategies for incompressible fluid-structure interaction problems: Nitsche type mortaring versus robin–robin coupling. *International Journal for Numerical Methods in Engineering*, 97:739–758, 2014.

[53] Miguel A Fernández, Mikel Landajuela, and Marina Vidrascu. Fully decoupled time-marching schemes for incompressible fluid/thin-walled structure interaction. *Journal of Computational Physics*, 297:156–181, 2015.

[54] Oyekola Oyekole, Catalin Trenchea, and Martina BukacÌ. A second-order in time approximation of fluid-structure interaction problem. *SIAM Journal on Numerical Analysis*, 56(1):590–613, 2018.

[55] 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. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.

[56] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22:745–762, 1968.

[57] M Breuer and M Münsch. Fluid-structure interaction using LES-A partitioned coupled predictor-corrector scheme. *PAMM*, 8(1):10515–10516, 2008.

[58] M. Breuer, G. De Nayer, M. Münsch, T. Gallinger, and R. Wüchner. Fluid-structure interaction using a partitioned semi-implicit predictor-corrector coupling scheme for the application of large-eddy simulation. *Journal of Fluids and Structures*, 29:107–130, 2012.

[59] M. Astorino, F. Chouly, and M. A. Fernández. Robin based semi-implicit coupling in fluid-structure interaction: Stability analysis and numerics. *SIAM Journal on Scientific Computing*, 31(6):4041–4065, 2009.

[60] Jón Tómas Grétarsson, Nipun Kwatra, and Ronald Fedkiw. Numerically stable fluid–structure interactions between compressible flow and solid structures. *Journal of Computational Physics*, 230(8):3062–3084, 2011.

[61] Tao He. A cbs-based partitioned semi-implicit coupling algorithm for fluid–structure interaction using mcibc method. *Computer Methods in Applied Mechanics and Engineering*, 298:252–278, 2016.

[62] T. He, K. Zhang, and T. Wang. Ac-cbs-based partitioned semi-implicit coupling algorithm for fluid-structure interaction using stabilized second-order pressure scheme. *Communications in Computational Physics*, 21(5):1449–1474, 2017.

[63] Tao He, Jian Yang, and Charalampos Baniotopoulos. Improving the cbs-based partitioned semi-implicit coupling algorithm for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 87(9):463–486, 2018.

[64] A Quaini and A Quarteroni. A semi-implicit approach for fluid-structure interaction based on an algebraic fractional step method. *Mathematical models and methods in Applied Sciences*, 17(6):957–983, 2007.

[65] Santiago Badia, Annalisa Quaini, and Alfio Quarteroni. Splitting methods based on algebraic factorization for fluid-structure interaction. *SIAM Journal on Scientific Computing*, 30(4):1778–1805, 2008.

[66] Alfio Quarteroni, Fausto Saleri, and Alessandro Veneziani. Factorization methods for the numerical approximation of navier–stokes equations. *Computer methods in applied mechanics and engineering*, 188(1-3):505–526, 2000.

[67] Paola Gervasio, Fausto Saleri, and Alessandro Veneziani. Algebraic fractional-step schemes with spectral methods for the incompressible navier–stokes equations. *Journal of Computational Physics*, 214(1):347–365, 2006.

[68] Soyibou Sy and Cornel Marius Murea. A stable time advancing scheme for solving fluid–structure interaction problem at small structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 198(2):210–222, 2008.

[69] Fabio Nobile, Matteo Pozzoli, and Christian Vergara. Time accurate partitioned algorithms for the solution of fluid–structure interaction problems in haemodynamics. *Computers & Fluids*, 86:470–482, 2013.

[70] Jie Liu, Rajeev K. Jaiman, and Pardha S. Gurugubelli. A stable second-order scheme for fluid–structure interaction with strong added-mass effects. *Journal of Computational Physics*, 270:687–710, 2014.

[71] TermoFluids CFD software, year = 2019, note = www.termofluids.com.

[72] Ivette Rodriguez, Ricard Borell, Oriol Lehmkuhl, Carlos D. Perez Segarra, and Assensi Oliva. Direct numerical simulation of the flow over a sphere at Re = 3700. *Journal of Fluid Mechanics*, 679:263–287, 2011.

[73] I. RodrÃguez, O. Lehmkuhl, J. Chiva, R. Borrell, and A. Oliva. On the flow past a circular cylinder from critical to super-critical reynolds numbers: Wake topology and vortex shedding. *International Journal of Heat and Fluid Flow*, 55:91 – 103, 2015.

[74] Néstor Balcázar, Lluís Jofre, Oriol Lehmkuhl, Jesús Castro, and Joaquim Rigola. A finite-volume/level-set method for simulating two-phase flows on unstructured grids. *International journal of multiphase flow*, 64:55–72, 2014.

[75] E Gutiérrez, F Favre, Nestor Balcazar, Ahmad Amani, and J Rigola. Numerical approach to study bubbles and drops evolving through complex geometries by using a level set–moving mesh–immersed boundary method. *Chemical Engineering Journal*, 349:662–682, 2018.

[76] PA Galione, O Lehmkuhl, J Rigola, and A Oliva. Fixed-grid numerical modeling of melting and solidification using variable thermo-physical properties–application to the melting of n-octadecane inside a spherical capsule. *International Journal of Heat and Mass Transfer*, 86:721–743, 2015.

[77] Eduard Bartrons, Carles Oliet, Enrique Gutierrez, Alireza Naseri, and Carlos David Pérez-Segarra. A finite volume method to solve the frost growth using dynamic meshes. *International Journal of Heat and Mass Transfer*, 124:615–628, 2018.

[78] Guillem Colomer, Rick Borrell, F Xavier Trias, and I Rodríguez. Parallel algorithms for Sn transport sweeps on unstructured meshes. *Journal of Computational Physics*, 232(1):118–135, 2013.

[79] Lluís Jofre, Ricard Borrell, Oriol Lehmkuhl, and Assensi Oliva. Parallel load balancing strategy for volume-of-fluid methods on 3-d unstructured meshes. *Journal of Computational Physics*, 282:269–288, 2015.

[80] R Borrell, J Chiva, O Lehmkuhl, G Oyarzun, I Rodríguez, and A Oliva. Optimising the Termofluids CFD code for petascale simulations. *International Journal of Computational Fluid Dynamics*, 30(6):425–430, 2016.

2

An efficient and robust semi-implicit method

Main content of this chapter has been published in:

A. Naseri, O. Lehmkuhl, I. Gonzalez, E. Bartrons, C. D. Pérez-Segarra, and A. Oliva. A semi-implicit coupling technique for fluid–structure interaction problems with strong added-mass effect. *Journal of Fluids and Structures*, 80:94–112, 2018.

Abstract. This chapter is concerned with the numerical simulation of fluid–structure interaction problems involving an incompressible viscous flow and an elastic structure. A semi-implicit coupling technique is presented which strongly couples the added-mass term of the fluid (pressure stress) to the structure, while the remaining terms are only loosely coupled. A thorough numerical analysis is carried out to verify the accuracy of the proposed method by comparing its results to experimental data and other numerical results from the literature. The performance and accuracy of the proposed method are also compared against a fully implicit coupling technique. Numerical tests show that semi-implicit coupling significantly reduces the computational cost of the simulations without undermining either the stability or the accuracy of the results. The question of implicit or explicit coupling of the dynamic mesh step is addressed by evaluating its effect on the overall accuracy and performance of the semi-implicit method. The implicit coupling of the dynamic mesh step is found to slightly improve the accuracy, while significantly increasing the computational cost. Moreover a comparison is made on the performance of the semi-implicit method with different interface solvers.

2.1 Background

Broadly, two different approaches could be used to solve FSI problems, called monolithic and partitioned methods. In monolithic approach one uses a single solver to solve fluid and structural governing equations simultaneously. As the equations are solved together, the interaction between the domains is inherently taken into account. The main advantage of the monolithic approach is the elimination of the need for any further coupling technique at the fluid-structure interface, which reduces the complexity of the problem. However, this approach requires using the same numerical methods to discretize and solve the fluid and structural equations, while they are different in nature and have their own considerations. This may cause monolithic methods to be less efficient or reliable in some applications [1]. Another disadvantage of the monolithic approach is its inability to exploit the already-developed fluid and structural solvers. Therefore, it requires a large software development effort and usually results in a less modular solver [1,2].

Partitioned methods, on the other hand, use separate solvers for fluid and structural equations and adopt a coupling scheme to account for the interaction of the domains. The coupling scheme determines the order and frequency in which the fluid and structural equations should be solved. It also determines the manner of communication and information exchange between the two solvers which is essentially restricted to the fluid-structure interface. Partitioned approach alleviates both disadvantages of the monolithic schemes. It allows using the most adapted numerical methods for each sub-problem. These methods are previously tested and verified on diverse cases which greatly increases the reliability of the FSI simulations. It also enables the use of the previously developed solvers for fluid and structural equations which saves a large development effort and increases modularity of the software. However, partitioned approach introduces a new challenge to the problem, i.e. the coupling between the two solvers [1,2].

Partitioned methods are further divided into explicit (or loosely coupled) and implicit (or strongly coupled) schemes. In an explicit coupling method, the fluid and structural equations are solved in sequence and only once at every time step. Consequently, explicit methods do not satisfy the exact coupling condition at the fluid-structure interface. The most basic explicit scheme is the conventional serial staggered method [3]. Implicit methods, in contrast, enforce the equilibrium condition at the interface by means of coupling iterations between the fluid and structural solvers at each time step. Fixed-point (Gauss-Seidel or Jacobi) iterations [1,4] and Newton-based methods [5–7] are the most commonly used techniques to carry out the FSI coupling iterations. Vector extrapolation methods have also been used for this purpose [8].

Explicit methods work well for aeroelastic simulations and problems involving compressible flows [9,10]. However they are unstable for a wide range of problems, especially ones with incompressible flow and low solid/fluid density ratios (values

close to one). The instability is regardless of the time step size or discretization schemes for each domain. It is inherent to the coupling method and is often called "added-mass effect". The instability rises due to the fact that fluid forces in the explicit coupling depend upon a predicted displacement of the structure, rather than the correct one. As the structure moves, it has to accelerate the bulk of the fluid around it as well. Thus, part of the fluid acts as an extra mass in the structural dynamics system–given rise to the name added-mass effect. This effect is particularly strong when densities of the fluid and the structure are similar. For any loosely coupled method there is a density ratio limit that the method would suffer instability beyond it [11, 12]. While added-mass effect causes instability in the loosely coupled schemes, it deteriorates convergence of the strongly coupled methods. Thus, a FSI problem with strong added-mass effect is also challenging for implicit methods, as it requires many coupling iterations to converge at each time step [11, 12].

Implicit methods provide stable solution for FSI problems with strong added-mass effect, of which explicit methods are incapable. However, performing several coupling iterations, i.e. solving the complete system of governing equations several times per time step, requires significantly higher computational resources. To alleviate this, Fernandez et al. [13] proposed a semi-implicit coupling technique in which they used a projection method to solve the fluid equations and only implicitly coupled the projection step to the structure. Therefore the pressure stress term of the fluid is strongly coupled to the structure. It is argued that the pressure stress term is the main contributor to the added-mass effect and coupling this term explicitly will cause numerical instability [11]. By implicit treatment of the added-mass term (pressure stress), the semi-implicit method maintains the favorable stability of the implicit schemes, while explicit treatment of the other terms helps avoiding excessive computational cost [13]. A very similar method was also proposed by Breuer et al. [14, 15] to solve FSI problems with turbulent flow. An analogous idea is present in the hybrid monolithic-partitioned method of Grétarsson et al. [16] for FSI problems with compressible flow. It strongly couples the fluid pressure and solid velocity by solving them implicitly in a monolithic manner, while the remaining terms are loosely coupled in a partitioned manner. Other semi-implicit methods are also reported in the literature which share the same basic idea, e.g. [17, 18].

Despite receiving attention from researchers, semi-implicit coupling technique is far from perfect. Many of the reported methods in the literature lack modularity and simplicity. Moreover most of the reported methods are only tested in a specific type of FSI problems and their robustness in dealing with different types of FSI problems is not evaluated. Besides, there are many unaddressed questions concerning different aspects of the semi-implicit coupling methods that require more work and attention. Some semi-implicit methods in the literature implicitly couple the dynamic mesh step of the fluid [14, 15], while others only explicitly couple it [13, 17, 18]. However, to

the best of our knowledge, there has been no study that evaluates the effect of this modification on the overall performance and accuracy of the semi-implicit coupling method.

In this work, we follow a semi-implicit approach to develop an efficient coupling technique for FSI problems with strong added-mass effect. We also try to address some of the open questions concerning semi-implicit methods. The main improvements and advantages of the proposed method are as following.

- It is simple, modular and matrix-free. The method is developed with the least possible mathematical and numerical complications. This results in a methodology which is easy to follow and simple to implement.
- It is robust. The method remains stable, accurate and efficient for a wide range of FSI problems, including ones with very large deformations. It is analyzed in three widely different test cases and has shown adequate stability and performance.
- It is computationally efficient. The proposed coupling technique along with the choices for numerical schemes and discretization, result in a fast and efficient overall methodology.
- It is suitable for simulating FSI problems with turbulent flow. The fluid solver and the coupling method are developed while special attention is paid to the particular considerations of turbulent flows.

A thorough analysis is carried out to verify the accuracy of the proposed method by comparing its results to experimental data as well as other numerical results from the literature. Numerical test cases are chosen to be very distinct in order to demonstrate the robustness of the method. Three test cases feature an internal flow contained by a deformable membrane, an external flow over a blunt body with rigid-body motion, and a cavity flow with a flexible bottom. Although the proposed method has been tested in turbulent FSI problems [19,20], the attention is kept on laminar test cases, where the characteristics of the FSI method could be better highlighted.

The accuracy and computational cost of the method is compared against a fully implicit coupling technique. Moreover the effect of implicit or explicit coupling of the dynamic mesh step on the overall performance and accuracy of the semi-implicit method is evaluated. A modified version of the proposed method with implicitly coupled dynamic mesh step is also used for the numerical tests and its accuracy and performance are studied. Besides, both fixed-point and Newton-Krylov methods are used to solve the coupling interface problem and a comparison is made on their performance. Thus the main contributions of this chapter could be highlighted as:

1. A simple, efficient and robust semi-implicit coupling method is proposed and its accuracy and good performance demonstrated through numerical tests.
2. The effect of implicit or explicit coupling of the dynamic mesh step on overall performance and accuracy of the semi-implicit scheme is evaluated.
3. Performance of fixed-point and Newton-Krylov interface solvers for semi-implicit coupling method is studied.

The rest of this chapter is organized as follows. In section 2, the governing equations for each sub-domain as well as the coupling conditions on the interface are presented. The discretization methods and numerical schemes are also described in this section. In section 3, the proposed semi-implicit coupling technique is elaborated. Section 4 deals with the description of the methods used to solve the resulting interface problem. Results of the numerical tests and comparisons are provided in section 4, while section 5 summarizes and concludes the article.

2.2 Governing equations and numerical methods

In this section, the governing equations for each sub-problem domain and the coupling conditions on the interface are presented. The fluid and structural domains are referred to as $\Omega_f(t)$ and $\Omega_s(t)$ respectively, as they both vary in time. The interface of the domains is denoted by $\Gamma(t) = \Omega_f(t) \cap \Omega_s(t)$. An Arbitrary Lagrangian-Eulerian (ALE) formulation together with a conforming mesh technique [21, 22] is used to solve the fluid flow in a moving domain. A Lagrangian formulation is used for the structural equations.

2.2.1 Fluid equations

The unsteady flow of an incompressible viscous fluid is governed by the Navier-Stokes equations. An ALE formulation of these equations in a moving domain is given by

$$\nabla \cdot \mathbf{u} = \mathbf{0} \tag{2.1}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{u} = \frac{1}{\rho_f} \nabla \cdot \boldsymbol{\sigma}_f \tag{2.2}$$

where $\mathbf{u}$ is the fluid velocity and ρ_f the fluid density. Vector $\mathbf{c}$ is the ALE convective velocity $\mathbf{c} = \mathbf{u} - \mathbf{w}$, which is the fluid velocity relative to a domain moving with a velocity $\mathbf{w}$.

The stress tensor $\boldsymbol{\sigma}_f$ is defined for a Newtonian fluid as

$$\boldsymbol{\sigma}_f = -p\mathbf{I} + 2\mu_f\boldsymbol{\gamma} \tag{2.3}$$

where p is the fluid pressure, $\mathbf{I}$ the unit tensor, μ_f the dynamic viscosity of the fluid and $\boldsymbol{\gamma}$ the strain rate tensor given by

$$\boldsymbol{\gamma} = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T) \tag{2.4}$$

2.2.2 Structural equations

The structural domain is governed by the nonlinear elastodynamics equation

$$\rho_s \frac{D^2\mathbf{d}}{Dt^2} = \nabla \cdot \boldsymbol{\sigma}_s \tag{2.5}$$

where $\mathbf{d}$ stands for the structural position with respect to the reference configuration, and the structural density is shown by ρ_s. The Cauchy stress tensor $\boldsymbol{\sigma}_s$ is related to the second Piola-Kirchhoff tensor $\mathbf{S}_s$ by

$$\mathbf{S}_s = J\mathbf{F}^{-1}\boldsymbol{\sigma}_s\mathbf{F}^T \tag{2.6}$$

where $\mathbf{F}$ is the deformation gradient $\mathbf{F} = \nabla\mathbf{d}$ and J is its determinant ($J = det(\mathbf{F})$).

The FSI coupling method is presented for a generic structural system at its full extent, however, simpler structural models are used for the numerical tests. The structural equations for each test case are explained in section 2.5.

2.2.3 Coupling conditions

The coupling conditions apply at the interface Γ and account for the interaction of the domains. They are derived from the kinematic and dynamic equilibrium between the domains, which yield to the following conditions on a non-slip type interface

$$\mathbf{u}_\Gamma = \frac{\partial\mathbf{d}_\Gamma}{\partial t} \tag{2.7}$$

$$\boldsymbol{\sigma}_s \cdot \mathbf{n}_\Gamma = \boldsymbol{\sigma}_f \cdot \mathbf{n}_\Gamma \tag{2.8}$$

for any point $\mathbf{x} \in \Gamma$, where $\mathbf{n}_\Gamma$ is the unit normal vector on the interface. Equation 2.7 represents equality of the velocity of the fluid and the structure on the interface to assure the kinematic equilibrium. Equation 2.8 represents equality of the traction on the interface for dynamic equilibrium.

2.2.4 Discretization and numerical methods

For fluid flow, a fractional-step projection method [23] along with an explicit time advancement is used to solve the velocity-pressure coupling of the momentum equation. This leads to a three step solution of the fluid governing equations from time step n to $n+1$, with a time increment of Δt

$$\mathbf{u}^{p} = \mathbf{u}^{n} - \Delta t[(\mathbf{u}^{n} - \mathbf{w}^{n+1}).\nabla \mathbf{u}^{n} - \frac{\mu_f}{\rho_f}\Delta \mathbf{u}^{n}] \tag{2.9}$$

$$\frac{\Delta t}{\rho_{\mathrm{f}}}\Delta p^{\mathrm{n+1}} = \nabla.\mathbf{u}^{\mathrm{p}} \tag{2.10}$$

$$\mathbf{u}^{\mathrm{n+1}} = \mathbf{u}^{\mathrm{p}} - \frac{\Delta t}{\rho_{\mathrm{f}}}\nabla p^{\mathrm{n+1}} \tag{2.11}$$

for $\mathbf{x} \in \Omega_f^{n+1}$. For the sake of having a simple notation, a first-order Euler explicit time scheme is used for equation 2.9, but an extension to higher order schemes is straightforward. The method begins with evaluation of a predicted velocity, $\mathbf{u}^{\mathrm{p}}$, without considering the pressure gradient term (equation 2.9). A pressure field is then evaluated by solving a Poisson's equation (equation 2.10) that enforces the incompressibility condition at the velocity correction step (equation 2.11).

In this work, the fractional-step method is used not only for solving the fluid equations, but also as a framework for the overall FSI solution algorithm, making it fundamental to the proposed FSI coupling method.

A finite-volume method is used for the spatial discretization of the fluid equations on a collocated, unstructured mesh with second-order symmetry-preserving schemes. Symmetry-preserving schemes conserve the kinetic energy of the flow in discrete level [24]. Conservation of kinetic energy is extremely important while dealing with turbulent flows [25]. A conjugate gradient solver with a diagonal preconditioner is used to solve the Poisson's equation. A modern review and comparative study of advanced methods for solution of the Poisson's equation can be found in [26]. More details on the in-house flow solver code and the numerical methods can be found in [25, 27]. Integral form of the governing equations and the spatial discretization are described in Appendix A.

Structural equations are discretized in space using a finite-volume method along with a second-order central difference scheme. A second-order temporal scheme is used to discretize the second time derivative. More information on the numerical methods for the structural equations is provided in section 2.5 for each test case.

2.2.5 Mesh movement technique

Before equations 2.9-2.11 could be solved, the new fluid mesh in Ω_f^{n+1} and the surface velocities $\mathbf{w}^{n+1}$ are needed. A parallel moving mesh technique, based on radial basis function interpolation method [28], is used to move the fluid grid in accordance to the new location of the interface and update the discretized fluid domain.

The method uses values of scattered data, i.e. the known displacements of the nodes on the FSI interface, to evaluate an interpolated value in a cloud of points, i.e. the interior vertices of the fluid grid. Therefore, it does not need the connectivity of the mesh elements and can be applied to both structured and unstructured grids. The interpolated displacement $\delta\mathbf{r}$ at a point $\mathbf{x}$ is evaluated by:

$$\delta\mathbf{r}(\mathbf{x}) = \sum_{i=1}^{nv} \gamma_i \varphi(||\mathbf{x} - \mathbf{x}_i||) + \mathbf{h}(\mathbf{x}) \tag{2.12}$$

where nv is the number of nodes on the FSI interface with known displacement and φ indicates the radial basis function. The radial basis function has been chosen to be the Wendland C^2 [29] since it preserves good quality of the dynamic mesh and allows to ignore the polynomial terms of the equation 2.12, $\mathbf{h}(\mathbf{x})$. The coefficients γ_i are determined by imposing the known solution on the interface

$$\delta\mathbf{r}(\mathbf{x}_i) = \delta\mathbf{d}(\mathbf{x}_i) \qquad i = 1, 2, ..., nv \tag{2.13}$$

for $\mathbf{x}_i \in \Gamma$, thus restricting the size of the system of equations to the number of known points nv.

Surface velocities are evaluated according to the so-called space conservation law (SCL). SCL states that the sum of the volumes swept by the surfaces of a control volume must be equal to the time rate of change of its volume v

$$\frac{\partial v}{\partial t} - \int_s \mathbf{w} \cdot \mathrm{d}\mathbf{A} = 0 \tag{2.14}$$

where s is the boundary of the control volume and $\mathbf{A}$ is the area vector pointing outward. For the discretized equations to be conservative in time, the surface velocities should satisfy SCL which guarantees no volume is lost while moving the grid. To satisfy the space conservation law exactly, surface velocities are evaluated by the volume swept by each surface $\mathbf{w}_{face} = \frac{\delta v}{A\Delta t}\mathbf{n}$ where A is the surface area, $\mathbf{n}$ surface normal vector, Δt time step and δv the volume swept by the face (see Figure 2.1).

The process of moving the fluid mesh and evaluating the surface velocities at a new time step would be concisely denoted by the function $\mathbf{M}$ in the following sections

$$(\Omega_f^{n+1}, \mathbf{w}^{n+1}) = \mathbf{M}(\mathbf{d}_\Gamma^{n+1}) \tag{2.15}$$

More details concerning the mesh movement technique can be found in [28].

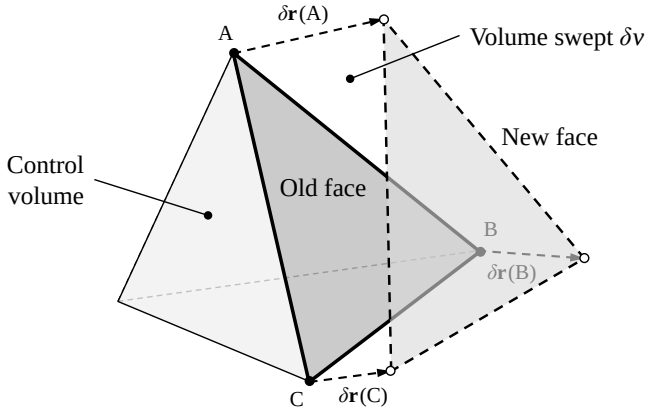

Figure 2.1: Volume swept by each face of an arbitrarily shaped polyhedral.

2.3 FSI coupling technique

A Dirichlet-Neumann (DN) decomposition of domains is used to solve the coupled FSI problem. In DN decomposition, fluid equations are solved for a known location of the interface and kinematic equilibrium (equation 2.7) is used as a Dirichlet boundary condition for fluid flow. Structural equations are solved for a known traction on the interface and are subject to Neumann boundary condition derived from dynamic equilibrium (equation 2.8). Thus the discrete fluid and structural equations can be represented as interface functions **F** and **S** so that

$$\boldsymbol{\sigma}_\Gamma = \mathbf{F}(\mathbf{d}_\Gamma) \tag{2.16}$$

$$\mathbf{d}_\Gamma = \mathbf{S}(\boldsymbol{\sigma}_\Gamma) \tag{2.17}$$

The discrete fluid function **F** includes equation 2.15 (mesh movement step) and equations 2.9-2.11 (Navier-Stokes equations). Given the current location of the interface, it moves the fluid mesh and solves the governing equations to obtain the fluid velocity and pressure fields. In particular, the fluid velocity and pressure on the interface are used to evaluate the fluid traction $\boldsymbol{\sigma}_\Gamma = \boldsymbol{\sigma}_f(p, \mathbf{u})_\Gamma \cdot \mathbf{n}_\Gamma$. The evaluated traction is then transferred to the structural function **S**, which solves the governing equations of the structure to obtain the new location of the interface.

Therefore the discrete FSI equations can be represented as an interface problem of the form

$$\mathbf{S} \circ \mathbf{F}(\mathbf{d}_\Gamma) = \mathbf{d}_\Gamma \tag{2.18}$$

with vector $\mathbf{d}_\Gamma$ and functions $\mathbf{F}$ and $\mathbf{S}$, all in the same time step.

In an implicit FSI coupling method, equation 2.18 is solved iteratively at each time step. These methods show adequate stability but are computationally expensive because they require solving the fluid and structural equations several times at every time step. In most of applications, the fluid solver takes considerably more computational effort than the structural solver.

In this study we follow a semi-implicit approach in which only the pressure stress term of the fluid equations is implicitly coupled to the structure. Pressure stress term is the main contributor to the added-mass effect and must be coupled implicitly to avoid numerical instability [11, 13]. Using a fractional step method for fluid equations allows us to split the pressure stress term of the fluid (equation 2.10) and couple it implicitly to the structure. On the other hand, the rest of the fluid equations (Eqs. 2.15, 2.9 and 2.11) are only explicitly coupled. The complete algorithm of solving the FSI problem at time step $n+1$ is as follows.

Semi-implicit FSI coupling method:
step 0: extrapolation of $\mathbf{d}_\Gamma$ from previous time steps:

$$\widetilde{\mathbf{d}}_\Gamma^{n+1} = 2.5\mathbf{d}_\Gamma^{n} - 2\mathbf{d}_\Gamma^{n-1} + 0.5\mathbf{d}_\Gamma^{n-2} \tag{2.19}$$

step 1: moving the fluid mesh (*explicitly coupled*):

$$(\Omega_f^{n+1}, \mathbf{w}^{n+1}) = \mathbf{M}(\widetilde{\mathbf{d}}_\Gamma^{n+1}) \tag{2.20}$$

step 2: ALE convection-diffusion equation (*explicitly coupled*):

$$\mathbf{u}^p = \mathbf{u}^n - \Delta t[(\mathbf{u}^n - \mathbf{w}^{n+1}).\nabla\mathbf{u}^n - \frac{\mu_f}{\rho_f}\Delta\mathbf{u}^n] \qquad \text{in } \Omega_f^{n+1} \tag{2.21}$$

step 3: pressure equation and structural solver (*implicitly coupled, solved iteratively*):

$$\mathbf{u}_\Gamma^p = \frac{{\mathbf{d}_\Gamma}^{n+1} - \mathbf{d}_\Gamma^n}{\Delta t} \qquad \text{on } \Gamma^{n+1} \tag{2.22}$$

$$\frac{\Delta t}{\rho_f}\Delta p^{n+1} = \nabla\cdot\mathbf{u}^p \qquad \text{in } \Omega_f^{n+1} \tag{2.23}$$

$${\boldsymbol{\sigma}_\Gamma}^{n+1} = \boldsymbol{\sigma}_f(p^{n+1}, \mathbf{u}^p)_\Gamma\cdot\mathbf{n}_\Gamma \qquad \text{on } \Gamma^{n+1} \tag{2.24}$$

$${\mathbf{d}_\Gamma}^{n+1} = \mathbf{S}({\boldsymbol{\sigma}_\Gamma}^{n+1}) \qquad \text{on } \Gamma^{n+1} \tag{2.25}$$

step 4: velocity correction (*explicitly coupled*):

$$\mathbf{u}^{\mathrm{n+1}} = \mathbf{u}^{\mathrm{p}} - \frac{\Delta t}{\rho_{\mathrm{f}}} \nabla p^{\mathrm{n+1}} \qquad \text{in } \Omega_f^{n+1} \tag{2.26}$$

$$\mathbf{u}_\Gamma^{n+1} = \frac{\mathbf{d}_\Gamma^{n+1} - \mathbf{d}_\Gamma^{n}}{\Delta t} \qquad \text{on } \Gamma^{n+1} \tag{2.27}$$

With this semi-implicit coupling approach, the FSI interface problem (equation 2.18) is modified into:

$$\mathbf{S} \circ \mathbf{f}(\mathbf{d}_\Gamma) = \mathbf{d}_\Gamma \tag{2.28}$$

or

$$\mathbf{R}(\mathbf{d}_\Gamma) = \mathbf{S} \circ \mathbf{f}(\mathbf{d}_\Gamma) - \mathbf{d}_\Gamma = 0 \tag{2.29}$$

which stands for the step 3 of the above algorithm. In the new FSI equation, instead of the complete fluid solver function **F**, only the pressure equation (denoted by **f**) is coupled to the structure implicitly. Again, $\mathbf{d}_\Gamma$ and the functions **f** and **S** are in the same time step.

The proposed methodology is similar to the semi-implicit methods at [13, 15, 17, 18] in keeping the ALE convection-diffusion equation (steps 2) out of the FSI coupling loop. Avoiding to iterate this equation at every time step significantly reduces the computational cost of the simulations. Unlike the methods in [13,15,17,18], the velocity correction step in the current method is outside the FSI coupling loop. FSI boundary condition is applied to the intermediate predicted velocity (rather than velocity itself) during the coupling iterations (equation 2.22), thus avoiding to calculate the corrected velocity field at each iteration. When the FSI convergence is reached, the final velocity field is evaluated using the final pressure field and the final boundary displacement. Avoiding to iterate the velocity correction step further reduces the computational cost of the simulations.

Moreover, by applying the boundary condition on the predicted velocity, the shear stress term on the boundary (in equation 2.24) is evaluated using the updated values of velocity. For example, in [13, 17] the shear stress term is similarly evaluated using the predicted velocity field, however in their method the predicted velocity on the boundary is evaluated once at each time step and is not updated during the coupling iterations. It means methods in [13, 17] use a constant velocity vector on the boundary to evaluate the shear stress. In the present method we update the predicted velocity on the interface at every iteration and evaluate the fluid traction using current values of velocity.

Unlike [13, 17] an explicit temporal discretization scheme is used for the ALE convection-diffusion equation (equation 2.21). Explicit time-marching schemes are mostly preferred for their simplicity and lower computational cost, especially in the case of a turbulent flow where small time steps are inevitable. Keeping the convection-diffusion step out of the FSI loop provides the opportunity to use an explicit time-marching method which ought not to be missed.

In the semi-implicit coupling method proposed by Breuer [15], the mesh movement step is implicitly coupled to the structure. Since the ALE convection-diffusion step is out of the FSI loop, updating the fluid mesh and recalculating the geometrical derivatives at each coupling iteration only reflects in the Poisson's equation for pressure. On the other hand, moving the computational grid is an expensive step and including it in the coupling iterations would increase the computational cost of the method. This leads to the question of whether it is necessary and worthy to include the dynamic mesh step in the coupling iterations. No analysis to asses the extent of necessity and effect of this modification on the accuracy of the results is provided by Breuer et al. [15] or in other published works, to the best of our knowledge. In this work we have studied the effect of implicit coupling of the dynamic mesh step on the accuracy and performance of the semi-implicit method. For this purpose a modified version of the proposed method (referred to as semi-implicit-M in the rest of the text) is also used to solve the numerical test cases. The modified version is similar to the original one, except for the mesh movement step (step 1) which is modified into $(\Omega_f^{n+1}, \mathbf{w}^{n+1}) = \mathbf{M}(\mathbf{d}_\Gamma^{n+1})$ and is repeated at every coupling iteration.

Finally, it should be noted that although the proposed coupling method requires a specific treatment of the fluid equations (using fractional-step method), it can be used with an arbitrary structural solver. Thus less attention is paid to the structural solver in this chapter. However, the total performance improvement -with respect to a fully implicit coupling method- would indeed depend on the efficiency of the structural solver too.

2.4 Interface solvers

In this section we discuss the iterative methods to solve the nonlinear interface problem, arisen from the FSI coupling (equation 2.28 or 2.29). Fixed-point (FP) and Newton-Krylov methods are two family of solvers that have been widely used for FSI problems. In this study we have used both methods and compared their performance in a semi-implicit coupling.

2.4.1 Fixed-point solver

This is a class of iterative solvers that are popular mostly for their simplicity. They are very easy to implement and have proved to be efficient and robust in many problems. Jacobi and Gauss-Seidel iterations are the most basic and popular methods.

The fixed-point form of the interface problem is equation 2.28:

$$\mathbf{S} \circ \mathbf{f}(\mathbf{d}_\Gamma) = \mathbf{d}_\Gamma$$

A block Gauss-Seidel method is used in this study with the extrapolated value of $\mathbf{d}_\Gamma$ (equation 2.19) as the initial guess. Each iteration begins with solving the coupled system of equations

$$\widehat{\mathbf{d}_{\Gamma k+1}} = \mathbf{S} \circ \mathbf{f}(\mathbf{d}_{\Gamma k}) \tag{2.30}$$

where k indicates the coupling iterations. The time step index is dropped for the sake of simplicity, as all the parameters are at the same time step. The interface residual is defined as

$$\mathbf{r}_{\Gamma k+1} = \widehat{\mathbf{d}_{\Gamma k+1}} - \mathbf{d}_{\Gamma k} \tag{2.31}$$

and the line search step to update the solution is

$$\mathbf{d}_{\Gamma k+1} = \mathbf{d}_{\Gamma k} + \omega_k \mathbf{r}_{\Gamma k+1} \tag{2.32}$$

where ω_k is the relaxation factor. Relaxation is necessary for the stability of the scheme. It has been shown in several studies that unrelaxed Gauss-Seidel method either converges very slowly or does not converge at all for FSI problems involving an incompressible flow [1, 4]. Our numerical tests found the unrelaxed method unstable for the problems in hand. The relaxation factor is evaluated using Aitken's Δ^2 method. For a vector equation, ω_k could be obtained from:

$$\omega_k = -\omega_{k-1} \frac{\mathbf{r}_{\Gamma k}^T (\mathbf{r}_{\Gamma k+1} - \mathbf{r}_{\Gamma k})}{(\mathbf{r}_{\Gamma k+1} - \mathbf{r}_{\Gamma k})^T (\mathbf{r}_{\Gamma k+1} - \mathbf{r}_{\Gamma k})} \tag{2.33}$$

with $\omega_0 = 0.5$ used in this work for the first iteration.

FSI convergence is achieved at every time step when the ℓ_2 norm of the interface residual is small enough to meet the convergence criterion:

$$\frac{||\mathbf{r}_{\Gamma k}||_2}{||\mathbf{r}_{\Gamma 0}||_2} < \epsilon \tag{2.34}$$

with a predefined tolerance of ϵ ($\epsilon = 10^{-5}$ is used in the numerical tests).

2.4.2 Newton-Krylov solver

Newton-Krylov method consists of two levels of iterative solvers. The first level is a Newton's method to linearize the problem and the second level is a Krylov subspace method to solve the resulting linear system of equations (see [30] for a review). Newton-Krylov methods normally show a better performance than fixed-point methods since the FSI problem is highly nonlinear.

The interface problem is of the form of equation 2.29:

$$\mathbf{R}(\mathbf{d}_\Gamma) = \mathbf{S} \circ \mathbf{f}(\mathbf{d}_\Gamma) - \mathbf{d}_\Gamma = 0$$

Applying Newton's method we have

$$\mathbf{R}'(\mathbf{d}_{\Gamma k})\Delta\mathbf{d}_{\Gamma k+1} = -\mathbf{R}(\mathbf{d}_{\Gamma k}) \tag{2.35}$$

$$\mathbf{d}_{\Gamma k+1} = \mathbf{d}_{\Gamma k} + \Delta\mathbf{d}_{\Gamma k+1} \tag{2.36}$$

where k is the coupling iteration index. The function $\mathbf{R}'$ denotes the interface Jacobian

$$\mathbf{R}'(\mathbf{d}_\Gamma) = \frac{d\mathbf{R}(\mathbf{d}_\Gamma)}{d\mathbf{d}_\Gamma} \tag{2.37}$$

Since the Jacobian matrix of the coupled system of fluid-structure equations is not easily accessible, a matrix-free Krylov subspace solver is used to solve the Newton's equation (equation 2.35). The advantage of the Krylov solvers is that they only need the product of the Jacobian matrix and a vector, rather than the Jacobian matrix itself. A first order Taylor series expansion is used to approximate the product of the Jacobian matrix and an arbitrary vector $\mathbf{v}$

$$\mathbf{R}'(\mathbf{d}_{\Gamma k})\mathbf{v} = \frac{\mathbf{R}(\mathbf{d}_{\Gamma k} + \delta\mathbf{v}) - \mathbf{R}(\mathbf{d}_{\Gamma k})}{\delta} \tag{2.38}$$

with δ evaluated as $\delta = \lambda(\lambda + \frac{||\mathbf{d}_{\Gamma k}||_2}{||\mathbf{R}(\mathbf{d}_{\Gamma k})||_2})$, where λ is a sufficiently small number ($\lambda = 10^{-4}$ in this work), as suggested in [4]. Other approaches to choose δ could be found in [30].

An unpreconditioned GMRES solver [31] is used in this study as the Krylov solver. GMRES is chosen for its favorable convergence and robustness, considering that the FSI problem is highly nonlinear. We have also tried the BiCGSTAB method [32] as the Krylov solver but it showed convergence problems in some of the numerical tests.

FSI convergence criterion for Newton-Krylov solver is identical to that of the fixed-point method (equation 2.34 with a tolerance $\epsilon = 10^{-5}$). However, its performance also depends on the tolerance for convergence of its inner Krylov solver ϵ_k. In this work we set a high tolerance for the Krylov solver so that Newton's equation is solved

with a rough accuracy. Therefore, Krylov solver takes less iterations to solve each Newton step, but more Newton steps are required to reach FSI convergence. We have seen that the overall efficiency of the method is improved in this arrangement, comparing to the case of an accurate Krylov solver. In order to optimize the performance of the method, ϵ_k is evaluated dynamically at every Newton step by

$$\epsilon_k = \gamma \left(\frac{||\mathbf{R}(\mathbf{d}_{\Gamma k})||_2}{||\mathbf{R}(\mathbf{d}_{\Gamma k-1})||_2} \right)^{\alpha} \tag{2.39}$$

with $\gamma = 1$ and $\alpha = (1 + \sqrt{5})/2$ as suggested in [33]. A minimum limit, $\epsilon_{k(min)} = 10^{-3}$, is set to avoid smaller values.

2.5 Numerical tests

Three numerical test cases are studied to demonstrate the accuracy, stability and the computational efficiency of the proposed methodology. The test cases are widely distinct and are chosen to demonstrate the robustness of the method in dealing with different types of FSI problems. All test cases are previously studied in the literature and represent important and practical FSI problems. The first test case is a bio-inspired FSI problem of incompressible flow inside a deformable vessel. The problem features a very low solid/fluid density ratio, which signifies a very strong added-mass effect. This test case is very challenging in terms of stability and convergence and is used here to demonstrate the stability of the method and its higher efficiency. The second test case is the vortex-induced vibration of an elastically mounted cylinder in low Reynolds number external flow. Unlike the other two test cases, the structure is not a thin-walled deformable membrane, but a rigid body moving inside the flow field. The added-mass effect is smaller for this test case but it features larger displacements. Moreover, there are available experimental data for this case that serve to validate the method. The third test case is a driven cavity with a flexible bottom. This test case features very large deformations and is used to further demonstrate the capability of the method to deal with large displacements of the solid boundary.

2.5.1 3D flow inside a deformable tube

This benchmark problem was proposed by [34] and studied, among others, by [4,6,13]. The problem is a 3D flow inside a straight tube with a deformable wall, motivated by the type of problems encountered in hemodynamics. The tube has a length of $l = 0.05$m, an inner radius of $R_0 = 0.005$m and a wall thickness of $h = 0.001$m. The fluid density and viscosity are $\rho_f = 1000\text{kg/m}^3$ and $\mu_f = 0.003\text{Pa} \cdot \text{s}$, respectively. The structural density is $\rho_s = 1200\text{kg/m}^3$, the Young modulus $E = 3 \times 10^5\text{N/m}^2$, the

Poisson ratio $\nu = 0.3$, and the Timoshenko factor is $k = 5/6$. The density ratio is very small ($\rho_s/\rho_f = 1.2$) which means there is a very strong added-mass effect present.

A simplified form of the Navier equations for vascular walls [35] is used to model the deformable structure. The model is derived from the theory of linear elasticity, considering only the radial deformation of the vessel and neglecting the shear stress terms in the structure. The governing equation of the structure reads

$$\rho_s h \frac{\partial^2 \mathbf{d}}{\partial t^2} - kGh \frac{\partial^2 \mathbf{d}}{\partial z^2} + \frac{Eh}{1-\nu^2} \frac{\mathbf{d}}{R_0^2} = \boldsymbol{\sigma}_\Gamma \tag{2.40}$$

where $\mathbf{d} = [d_1, 0, 0]^T$ and $\boldsymbol{\sigma}_\Gamma = [\sigma_1, 0, 0]^T$ in a cylindrical coordinate (r, θ, z). A detailed description of the model could be found in [35, 36].

The tube wall is considered a thin structure so a 2D grid in the cylindrical coordinate is used for the structure. The structural grid nodes match the fluid mesh on the interface so there is no need for interpolation of parameters between the domains. Structural equation is discretized in space using a finite-volume method along with a second-order central difference scheme. An implicit second-order finite difference scheme is used to discretize the second time derivative.

The tube is clamped at both ends and the fluid is initially at rest. An overpressure of 1333.2Pa is applied at the tube inlet during a period of 0.003s and a constant pressure of 0Pa afterwards. Pressure at the outlet is 0Pa during the whole simulation. Neumann boundary condition is used for velocity at both inlet and outlet boundaries. Simulations are carried out during 0.01s with a constant time step of $\Delta t = 10^{-4}s$.

Propagation of the pressure wave with a finite velocity is observed inside the tube. Figure 2.2 shows the pressure contour plots at three different time instants: t=0.0025, 0.005 and 0.0075 s. Deformation of the tube wall is magnified by a factor of 10 to be visible more clearly.

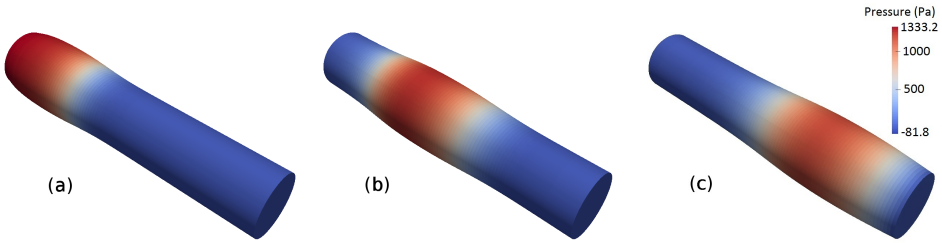

Figure 2.2: Pressure wave propagation inside the deformable vessel: (a) t=0.0025s, (b) t=0.005s and (c) t=0.0075s.

The mesh-independency of the results is assessed by using three different grids to solve the problem. Two parameters are used to evaluate the accuracy, namely the inner radius of the mid-point of the tube at $t = 0.005s$, and the outlet mass flow rate at $t = 0.009s$. The two parameters are chosen because they represent the physics of the problem. The timing is also chosen to represent the instants of interest, i.e. the instants when the peak of the wave is in the area of the target parameter. Table 2.1 contains the information of the grids and results of the simulation.

Table 2.1: Mesh-independency of the results for deformable tube test case.

Mesh name	Grid size		mid-point radius	outlet mass flow
	Fluid	Structure	t=0.005s (mm)	t=0.009s (g/s)
M1	3742	1267	5.01	24.50
M2	8776	2760	5.05	25.38
M3	19156	4450	5.05	25.49

As noticeable in Table 2.1, results obtained by mesh M2 and M3 are very similar. The M1 mesh gives good results for mid-point radius but the outlet mass flow is about 4% different. Mesh M2 is used to carry out the rest of the simulations for the sake of both accuracy and computational cost.

In order to verify the accuracy of the proposed method, the problem was also solved using a fully implicit coupling technique to generate reference results. For the implicit coupling, equation 2.18 is solved iteratively using the interface solvers described in section 2.4. It means all fluid terms and the dynamic mesh step are strongly coupled to the structure via coupling iterations. Table 2.2 represents the results obtained by the implicit, semi-implicit and semi-implicit-M coupling methods. Same parameters are used to evaluate the accuracy of the methods.

Table 2.2: Comparison of the accuracy of the implicit and semi-implicit methods for deformable tube test case.

Coupling method	mid-point radius t=0.005s (mm)	outlet mass flow t=0.009s (g/s)
Implicit	5.05	25.67
Semi-implicit	5.05	25.38
Semi-implicit-M	5.05	25.57

Results in Table 2.2 demonstrate the adequate accuracy of the proposed semi-implicit coupling method. The maximum error in mid-point radius is less than 0.1% while outlet mass flow rate has an error of 1.1% (with respect to results of the implicit

method). The level of the error is very low which suggests the semi-implicit coupling method does not degrade the accuracy of the solution, compared to a fully implicit method. Including the mesh movement step in the coupling iterations (semi-implicit-M method) further reduces the error to under 0.4%.

Moreover, Figure 2.3 compares the transient results obtained by the semi-implicit method against those of the fully implicit scheme. The picture in the left shows the radius at the mid-point of the tube while the picture in the right depicts the mass flow rate at the outlet, during the simulation time. Results of the semi-implicit method agree very well with those of implicit method at every time step which further verifies the accuracy of the proposed method.

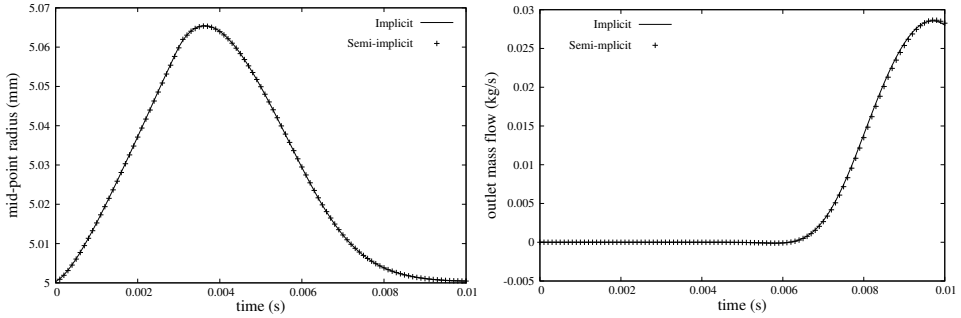

Figure 2.3: Comparison of the transient results obtained by implicit and semi-implicit coupling methods, left: inner radius of the mid-point of the tube, right: outlet mass flow rate.

Table 2.3 compares the performance of the implicit and semi-implicit coupling methods. The performance criteria are the average number of coupling iterations at each time step and the overall CPU time of the simulations. CPU times are presented in non-dimensional form. They are normalized by the smallest value, which is that of the semi-implicit method. All three coupling methods have been used with the same interface solver (FP-Aitken). Simulations were carried out on a machine with two Quad-Core AMD Opteron 2376 CPUs (8 cores in total) and 16GB of RAM. The machine was used exclusively for the solution of each case with identical conditions in order to attain comparable CPU times.

Data in Table 2.3 demonstrate the significantly lower computational cost of the proposed semi-implicit coupling method with respect to implicit coupling technique. Comparing the total CPU times of the implicit and semi-implicit methods shows that semi-implicit coupling reduces the computational time by 92%. This significant reduction in CPU time is because the semi-implicit method avoids iterating the expensive steps of the solution procedure described in section 2.3.

The required CPU time for the semi-implicit-M method is significantly higher. It

Table 2.3: Performance comparison of implicit and semi-implicit coupling methods in deformable tube test case.

Coupling technique	Average No. of coupling iterations	Normalized CPU time
Implicit	39.7	12.9
Semi-implicit	36.3	1.0
Semi-implicit-M	38.5	11.2

takes 11.2 times more computational time than the semi-implicit method, and it is only 13% faster than the implicit coupling scheme. Including the mesh movement step in the coupling iterations is the reason for such a high computational cost of the modified semi-implicit method. Clearly the relative CPU time of this method highly depends on the performance of the dynamic mesh technique. With a simple and fast mesh movement tool, its performance would be closer to the semi-implicit method. However, in many applications the quality of the mesh near the solid surfaces is of utmost importance. Hence an advanced mesh movement technique which guarantees the high quality of the mesh elements in vicinity of the moving boundary is indispensable. Such advanced mesh movement techniques (like the method used in this work) are normally very costly, which remarkably increases the computational cost of the simulations.

Based on results in Table 2.2 and 2.3, including the mesh movement step in the coupling iterations only slightly improves the accuracy of the results, while significantly increasing the computational cost. As discussed in section 2.3, updating the fluid mesh in the semi-implicit method only affects the coefficients of the Poisson's equation for pressure. We have seen in the present numerical test that the change in the pressure field due to updating the mesh, and the subsequent changes in the location of the interface and other flow parameters, are marginal. This could be partly due to the relatively small deformation of the structure and small time step size in the current test case. In the following subsections we would study this effect in FSI problems with larger displacement of the interface boundary.

It is worth to note that the semi-implicit method reduces the computational cost of each coupling iteration but does not particularly affect the number of coupling iterations required for convergence. The average number of coupling iterations for implicit and semi-implicit techniques are very similar (Table 2.3). The number of iterations to reach convergence at each time step is mostly affected by the iterative solver that is used to carry out the coupling iterations. Two basically different interface solvers were used to solve the coupled interface problem, as explained in section 2.4. The performance of these solvers are reflected in Table 2.4, using four criteria. The

first criterion is the average number of coupling iterations at each time step. The second criterion represents the number of times that the coupled equations ($\mathbf{S} \circ \mathbf{f}(\mathbf{d}_\Gamma)$) are solved at each coupling iteration. This number is essentially equal to 1 for the fixed-point solver but is higher for the Newton-Krylov method, because it undergoes an inner loop inside every coupling iteration. The third criterion is the number of times that the coupled equations are solved at each time step (the product of the first and the second criteria). The last criterion is the overall CPU time of the simulations, normalized by the smallest value which is that of the Newton-Krylov method.

Table 2.4: Performance comparison of different interface solvers for semi-implicit coupling method in deformable tube test case.

Interface solver	Average No. of coupling iterations	No. of $\mathbf{S} \circ \mathbf{f}(\mathbf{d}_\Gamma)$ solutions per coupling iteration	Average No. of $\mathbf{S} \circ \mathbf{f}(\mathbf{d}_\Gamma)$ solutions per time step	Normalized CPU time
FP-Aitken	36.3	1	36.3	1.27
Newton-Krylov	4.0	4.6 (average)	18.4	1.00

It is seen in the data of Table 2.4 that the nonlinear Newton-Krylov method outperforms the fixed-point method. It reduces the average number of solving the coupled equations by almost half with respect to FP-Aitken method (third criterion in Table 2.4). Comparing the CPU times, FP-Aitken takes 27% more computational time than the Newton-Krylov solver. It should be noted that comparing the number of coupling iterations for these two methods is not appropriate because Newton-Krylov method takes considerably more computational time per coupling iteration.

The current test case is a FSI problem with a very strong added-mass effect that takes many coupling iterations to satisfy the equilibrium condition on the interface. The reduction in computational time by using the Newton-Krylov method is tangible. It should be mentioned that although Newton-Krylov method outperforms the FP-Aitken, it introduces more complexity to the problem due to evaluation of the Jacobian. It is also worth to mention that an unpreconditioned Krylov solver has been used to solve the Newton's equation in this work. Using a preconditioner normally boosts the performance of a Krylov solver, however, designing a preconditioner in FSI applications is not straightforward due to the unavailability of the Jacobian matrix. In the next subsection we will compare the performance of both interface solvers in a FSI problem that is far less demanding in terms of required number of coupling iterations.

2.5.2 Vortex-induced vibration of a circular cylinder

In this test case we solve the external flow over an elastically mounted circular cylinder and study the vibrations induced by the flow vortices. Vortex-induced vibration (VIV) is an important class of FSI problems with a wide range of applications in aerodynamics and offshore engineering.

The cylinder is elastically mounted and oscillates due to the fluctuating fluid forces that are originated from vortex shedding phenomenon. In a certain range of Reynolds number, the vortex shedding frequency changes to match the natural structural frequency of the cylinder motion. This range of Reynolds number is called *lock-in* region since the vortex shedding no longer occurs in the Strouhal frequency, but in the natural structural frequency of the cylinder.

In order to validate the proposed method, a series of VIV simulations in low Reynolds numbers was carried out to numerically reproduce the experimental results of Anagnostopoulos and Bearman [37]. The cylinder was constrained to oscillate transversely only, as per the experiments. A sketch of the domain and problem setup is shown in Figure 2.4. The domain size is chosen based on previous experience on flow over a fixed cylinder and guidance from other VIV studies in the literature. A uniform flow with a velocity U_∞ enters the domain at the inlet boundary. Pressure is set to zero at the outlet while a Neumann boundary condition is used for the velocity. For the sake of computational efficiency, the dynamic mesh and ALE formulation is restricted to a zone of 5D distance from the center of the cylinder. The computational grid in the rest of the domain is fixed (not moving) and an Eulerian formulation is applied.

The equation of motion of the cylinder is described by

$$m\frac{\partial^2 \mathbf{d}}{\partial t^2} + c\frac{\partial \mathbf{d}}{\partial t} + k\mathbf{d} = \mathbf{q}(t) \tag{2.41}$$

where $\mathbf{d} = [0, y, 0]^T$, y being the vertical location of the center of the cylinder. The oscillation system parameters, m stands for the cylinder mass, c the damping coefficient and k is the spring stiffness. The vertical component of the time-variant fluid force (pressure and shear forces) on the cylinder is shown by $L(t)$ so that $\mathbf{q}(t) = [0, L(t), 0]^T$. The cylinder has a natural frequency of $f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}}$. Table 2.5 shows the definition of the relevant non-dimensional numbers and their values. All non-dimensional numbers equal those of the experiments [37]. The Reynolds number varies between 90 to 140 and the associated reduced velocity between 5.01 to 7.80. The variable l in the definition of mass ratio is the length of the cylinder.

To evaluate the mesh-independency of the results, three fluid grids are used to solve the problem at Re=100. Table 2.6 contains the information of the grids and compares their results. Parameters in the table are mesh size, amplitude of the

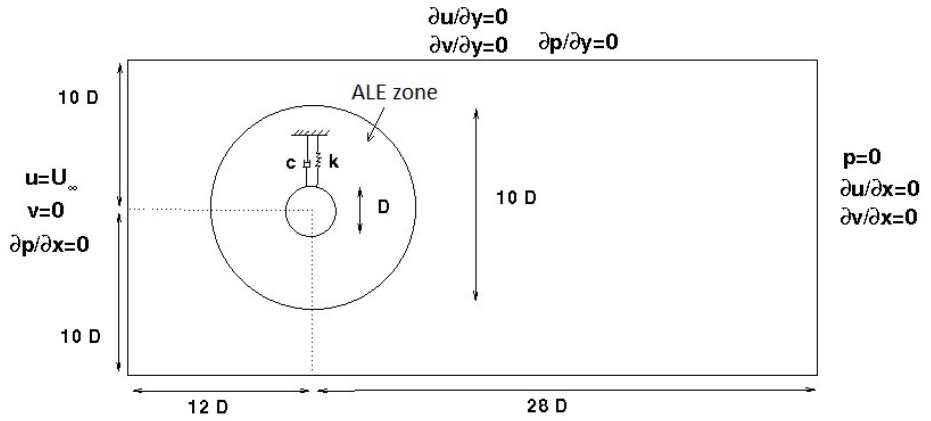

Figure 2.4: Schematic view of the domain and problem setup for the VIV test case.

Table 2.5: Relevant non-dimensional numbers of the VIV problem.

Name	Definition	Value
Reynolds (Re)	$\rho_f U_\infty D/\mu_f$	90-140
Reduced velocity (U_r)	$U_\infty/(f_n D)$	5.01-7.80
Mass ratio	$\rho_f D^2 l/2m$	0.00427
Damping ratio	$c/2\sqrt{km}$	0.0012

vibrations $A^* = y_{max}/D$, vortex shedding frequency f divided by natural frequency of the cylinder f_n, and the drag coefficient of the cylinder C_d.

Table 2.6: mesh-independency of VIV results at Re=100.

Mesh name	No. of control volumes	A^*	f/f_n	C_d
M1	7195	0.394	0.967	1.40
M2	13685	0.418	0.988	1.59
M3	27091	0.423	0.992	1.61

As results in Table 2.6 demonstrate, the grids M2 and M3 yield very similar results. Considering both accuracy and computational cost, mesh M2 is used for the rest of the simulations. Figure 2.5 shows the mesh M2 at its original condition (Figure 2.5-a) and

when the cylinder is at the peak of its oscillation (displacement of 0.42D for Re=100), in Figure 2.5-b. A closer zoom is also provided to better see the mesh elements around the cylinder. It could be seen in the figure that the mesh movement technique has preserved the quality of the mesh, particularly in the vicinity of the solid boundary.

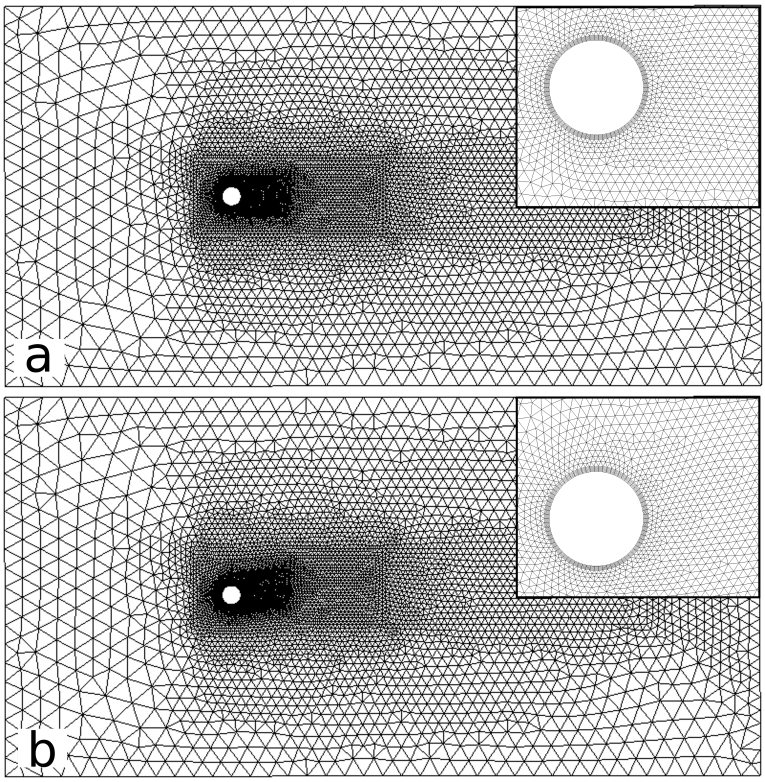

Figure 2.5: Computational grid around the cylinder, a: original mesh when y=0, b: moved mesh when y=-0.42D.

Figure 2.6 shows the results of the VIV simulations and compares them against experimental data [37] and other numerical results [18, 38, 39]. Plotted data are the amplitude of cylinder vibration and vortex shedding frequency for different Reynolds numbers.

As seen in the Figure 2.6, the method has captured the lock-in phenomenon. For Reynolds numbers $Re < 95$ and $Re > 115$ the vortex shedding occurs at the Strouhal frequency, the cylinder is *unlocked* and the amplitude of the oscillations is small. For $95 < Re < 115$ the cylinder is *locked-in* and the amplitude of the oscillations is

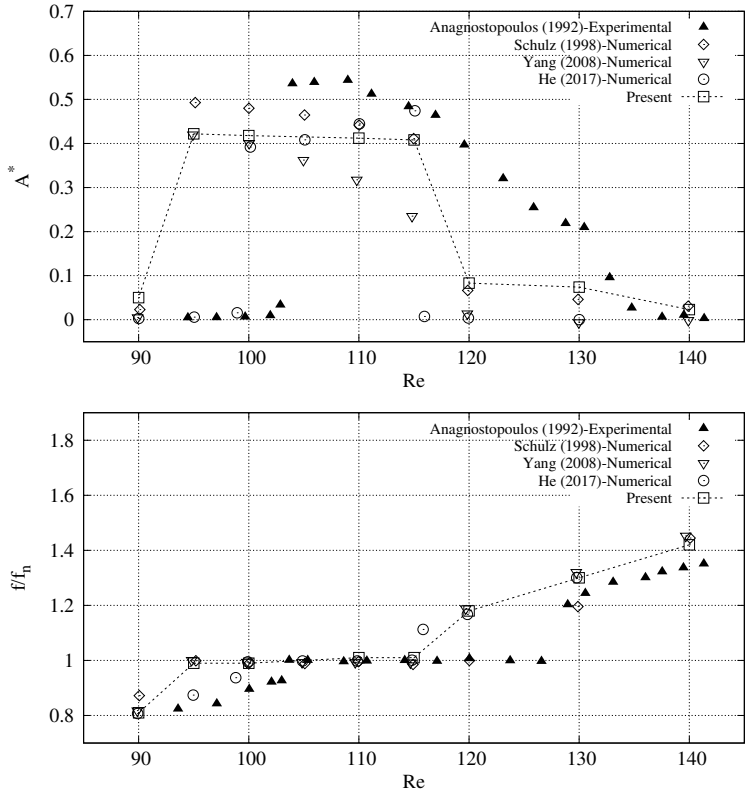

Figure 2.6: Amplitude of the cylinder vibration and vortex shedding frequency for different Reynolds numbers.

significantly larger. The vortex shedding no longer occurs at the Strouhal frequency but in the natural structural frequency of the cylinder. The present numerical results are seen to agree fairly well with the experimental and other numerical data. The large vibration amplitudes in the lock-in region and the change of vortex shedding frequency to match the natural structural frequency of the cylinder are well captured. However, the maximum displacements are smaller than the experiments but in agreement with other numerical results. Moreover there is a slight shift in the lock-in region, i. e. the beginning and end of the numerical lock-in region occur at lower Reynolds numbers than their experimental counterparts. This shift happens for other numerical results as well [18,38,39]. The discrepancies may originate from the 3-D effects in the experiments that a 2-D simulation like this work can not capture. In

the experimental study, the authors mention that no end plates were used on the cylinder [37]. Moreover the Reynolds numbers are close to the region of transition to 3-D (around Re=180 for fixed cylinder). Thus it is possible that the Reynolds number locally exceeded the transition range and some 3-D effects were introduced to the flow.

Table 2.7 represents the VIV results obtained by the implicit, semi-implicit and semi-implicit-M coupling methods at Re=100. Results demonstrate that the accuracy of the proposed semi-implicit method is comparable to a fully implicit coupling technique. The discrepancies in maximum displacement, frequency and drag coefficient obtained by the semi-implicit method (with respect to implicit scheme) are 2.1%, 0.5% and 1.8% respectively. This clearly shows the capability of the proposed method in producing accurate results. Results of the modified version of the method, the semi-implicit-M, have a difference under 1% for all three parameters with respect to the implicit method. Again we see that implicit coupling of the dynamic mesh step (semi-implicit-M method) improves the accuracy of the results, however this improvement is not significant. This observation is similar to the previous test case where a slight improvement in the accuracy of the results were observed. This suggests that even in a FSI case with large deformations, explicit coupling of the dynamic mesh step does not noticeably reduce the accuracy of the results. This observation is also examined in the next test case with much larger structural deformations.

Table 2.7: Comparison of the VIV results obtained by different coupling methods at Re=100.

Coupling method	A^*	f/f_n	C_d
Implicit	0.427	0.993	1.62
Semi-implicit	0.418	0.988	1.59
Semi-implicit-M	0.424	0.990	1.60

Table 2.8 compares the performance of the implicit and semi-implicit coupling methods for the VIV case at Re=100. The performance criteria are the average number of coupling iterations and the overall CPU time of the simulations. All three coupling methods have been used with the same interface solver (FP-Aitken).

First thing to note in Table 2.8 is the much lower average number of iterations, comparing to the previous test case. While the previous test case required nearly 40 FSI coupling iterations per time step, the current problem only needs 2 or 3 iterations to converge. Since the density ratio for the current test case is relatively large ($\rho_s/\rho_f = 149$), the added-mass effect is much smaller than the previous problem. Comparing Table 8 and Table 3 shows how strong added-mass effect makes a problem challenging in terms of stability and convergence. Because of the much smaller number of iterations, the CPU time ratio is also much smaller. The implicit method

Table 2.8: Performance comparison of implicit and semi-implicit coupling methods for VIV simulation at Re=100.

Coupling technique	Average No. of coupling iterations	Normalized CPU time
Implicit	3.2	2.9
Semi-implicit	2.1	1.0
Semi-implicit-M	3.0	2.6

takes 2.9 times more CPU time (comparing to 12.9 times in the previous test case). Nevertheless, the superior performance of the semi-implicit method is still remarkable. It reduces the computational time by 65% with respect to the implicit method, while introducing a maximum of 2% error to the results. Again we see that the semi-implicit-M method noticeably increases the CPU time and its performance is not significantly better than the implicit scheme. Based on both test cases, we can conclude that implicit coupling of the dynamic mesh step only slightly improves the accuracy while significantly increasing the computational time.

Performance of different interface solvers for semi-implicit coupling method are compared in Table 2.9, using four criteria similar to the previous test case.

Table 2.9: Performance comparison of different interface solvers for semi-implicit coupling method for VIV case at Re=100.

Interface solver	Average No. of coupling iterations	No. of $\mathbf{S} \circ \mathbf{f}(\mathbf{d}_\Gamma)$ solutions per coupling iteration	Average No. of $\mathbf{S} \circ \mathbf{f}(\mathbf{d}_\Gamma)$ solutions per time step	Normalized CPU time
FP-Aitken	2.1	1	2.1	1.02
Newton-Krylov	1.4	1.1 (average)	1.5	1.00

As results in Table 2.9 show, the performance of the interface solvers are practically identical. The fixed-point method takes 2% more computational time which is negligible. This could be understood based on the low number of coupling iterations required for convergence in this test case. For such low number of iterations, a more advanced nonlinear interface solver (the Newton-Krylov method) does not have a practical advantage over a simpler fixed-point solver. Considering the simplicity of the FP-Aitken method, it is actually a better candidate for this class of FSI problems.

2.5.3 Driven cavity with a flexible bottom

A third test case is solved to further demonstrate the capability of the proposed method to stably solve FSI problems with large deformation of the solid boundary. The test case is a 2-D lid-driven cavity with a flexible bottom, as studied in [4,8,12]. The cavity is of a $1m \times 1m$ dimension. The top boundary of the cavity is moving with an oscillatory speed of $u(t) = 1 - cos(\omega t)$ with $\omega = 2\pi/5$. There are two openings of $0.1m$ length on the sidewalls that allow the fluid to enter to and exit from the domain. The openings are devised to make sure the incompressibility of the fluid does not constrain the structural deformation. Figure 2.7 shows a schematic description of the problem. The fluid density and viscosity are $\rho_f = 1.0kg/m^3$ and $\mu_f = 0.01Pa.s$, respectively. The flexible structure at the bottom has a thickness of $h = 0.05m$ and Young modulus $E = 250N/m^2$. The problem is solved for two different structural densities of $\rho_s = 50$ and $5kg/m^3$.

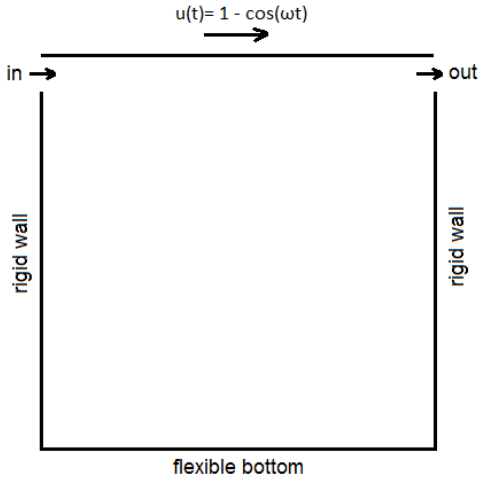

Figure 2.7: Schematic view of the problem setup for driven cavity with flexible bottom.

The flexible bottom is modeled as an Euler-Bernoulli beam, governed by the following equation:

$$\rho_s A \frac{\partial^2 \mathbf{d}}{\partial t^2} + EI \frac{\partial^4 \mathbf{d}}{\partial x^4} = q(x,t) \tag{2.42}$$

where $\mathbf{d} = [0, y, 0]^T$ in a Cartesian coordinate (x,y,z), A is the cross section area of the beam, I the second moment of area, and q is the load per unit length.

Structural governing equation is discretized in space using a finite-volume method along with a second-order central difference scheme. An implicit second-order finite difference scheme is used to discretize the second time derivative. A classical 31×31 grid is used to solve the problem. A 100×100 mesh was also used to solve one case with results changing less than 1%. A constant time step of $\Delta t = 0.001s$ is used for the simulations.

Figure 2.8 shows the fluid domain with structural deformation at the bottom, at $t = 19s$ (near maximum deflection) for the case with $\rho_s = 50kg/m^3$. Figure 2.9 and 2.10 compare the transient results obtained by the proposed semi-implicit method against a fully implicit coupling scheme. The results of the semi-implicit-M method are not shown in the figures for the sake of clarity of the pictures. As seen in the figures, the semi-implicit method provides an excellent accuracy with reference to an implicit method. The discrepancy for the case with $\rho_s = 50$ is very small (0.4% max). The semi-implicit-M method reduces the discrepancy marginally by 0.1%. For the case with $\rho_s = 5$, the semi-implicit method has a maximum of 2.5% error in the peak and trough points, with respect to the implicit method. The semi-implicit-M method reduces this discrepancy to 1.4%. It must be noted that the over-prediction at the peak points by the semi-implicit method does not indicate a stability problem. Although the peak values in Fig 10 are bigger for the semi-implicit method, the trough points are higher as well. It means the semi-implicit method does not amplify the oscillations artificially. The oscillation amplitude matches very well with that of the implicit method (0.5% max. discrepancy) and only the center line is slightly shifted upwards.

Table 2.10 contains the average number of coupling iterations and the overall solution time for each method, normalized by the smaller one. All three coupling methods have been used with the same interface solver (FP-Aitken). As results shows, the semi-implicit method is significantly cheaper than the fully implicit scheme, especially when higher number of coupling iterations are required. Again the semi-implicit-M method has a significantly higher CPU time compared to the semi-implicit method.

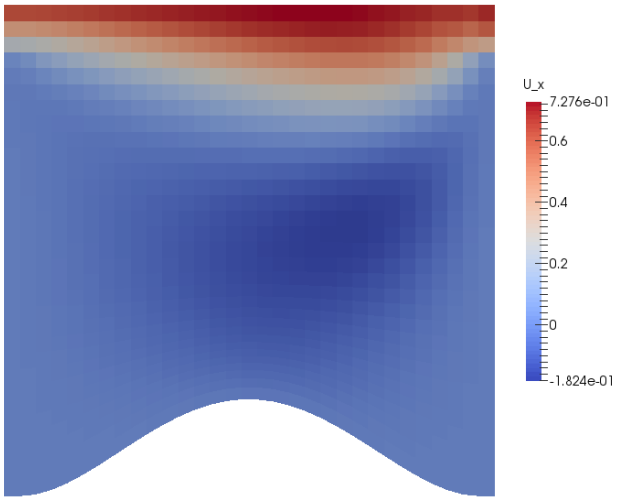

Figure 2.8: Flow field inside the cavity with deformed bottom, $t = 19s$, $\rho_s = 50kg/m^3$.

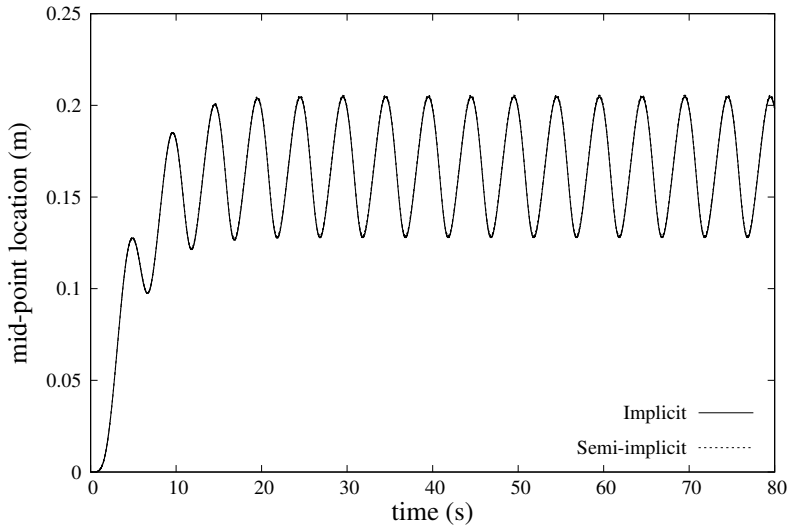

Figure 2.9: Displacement of the mid-point of the structure over time, $\rho_S = 50kg/m^3$.

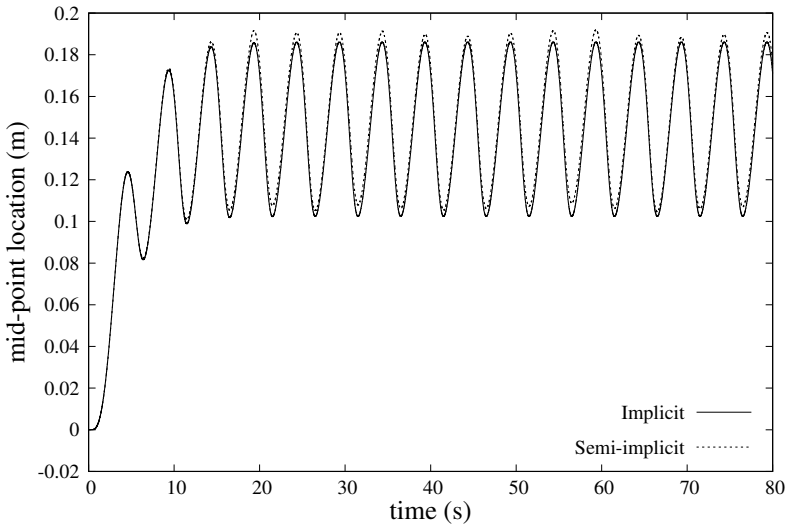

Figure 2.10: Displacement of the mid-point of the structure over time, $\rho_S = 5kg/m^3$.

Table 2.10: Performance comparison of implicit and semi-implicit coupling methods for driven cavity with flexible bottom.

	Coupling technique	Average No. of coupling iterations	Normalized CPU time
$\rho_s = 50$	Implicit	2.5	2.2
	Semi-implicit	3.1	1.0
	Semi-implicit-M	2.6	2.0
$\rho_s = 5$	Implicit	7.8	5.9
	Semi-implicit	6.3	1.0
	Semi-implicit-M	6.1	4.2

The results of the third test case support the conclusions drawn in the previous sections. It shows that the semi-implicit method is capable of delivering a very good accuracy, compared to a fully implicit scheme. It also shows that implicit coupling of the dynamic mesh step (semi-implicit-M method) does not significantly improve the accuracy of the results, even for a case with large structural deformations. However, the semi-implicit-M method considerably increases the computational time.

2.6 Conclusions

A semi-implicit coupling method is proposed for fluid-structure interaction problems with strong added-mass effect. A fractional-step method is used to split the pressure stress term of the fluid and implicitly couple it to the structure. The remaining fluid terms are only explicitly coupled. An ALE formulation and conforming mesh technique is used to solve the fluid flow in a moving domain. A parallel radial basis function method is used to move the computational grid. The main advantages of the proposed method are:

- It is very simple, modular and completely matrix-free.
- Its robustness is tested in different FSI problems.
- It is computationally efficient.

Numerical tests are performed on three widely different test cases, which demonstrate adequate stability, accuracy and efficiency of the proposed method in different types of FSI problems, including ones with large deformations. Results of the simulations are validated against experimental data and other numerical results from the literature. A comparison is made between the accuracy and performance of the proposed semi-implicit method and a fully implicit coupling technique. Results show that the semi-implicit method significantly reduces the computational cost of the simulations without undermining either stability or accuracy of the results.

Moreover the effect of implicit or explicit coupling of the dynamic mesh step on overall performance and accuracy of the semi-implicit method is evaluated. Results show that implicit coupling of the dynamic mesh step remarkably increases the computational cost while only slightly improving the accuracy. This conclusion stands even for problems with large structural deformation.

Furthermore, we have used both fixed-point and Newton-Krylov methods to solve the interface problem. It is shown that the Newton-Krylov solver outperforms the fixed-point method in a problem that requires many iterations to converge. However, it also introduces extra complexity to the problem because of calculating the Jacobian. In a FSI problem which does not require many coupling iterations, fixed-point method with Aitken's relaxation is a better candidate, considering its simplicity and good performance.

References

[1] Joris Degroote. Partitioned simulation of fluid-structure interaction. *Archives of Computational Methods in Engineering*, 20:185–238, 2013.

[2] Gene Hou, Jin Wang, and Anita Layton. Numerical methods for fluid-structure interaction - A review. *Communications in Computational Physics*, 12(2):337–377, 2012.

[3] M Lesoinne and C Farhat. Higher-order subiteration-free staggered algorithm for nonlinear transient aeroelastic problems. *AIAA Journal*, 36(9):1754–1757, 1998.

[4] U. Küttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43:61–72, 2008.

[5] C Michler, E H Van Brummelen, and R De Borst. An interface Newton-Krylov solver for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 47(10-11):1189–1195, 2005.

[6] Miguel Angel Fernández and Marwan Moubachir. A Newton method using exact Jacobians for solving fluid–structure coupling. *Computers and Structures*, 83:127–142, 2005.

[7] Jean Frédéric Gerbeau and Marina Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:631–647, 2003.

[8] U. KuÌ€ttler and W. A. Wall. Vector extrapolation for strong coupling fluid-structure interaction solvers. *Journal of Applied Mechanics*, 76(2):021205, 2009.

[9] Charbel Farhat, Kristoffer G. van der Zee, and Philippe Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195:1973–2001, 2006.

[10] EH Van Brummelen. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *Journal of Applied mechanics*, 76(2):021206, 2009.

[11] 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*, 194:4506–4527, 2005.

[12] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196:1278–1293, 2007.

[13] 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. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.

[14] M Breuer and M Münsch. Fluid-structure interaction using LES-A partitioned coupled predictor-corrector scheme. *PAMM*, 8(1):10515–10516, 2008.

[15] M. Breuer, G. De Nayer, M. Münsch, T. Gallinger, and R. Wüchner. Fluid-structure interaction using a partitioned semi-implicit predictor-corrector coupling scheme for the application of large-eddy simulation. *Journal of Fluids and Structures*, 29:107–130, 2012.

[16] Jón Tómas Grétarsson, Nipun Kwatra, and Ronald Fedkiw. Numerically stable fluid–structure interactions between compressible flow and solid structures. *Journal of Computational Physics*, 230(8):3062–3084, 2011.

[17] M. Astorino, F. Chouly, and M. A. Fernández. Robin based semi-implicit coupling in fluid-structure interaction: Stability analysis and numerics. *SIAM Journal on Scientific Computing*, 31(6):4041–4065, 2009.

[18] T. He, K. Zhang, and T. Wang. Ac-cbs-based partitioned semi-implicit coupling algorithm for fluid-structure interaction using stabilized second-order pressure scheme. *Communications in Computational Physics*, 21(5):1449–1474, 2017.

[19] I González, O Lehmkuhl, A Naseri, J Rigola, and A Oliva. Fluid-structure interaction of a reed type valve. In *International Compressor Engineering Conference*, page 2490. Purdue University Libraries, 2016.

[20] I. González, A. Naseri, J. Rigola, C. D. Pérez-Segarra, and A. Oliva. A fluid-structure interaction solver for the fluid flow through reed type valves. *IOP Conference Series: Materials Science and Engineering*, 232(1):012032, 2017.

[21] CW Hirt, Anthony A Amsden, and JL Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of computational physics*, 14(3):227–253, 1974.

[22] Jean Donea, S Giuliani, and Jean-Pierre Halleux. An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions. *Computer methods in applied mechanics and engineering*, 33(1-3):689–723, 1982.

[23] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22:745–762, 1968.

[24] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187(1):343–368, 2003.

[25] Lluís Jofre, Oriol Lehmkuhl, Jordi Ventosa, F Xavier Trias, and Assensi Oliva. Conservation properties of unstructured finite-volume mesh schemes for the navier-stokes equations. *Numerical Heat Transfer, Part B: Fundamentals*, 65(1):53–79, 2014.

[26] A Gholami, D Malhotra, H Sundar, and G Biros. FFT, FMM, or multigrid? A comparative study of state-of-the-art Poisson solvers for uniform and nonuniform grids in the unit cube. *SIAM Journal on Scientific Computing*, 38(3):C280–C306, 2016.

[27] F. X. Trias, O. Lehmkuhl, A. Oliva, C. D. Pérez-Segarra, and R.W.C.P. Verstappen. Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids. *Journal of Computational Physics*, 258:246–267, 2014.

[28] O. Estruch, O. Lehmkuhl, R. Borrell, C. D Pérez Segarra, and A. Oliva. A parallel radial basis function interpolation method for unstructured dynamic meshes. *Computers and Fluids*, 80:44–54, 2013.

[29] Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995.

[30] D. a. Knoll and David E. Keyes. Jacobian-free Newton-Krylov methods: A survey of approaches and applications. *Journal of Computational Physics*, 193(2):357–397, 2004.

[31] Yousef Saad and Martin H. Schultz. GMRES: A Generalized Minimal Residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[32] Henk A Van der Vorst. Bi-CGSTAB: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems. *SIAM Journal on scientific and Statistical Computing*, 13(2):631–644, 1992.

[33] Stanley C. Eisenstat and Homer F. Walker. Choosing the forcing terms in an inexact Newton method. *SIAM Journal on Scientific Computing*, 17(1):16–32, 1996.

[34] L. Formaggia, J. F. Gerbeau, F. Nobile, and A. Quarteroni. On the coupling of 3D and 1D Navier-Stokes equations for flow problems in compliant vessels. *Computer Methods in Applied Mechanics and Engineering*, 191(6-7):561–582, 2001.

[35] Alfio Quarteroni, Massimiliano Tuveri, and Alessandro Veneziani. Computational vascular fluid dynamics: Problems, models and methods. *Computing and Visualization in Science*, 2:163–197, 2000.

[36] A. Naseri, O. Lehmkuhl, I. Gonzalez, and A. Oliva. Partitioned semi-implicit methods for simulation of biomechanical fluid-structure interaction problems. *Journal of Physics: Conference Series*, 745(3):032020, 2016.

[37] P Anagnostopoulos and PW Bearman. Response characteristics of a vortex-excited cylinder at low reynolds numbers. *Journal of Fluids and Structures*, 6(1):39–50, 1992.

[38] Karl W Schulz and Yannis Kallinderis. Unsteady flow structure interaction for incompressible flows using deformable hybrid grids. *Journal of Computational Physics*, 143(2):569–597, 1998.

[39] J Yang, S Preidikman, and E Balaras. A strongly coupled, embedded-boundary method for fluid–structure interactions of elastically mounted rigid bodies. *Journal of Fluids and Structures*, 24(2):167–182, 2008.

3

A second-order time accurate semi-implicit method

Main content of this chapter has been published in:

A. Naseri, I. Gonzalez, A. Amani, C. D. Pérez-Segarra, and A. Oliva. A second–order time accurate semi–implicit method for fluid–structure interaction problems. *Journal of Fluids and Structures*, 86:135–155, 2019.

Abstract. This chapter is concerned with numerical solution of fluid-structure interaction (FSI) problems involving an incompressible viscous flow and an elastic structure. A semi-implicit partitioned method with second-order temporal accuracy is proposed. The method separates the pressure term of the fluid equations and strongly couples it to the structure, while the remaining fluid terms and the geometrical nonlinearities are treated explicitly. A second-order projection method is used to solve the fluid equations and also as a framework for the FSI coupling. Particular attention is paid to the boundary conditions for fluid equations and the accuracy of the fluid pressure on the common interface. The proposed coupling method retains the second-order accuracy for fully-coupled nonlinear FSI problems. Extensive numerical tests are carried out on a number of benchmark FSI problems and the second-order temporal accuracy for all the variables of interest (fluid velocity and pressure, and structural displacement) is demonstrated.

3.1 Background

Partitioned methods for FSI problems are generally divided into explicit (or loosely coupled) and implicit (or strongly coupled) techniques. Explicit partitioned methods solve the fluid and structural equations in sequence and only once per time step. These methods do not satisfy the exact equilibrium conditions on the interface, which causes instability issues in a range of FSI problems (the so-called added-mass instability) [1,2]. Implicit methods, on the other hand, use coupling iterations between fluid and structural solvers to enforce the equilibrium condition on the interface. These methods are stable for problems with strong added-mass effect, however, their computational cost is generally high due to the repetitive solution of the governing equations at each time step [3,4].

In a partitioned method, the equilibrium conditions on the interface are applied as boundary conditions on each sub-problem, through a decomposition method. Dirichlet-Neumann decomposition is a classical and widely used method in partitioned solution of FSI problems (see e.g. [5–7]). The name of the method indicates that a Dirichlet boundary condition is used for the fluid equations and a Neumann boundary condition for the structure. Therefore, the fluid equations are solved for a known displacement of the solid, while the structural equations are solved for a known stress on the interface. This is a simple decomposition and it is consistent with the most common numerical methods for fluid and structural equations. There is a more recent class of Robin-based decomposition methods that use a Robin boundary condition for the fluid and either a Robin or Neumann boundary condition for the structure [8–10]. Robin-based methods are gaining popularity as they allow a loosely-coupled and yet added-mass-free FSI coupling. However, they require using an especial fluid solver capable of handling a Robin boundary condition. Thus they might not be readily usable with some of the most common fluid solvers. In this work we use a Dirichlet-Neumann decomposition for its advantages of simplicity and consistency.

Instability in loosely coupled methods with Dirichlet-Neumann decomposition is caused by the added-mass effect. The added-mass instability is, in principle, independent of the time step size or the particular discretization method used for each sub-problem solver. It is rather inherent to the coupling method and it has a particularly strong effect in FSI problems with incompressible flow and similar densities of fluid and solid [1,2]. It is argued that the fluid pressure term is the main contributor to the added-mass effect and its explicit coupling would cause instability issues [1]. This was the main motivation for a new category of partitioned methods, first proposed by Fernandez et al. [11], and called semi-implicit methods. In a semi-implicit coupling method, the fluid pressure term is segregated and strongly coupled to the structure, while the remaining fluid terms are only loosely coupled. Segregation of the pressure term could be naturally achieved by using a classical Chorin-Temam pro-

jection method [12]. Strong coupling of the fluid pressure and structural deformation eliminates the added-mass instability issue, while loose coupling of the remaining fluid terms helps avoiding excessive computational cost [11].

A similar semi-implicit method was proposed by Breuer et al. [13] where the geometrical nonlinearities are also coupled implicitly. Astorino et al. [14] improved the stability of the method in [11] by using a specific Robin treatment of the explicit part of the coupling derived from Nitsche's method. An enhanced semi-implicit method was proposed by Naseri et al. [15, 16]. Numerical tests showed that the accuracy of the method in a practical problem is very similar to a fully implicit scheme (monolithic solution), while its computational cost is remarkably smaller [16]. The application of the method was extended to turbulent flow [17,18] and non-Newtonian fluids [19]. Other similar semi-implicit methods are presented in [20, 21] which use a characteristic-based split (CBS) scheme instead of Chorin-Temam projection method. It should be noted that these semi-implicit methods are different from the methods in [22–24]–which are also sometimes called semi-implicit. In the methods in [22–24], the location of the interface is treated explicitly (extrapolated in time) and the fluid mesh is moved once in a time step, however, the fluid and structural equations are solved completely at each coupling iteration (i.e. only the geometrical nonlinearities are treated explicitly).

Semi-implicit coupling techniques rely on a projection method to solve the fluid equations and segregate the pressure term. The fluid pressure term is then strongly coupled to the structure via coupling iterations. Thus, the projection method does not only serve to solve the fluid equations but also as a framework for the FSI coupling. The semi-implicit methods in [11, 14, 16] have used a first-order Chorin-Temam projection method, while [20] have used a first-order CBS scheme for this purpose. Therefore, the overall temporal accuracy of these methods is at most one. Methods in [13, 21] have used apparently second-order pressure splitting schemes, but no error analysis (neither analytical nor numerical) was presented to show that a second-order temporal accuracy was actually achieved for a FSI solution. Extending temporal accuracy of projection methods to higher orders is not straightforward, as discussed in [25–27]. Although it is relatively easy to achieve second-order accuracy for velocity, fluid pressure remains only first-order accurate for many projection methods in the literature [25–27]. Considering that the fluid pressure is a main acting force on the structure, second-order accuracy for pressure is essential to achieve a second-order FSI solution. Moreover, mesh-conforming FSI solution methods require solving the Arbitrary Lagrangian–Eulerian (ALE) form of the Navier-Stokes equations on a moving mesh. A method for solving the fluid equations on a dynamic grid and evaluating the geometrical terms arising from the ALE formulation does not necessarily preserve the order of accuracy of the method on a fixed grid (see e.g. [28]). Furthermore, if the FSI coupling technique is not properly designed, the second-order

accuracy for the coupled problem is not guaranteed, even though each sub-problem possessed such accuracy.

In point of fact, there are few second-order time accurate methods among other types of implicit and explicit partitioned methods in the literature. Farhat et al. [29] proposed a second-order loosely coupled partitioned method for FSI problems in aeroelasticity. This method is explicit so it is not suitable for strong added-mass problems. Nobile et al. [23] reported up to fourth-order time-accurate implicit partitioned methods. In some versions of the proposed method, the location of the interface is extrapolated in time (explicit treatment of geometrical nonlinearities). Liu et al. [24] proposed a second-order technique based on a combined field method with explicit treatment of the interface location. The method was shown to be stable regardless of the mass ratio. Oyekole et al. [30] proposed a second order partitioned method based on a Robin boundary condition for the fluid. The structural inertia term is included in the Robin boundary condition for the fluid which makes the scheme stable for strong added-mass cases.

In this work, we propose a semi-implicit partitioned FSI method that is second-order accurate in time. This is, to the best of our knowledge, the first semi-implicit partitioned FSI method with a demonstrated second-order accuracy. A projection method is used to segregate the fluid pressure term, which is then strongly coupled to the structure via coupling iterations. In order to obtain second-order accuracy in a FSI solution, four important steps are taken:

i) Presenting an incremental projection method and discretization in time that actually yields second-order accuracy for fluid pressure, as well as velocity.

ii) Deriving specific projection-consistent boundary conditions for all fluid boundaries including the interface with the solid.

iii) Developing an ALE scheme on a moving grid and evaluating the arisen geometrical terms with second-order accuracy.

iv) Properly coupling the fluid and structural solvers in order to retain the second-order accuracy for a coupled nonlinear FSI problem.

The second-order accuracy of the method for realistic nonlinear FSI problems is demonstrated through rigorous numerical tests. Three widely distinct FSI test cases are studied and an analysis is made to show the second-order rate of convergence of the error.

The remaining of this chapter is organized as follows. In section 2 the governing equations for each sub-problem and the coupling conditions are presented. Section 3 describes the proposed numerical method, while numerical tests are presented in section 4. Section 5 summarizes and concludes the article.

3.2 Governing Equations

In this section, the governing equations for each sub-problem domain and the coupling conditions on the interface are presented. The fluid and structural domains are referred to as $\Omega_f(t) \subset \mathbb{R}^3 \times (0,T)$ and $\Omega_s(t) \subset \mathbb{R}^3 \times (0,T)$ respectively, as they both vary in time $t \in (0,T)$. The fluid-structure interface is the shared boundary of the domains, denoted by $\Gamma(t) = \partial\Omega_f(t) \cap \partial\Omega_s(t)$.

3.2.1 Fluid equations

The unsteady flow of an incompressible viscous fluid is governed by the Navier-Stokes equations. An Arbitrary Lagrangian-Eulerian (ALE) formulation of these equations in a moving domain is given by:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{u} = \frac{1}{\rho_f} \nabla \cdot \boldsymbol{\sigma}_f \tag{3.1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{3.2}$$

where $\mathbf{u}$ is the fluid velocity and ρ_f the fluid density. Vector $\mathbf{c}$ is the ALE convective velocity $\mathbf{c} = \mathbf{u} - \mathbf{w}$, which is the fluid velocity relative to a domain moving with a velocity $\mathbf{w}$. The stress tensor $\boldsymbol{\sigma}_f$ is defined for a Newtonian fluid as:

$$\boldsymbol{\sigma}_f = -p\mathbf{I} + 2\mu_f \boldsymbol{\gamma} \tag{3.3}$$

where p is the fluid pressure, $\mathbf{I}$ the unit tensor, μ_f the dynamic viscosity of the fluid and $\boldsymbol{\gamma}$ the strain rate tensor given by:

$$\boldsymbol{\gamma} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \tag{3.4}$$

3.2.2 Structural equations

The structural domain is governed by the nonlinear elastodynamics equation:

$$\rho_s \frac{\partial^2 \mathbf{d}}{\partial t^2} = \nabla \cdot \mathbf{P} \tag{3.5}$$

where $\mathbf{d}$ stands for the structural position with respect to the reference configuration, and the structural density is shown by ρ_s. The tensor $\mathbf{P}$ is the first Piola-Kirchhoff tensor, which is related to the Cauchy stress tensor $\boldsymbol{\sigma}_s$ by:

$$\mathbf{P} = J\boldsymbol{\sigma}_s \mathbf{F}^{-T} \tag{3.6}$$

where $\mathbf{F}$ is the deformation gradient $\mathbf{F} = \nabla \mathbf{d}$ and J its determinant ($J = det(\mathbf{F})$).

The FSI method is presented for a general structure, however, for the numerical tests, different simplified structural models are considered which are described for each test case in section 3.4.

3.2.3 Coupling conditions

The coupling conditions apply at the interface Γ and account for the interaction of the domains. They are derived from the kinematic and dynamic equilibrium between the domains, which yield to the following conditions on a non-slip type interface:

$$\mathbf{u}_\Gamma = \frac{\partial \mathbf{d}_\Gamma}{\partial t} \tag{3.7}$$

$$\boldsymbol{\sigma}_s \mathbf{n}_\Gamma = \boldsymbol{\sigma}_f \mathbf{n}_\Gamma \tag{3.8}$$

for any $\mathbf{x} \in \Gamma$, where $\mathbf{n}_\Gamma$ is the unit normal vector on the interface. Equation (3.7) represents the equality of the velocities of the fluid and the structure on the interface to assure the kinematic equilibrium. Equation (3.8) represents the equality of the traction on the interface for dynamic equilibrium.

3.3 Numerical Method

In this section, the temporal discretization of the governing equations and the FSI coupling method are presented. This time-discretized set of equations are independent of the choice for spatial discretization, which could be carried out using either a finite-volume or a finite-element method. The spatial discretization methods used in this work are described in Appendix A.

3.3.1 Fluid solver

Using a second-order central difference scheme around the mid-time-step point $t^{n+1/2}$ for the transient term of the momentum equation, a second-order time-discretized form of the Eq. (3.1) and (3.2) reads:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -(\mathbf{c} \cdot \nabla \mathbf{u})^{n+1/2} + \frac{\mu_f}{\rho_f} \nabla^2 \mathbf{u}^{n+1/2} - \frac{1}{\rho_f} \nabla p^{n+1/2} \tag{3.9}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \tag{3.10}$$

for any $\mathbf{x} \in \Omega_f^{n+1}$, and a proper set of boundary conditions:

$$\begin{cases} \mathbf{u}_{\Gamma}^{n+1} = \dfrac{\partial \mathbf{d}_{\Gamma}^{n+1}}{\partial t} & \text{on } \Gamma^{n+1} \\ \mathbf{u}_{\partial_D}^{n+1} = \mathbf{u}_b & \text{on } \partial_D \Omega_f^{n+1} \\ \dfrac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}}|_{\partial_N} = 0 & \text{on } \partial_N \Omega_f^{n+1} \end{cases} \tag{3.11}$$

where ∂_D and ∂_N represent, respectively, the Dirichlet and Neumann fluid boundaries, $\mathbf{n}$ is the normal unit vector and $\mathbf{u}_b$ is the assigned velocity on the boundary. The boundary condition on the fluid-structure interface Γ^{n+1} comes from the kinematic equilibrium condition and is applied as a Dirichlet boundary condition for the fluid equations (Eq. $(3.11)_1$).

We use an incremental pressure-correction projection method, similar to [31, 32], to solve the momentum equation. Thus, an intermediate velocity is evaluated using the last known pressure field. Unlike the original projection method of Chorin-Temam [12], this method does not impose a formal first-order splitting error. We use an explicit Adams-Bashforth method for the convective term and a Crank-Nicolson method for the diffusive term. Therefore, the intermediate velocity field, $\mathbf{u}^*$, is evaluated as:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -[\frac{3}{2}(\mathbf{c} \cdot \nabla \mathbf{u})^n - \frac{1}{2}(\mathbf{c} \cdot \nabla \mathbf{u})^{n-1}] + \frac{\mu_f}{2\rho_f}(\nabla^2 \mathbf{u}^* + \nabla^2 \mathbf{u}^n) - \frac{1}{\rho_f}\nabla p^{n-1/2} \tag{3.12}$$

This velocity field is then projected onto a space of divergence-free vector fields:

$$\mathbf{u}^* = \mathbf{u}^{n+1} + \frac{\Delta t}{\rho_f}\nabla \phi^{n+1} \tag{3.13}$$

$$\nabla \cdot \mathbf{u}^{n+1} = 0 \tag{3.14}$$

where ϕ^{n+1} is a scalar field obtained by:

$$\nabla^2 \phi^{n+1} = \frac{\rho_f}{\Delta t}\nabla \cdot \mathbf{u}^* \tag{3.15}$$

Substituting Eq. (3.13) into Eq. (3.12) and comparing it to Eq. (3.9), the equation to recover the fluid pressure at the mid-time step is obtained as:

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1} - \frac{\mu_f \Delta t}{2\rho_f} \nabla^2 \phi^{n+1} \tag{3.16}$$

Pressure at the new time station t^{n+1} could be evaluated by a second-order extrapolation from half-time levels:

$$p^{n+1} = \frac{3}{2} p^{n+1/2} - \frac{1}{2} p^{n-1/2} \tag{3.17}$$

Pressure at t^{n+1} is not used in the discretized fluid equations but it is needed to evaluate the fluid force on the structure at the new time step.

Remark 1. The last term in the pressure recovery equation (Eq. (3.16)) was missed in some projection methods, however, it is essential for retaining the second-order accuracy for pressure up to the boundary. It was first introduced (in a slightly different form) in [33]. A common practice in the literature is to use a uniform Neumann boundary condition (zero normal gradient) for the Poisson's equation for ϕ (Eq. (3.15)). Without the last term of Eq. (3.16), the Neumann boundary condition for the scalar field ϕ transmits to the pressure itself and creates an artificial boundary layer which degrades the accuracy. Interested readers are advised to consult [25,27] for some analyses. The boundary conditions in the present work are further discussed in the next section.

Remark 2. A common inaccuracy in many methods in the literature is omitting the extrapolation of pressure (Eq. (3.17)) and evaluating p^{n+1} by Eq. (3.16). However, that would not be consistent with the central time discretization at Eq. (3.9). If the pressure at Eq. (3.16) were considered to be at time t^{n+1} instead of $t^{n+1/2}$, it will always carry a first-order error due to the time lag.

3.3.2 Boundary conditions

In this section the boundary conditions for the predicted velocity $\mathbf{u}^*$ and the scalar field ϕ are described in detail. We recognize three regions of the boundary with different boundary conditions, as in Eq. (3.11). The first region is the fluid-structure interface Γ. The boundary condition for velocity comes from the kinematic equilibrium on the interface (Eq. $(3.11)_1$). Using a second-order backward difference scheme it reads:

$$\mathbf{u}_\Gamma^{n+1} = \frac{\partial \mathbf{d}_\Gamma^{n+1}}{\partial t} = \frac{3\mathbf{d}_\Gamma^{n+1} - 4\mathbf{d}_\Gamma^{n} + \mathbf{d}_\Gamma^{n-1}}{2\Delta t} \qquad \text{on } \Gamma^{n+1} \tag{3.18}$$

which is a Dirichlet boundary condition for velocity. However, there is no specific boundary condition for pressure. A common approach is to use a uniform Neumann boundary condition (zero normal gradient on the boundary) for pressure. Such a boundary condition would create an artificial boundary layer near the interface that would degrade the accuracy. Numerical tests in [25, 27] demonstrate the artificial boundary layer and the loss of accuracy on a wall boundary.

In this work we use a zero normal gradient boundary condition for the scalar field ϕ:

$$\mathbf{n} \cdot \nabla \phi^{n+1}|_\Gamma = 0 \qquad \text{on } \Gamma^{n+1} \tag{3.19}$$

which is a very convenient boundary condition for the Poisson equation (Eq. (3.15)). However, because of the last term in the pressure recovery equation, Eq. (3.16), the Neumann boundary condition is not transmitted to the pressure itself and the artificial boundary layer is avoided. This is essential to achieve a second-order accuracy for pressure up to the boundary.

For the predicted velocity field, we drive a boundary condition consistent with Eq. (3.13):

$$\mathbf{u}_\Gamma^* = \mathbf{u}_\Gamma^{n+1} + \frac{\Delta t}{\rho_f} \nabla \phi^{n+1}|_\Gamma \tag{3.20}$$

Dividing Eq. (3.20) into its components using the normal and tangential unit vectors on the boundary, $\mathbf{n}$ and $\boldsymbol{\tau}$, we get:

$$\mathbf{n} \cdot \mathbf{u}^*|_\Gamma = \mathbf{n} \cdot \mathbf{u}^{n+1}|_\Gamma + \frac{\Delta t}{\rho_f} \mathbf{n} \cdot \nabla \phi^{n+1}|_\Gamma \tag{3.21}$$

$$\boldsymbol{\tau} \cdot \mathbf{u}^*|_\Gamma = \boldsymbol{\tau} \cdot \mathbf{u}^{n+1}|_\Gamma + \frac{\Delta t}{\rho_f} \boldsymbol{\tau} \cdot \nabla \phi^{n+1}|_\Gamma \tag{3.22}$$

Considering boundary conditions (3.18) and (3.19), and using an explicit extrapolation for the gradient term, the boundary conditions for predicted velocity are obtained as:

$$\begin{cases} \mathbf{n} \cdot \mathbf{u}^*|_\Gamma = \mathbf{n} \cdot \dfrac{\partial \mathbf{d}_\Gamma^{n+1}}{\partial t}|_\Gamma \\ \\ \boldsymbol{\tau} \cdot \mathbf{u}^*|_\Gamma = \boldsymbol{\tau} \cdot \dfrac{\partial \mathbf{d}_\Gamma^{n+1}}{\partial t}|_\Gamma + \dfrac{\Delta t}{\rho_f} \boldsymbol{\tau} \cdot (2\nabla \phi^n - \nabla \phi^{n-1})|_\Gamma \end{cases} \qquad \text{on } \Gamma^{n+1} \tag{3.23}$$

Similar boundary conditions are used for other fluid boundary regions with a Dirichlet boundary condition for velocity ($\partial_D \Omega_f^{n+1}$ in Eq. (3.11)), including stationary

walls and flow inlets with a known velocity. Equations (3.23) are modified to use the assigned velocity in place of the interface velocity (e.g. $\mathbf{u}_b = 0$ for stationary walls).

The third fluid boundary region is where a Neumann boundary condition is used for velocity (referred to as $\partial_N \Omega_f^{n+1}$ in Eq. (3.11)). Pressure on the boundary could be specified by a Dirichlet boundary condition (e.g. an outlet with a known discharge pressure) or left without any specific boundary condition (e.g in openings and vents with unknown pressure). In case of a Dirichlet boundary condition for pressure, an equivalent Dirichlet condition is applied on the scalar field ϕ. In the case that no physical boundary condition is specified for pressure, a zero normal gradient boundary condition is used for ϕ. Therefore, the set of boundary conditions are as follows:

$$\begin{cases} \dfrac{\partial \mathbf{u}^{n+1}}{\partial \mathbf{n}}|_{\partial_N} = 0 & \\ & \text{on } \partial_N \Omega_f^{n+1} \\ \phi^{n+1} = \phi_b \qquad \text{or} \qquad \mathbf{n} \cdot \nabla \phi^{n+1}|_{\partial_N} = 0 & \end{cases} \tag{3.24}$$

Again a consistent boundary condition for the predicted velocity is derived. Getting normal derivative of Eq. (3.13) and applying condition $(3.24)_1$, it reads:

$$\frac{\partial \mathbf{u}^*}{\partial \mathbf{n}}|_{\partial_N} = \frac{\Delta t}{\rho_f} \frac{\partial}{\partial \mathbf{n}} \nabla \phi^{n+1}|_{\partial_N} \qquad \text{on } \partial_N \Omega_f^{n+1} \tag{3.25}$$

Depending on the type of boundary condition for ϕ (conditions $(3.24)_2$), the normal or tangential component of the right hand side term in Eq. (3.25) might be zero. The non-zero components of the derivative of the gradient are evaluated on the boundary and the consistent boundary condition (Eq. (3.25)) is applied on the predicted velocity field.

Remark 3. It should be mentioned that similar consistent boundary conditions for simpler boundary of an stationary wall (Dirichlet boundary condition for velocity) were proposed in [25, 34].

3.3.3 Dynamic mesh

We use a conforming mesh technique, which means the fluid mesh moves to adapt to the new location of the interface. A parallel moving mesh technique, based on radial basis function interpolation method [35], is used to move the fluid grid in accordance to the new location of the interface and define the discretized fluid domain at the new time step Ω_f^{n+1}.

The method uses the known displacement on the interface to evaluate an interpolated value for the interior vertices of the fluid grid. A great advantage of this method is that it does not need the connectivity of the mesh elements and can be applied to both structured and unstructured grids. The interpolated displacement $\delta\mathbf{r}$ at a grid vertex $\mathbf{x}_v$ is evaluated as:

$$\delta\mathbf{r}(\mathbf{x}_v) = \sum_{i=1}^{nv} \gamma_i \varphi(||\mathbf{x}_v - \mathbf{x}_i||) \tag{3.26}$$

where nv is the number of nodes on the interface $\mathbf{x}_i$, and φ is the radial basis function. The Wendland C^2 function [36] is used for φ since it preserves good quality of the dynamic mesh, specially near the moving interface. The weight coefficients γ_i are evaluated using the known displacements on the interface nodes

$$\delta\mathbf{r}(\mathbf{x}_i) = \delta\mathbf{d}(\mathbf{x}_i) \qquad i = 1, 2, ..., nv \tag{3.27}$$

for $\mathbf{x}_i \in \Gamma$. Therefore, the size of the system of equations to obtain the weight coefficients is limited to the number of known points nv.

Surface velocities are evaluated according to the space conservation law (SCL) which guarantees no volume is lost while moving the grid. For any control volume in the fluid domain, the SCL is stated as:

$$\frac{\partial v}{\partial t} - \int_s \mathbf{w} \cdot \mathrm{d}\mathbf{A} = 0 \tag{3.28}$$

where v and s stand, respectively, for the volume and the boundary surface of a control volume. As before, $\mathbf{w}$ is the domain velocity and $\mathbf{A}$ is the area vector pointing outward.

Time rate of change of volume is equal to the sum of volumes swept by each face of a control volume. In this work we evaluate the domain velocity at each face, $\mathbf{w}_{face}$, based on the volume swept by that face. With a second-order backward discretization it reads:

$$\mathbf{w}_{face}^{n+1} = 3(\frac{\delta v}{A\Delta t}\mathbf{n})^{n+1} - (\frac{\delta v}{A\Delta t}\mathbf{n})^{n} \tag{3.29}$$

where A is the surface area, $\mathbf{n}$ unit normal vector of the face, Δt time step and δv the volume swept by the face at one time step (see Figure 3.1).

More detailed description of the dynamic mesh method could be found in [16, 35]. In the remaining of this chapter, we will use the notation $\mathcal{M}$ to refer to the mesh movement step:

$$(\Omega_f^{n+1}, \mathbf{w}^{n+1}) = \mathcal{M}(\mathbf{d}_\Gamma^{n+1}) \tag{3.30}$$

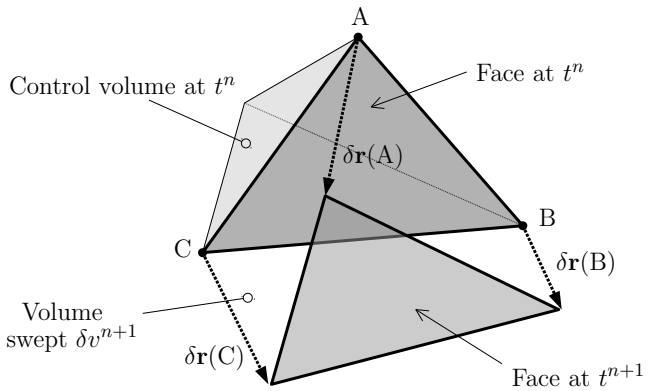

Figure 3.1: Volume swept (δv) by each face of an arbitrarily shaped polyhedral. The displacement of each vertex is shown by δr.

3.3.4 Structural solver

Structural equations are discretized in time using a second-order Newmark method. Defining the structural velocity $\mathbf{v} = \frac{\partial \mathbf{d}}{\partial t}$, we update the velocity and displacement of the structure as:

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \frac{\Delta t}{2\rho_s}[\nabla \cdot \mathbf{P}(\mathbf{d}^{n+1}) + \nabla \cdot \mathbf{P}(\mathbf{d}^n)] \tag{3.31}$$

$$\mathbf{d}^{n+1} = \mathbf{d}^n + \Delta t \mathbf{v}^n + \frac{\Delta t^2}{4\rho_s}[\nabla \cdot \mathbf{P}(\mathbf{d}^{n+1}) + \nabla \cdot \mathbf{P}(\mathbf{d}^n)] \tag{3.32}$$

In this work we use the structural solver as a black-box module. Any structural solver with a second order temporal accuracy could be used. In the remaining of this chapter, we use the notation $\mathcal{S}$ to refer to the structural solver as a function of surface stress on the interface:

$$\mathbf{d}_\Gamma = \mathcal{S}(\boldsymbol{\sigma}_\Gamma) \tag{3.33}$$

where $\mathbf{d}_\Gamma$ is the location of the interface and $\boldsymbol{\sigma}_\Gamma$ is the surface stress on the interface exerted by the fluid $\boldsymbol{\sigma}_\Gamma = \boldsymbol{\sigma}_f(p, \mathbf{u})|_\Gamma \mathbf{n}_\Gamma$.

3.3.5 Coupled problem

We present a semi-implicit FSI coupling method in which only the pressure term of the fluid is strongly coupled to the structure. The remaining fluid terms as well as the dynamic mesh step are evaluated only once per time step. Using a projection method for fluid equations allows us to effectively segregate the pressure term and couple it implicitly to the structure. Strong coupling of the fluid pressure and structural deformation provides for the stability of the method for FSI problems with strong added-mass effect. Loose coupling of the remaining terms helps to avoid excessive computational cost.

The complete FSI solution method from time step t^n to t^{n+1} is as follows:

step 0: extrapolation of $\mathbf{d}_\Gamma$ from previous time steps:

$$\widetilde{\mathbf{d}}_\Gamma^{n+1} = 2.5\mathbf{d}_\Gamma^n - 2\mathbf{d}_\Gamma^{n-1} + 0.5\mathbf{d}_\Gamma^{n-2} \tag{3.34}$$

step 1: moving the fluid mesh (*explicitly coupled*):

$$(\Omega_f^{n+1}, \mathbf{w}^{n+1}) = \mathcal{M}(\widetilde{\mathbf{d}}_\Gamma^{n+1}) \tag{3.35}$$

step 2: ALE convection-diffusion equation (*explicitly coupled*):

$$\begin{aligned} \frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -[\frac{3}{2}(\mathbf{c}\cdot\nabla\mathbf{u})^n - \frac{1}{2}(\mathbf{c}\cdot\nabla\mathbf{u})^{n-1}] \\ + \frac{\mu_f}{2\rho_f}(\nabla^2\mathbf{u}^* + \nabla^2\mathbf{u}^n) - \frac{1}{\rho_f}\nabla p^{n-1/2} \qquad \text{in } \Omega_f^{n+1} \end{aligned} \tag{3.36}$$

step 3: fluid pressure and structural equations (*implicitly coupled, solved iteratively*):

$$\mathbf{n}_\Gamma \cdot \mathbf{u}_\Gamma^* = \mathbf{n}_\Gamma \cdot (\frac{3\mathbf{d}_\Gamma^{n+1} - 4\mathbf{d}_\Gamma^n + \mathbf{d}_\Gamma^{n-1}}{2\Delta t}) \qquad \text{on } \Gamma^{n+1} \tag{3.37}$$

$$\boldsymbol{\tau}_\Gamma \cdot \mathbf{u}_\Gamma^* = \boldsymbol{\tau}_\Gamma \cdot (\frac{3\mathbf{d}_\Gamma^{n+1} - 4\mathbf{d}_\Gamma^n + \mathbf{d}_\Gamma^{n-1}}{2\Delta t}) + \frac{\Delta t}{\rho_f}\boldsymbol{\tau}_\Gamma \cdot (2\nabla\phi^n - \nabla\phi^{n-1})|_\Gamma \qquad \text{on } \Gamma^{n+1} \tag{3.38}$$

$$\nabla^2\phi^{n+1} = \frac{\rho_f}{\Delta t}\nabla\cdot\mathbf{u}^* \qquad \text{in } \Omega_f^{n+1} \tag{3.39}$$

$$p^{n+1/2} = p^{n-1/2} + \phi^{n+1} - \frac{\mu_f \Delta t}{2\rho_f}\nabla^2\phi^{n+1} \qquad \text{in } \Omega_f^{n+1} \tag{3.40}$$

$$p^{n+1} = \frac{3}{2}p^{n+1/2} - \frac{1}{2}p^{n-1/2} \qquad \text{in } \Omega_f^{n+1} \tag{3.41}$$

$$\boldsymbol{\sigma}_\Gamma^{n+1} = \boldsymbol{\sigma}_f(p^{n+1}, \mathbf{u}^*)|_\Gamma \mathbf{n}_\Gamma \qquad \text{on } \Gamma^{n+1} \tag{3.42}$$

$$\mathbf{d}_\Gamma^{n+1} = \mathcal{S}(\boldsymbol{\sigma}_\Gamma^{n+1}) \qquad \text{on } \Gamma^{n+1} \tag{3.43}$$

step 4: velocity correction (*explicitly coupled*):

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{\Delta t}{\rho_f}\nabla\phi^{n+1} \qquad \text{in } \Omega_f^{n+1} \tag{3.44}$$

$$\mathbf{u}_\Gamma^{n+1} = \frac{3\mathbf{d}_\Gamma^{n+1} - 4\mathbf{d}_\Gamma^{n} + \mathbf{d}_\Gamma^{n-1}}{2\Delta t} \qquad \text{on } \Gamma^{n+1} \tag{3.45}$$

Every time step starts with predicting the location of the interface by means of an extrapolation from previous time steps. The fluid mesh is then moved to adapt to the predicted location of the interface (step 1). Therefore, the geometrical nonlinearities are treated explicitly. The convection-diffusion equation in step 2 is also solved only once per time step. Step 3 is the implicit part of the coupling in the above algorithm where fluid pressure is strongly coupled to the structural deformation. Equations in step 3 are solved together (iteratively). This step provides for the stability of the method for FSI problems with strong added-mass effect. Note that the current structural deformation $\mathbf{d}_\Gamma^{n+1}$ is used in Eq. (3.37) and (3.38), while it is evaluated in Eq. (3.43), which shows the implicit coupling between the equations in step 3. A Newton-Krylov method with approximated Jacobian [16] is used to carry out the coupling iterations in step 3.

As seen in the Eq. (3.37) and (3.38), the kinematic equilibrium on the interface is applied on the predicted velocity field (in a corrected consistent manner) during the coupling iterations. The predicted velocity is also used to evaluate the shear stress term in Eq. (3.42). When the convergence is achieved for the iterative process in step 3, the velocity field is corrected using the converged field ϕ^{n+1} and the coupling condition is applied on the final velocity (step 4).

3.3.6 Temporal accuracy

An error analysis is required to evaluate the temporal accuracy of the proposed method in a FSI problem. Analytical and numerical analysis could be used for this purpose. Analytical energy and error estimates have been used in some previous

studies to evaluate stability and order of convergence of FSI methods (e.g. [10,30]). However, these analyses were limited to a simplified linear system of equations (Stokes equations for fluid and a lower dimensional, linear elastic model for structure). The non-linear term in fluid equations represents an important physical aspect and also changes the mathematical characteristics of the governing equations. The non-linear term in solid equations has a similar importance. Evaluating the error bound and convergence rate of numerical methods applied to realistic FSI problems normally rely on numerical experiments.

In this study numerical tests are carried out on three widely used FSI test cases to demonstrate the second-order accuracy of the proposed method. For that purpose, test solutions are carried out using increasingly larger time steps and their error is evaluated at a certain point in time. The spatial grid is kept constant for all the test solutions. In the lieu of an exact solution, reference numerical results are generated for each test case, using a very small time step size and identical spatial grid. Test solutions are compared to the reference solution to evaluate the error for each variable. The errors with respect to the reference solution are evaluated at each spatial point and presented in both L_2 and L_∞ norms. The temporal order of accuracy is determined by evaluating the slope of the graph of error against time step size, in a logarithmic scale.

It should be noted that the global error contains both spatial and temporal components and behaves, in general terms, as $O(\Delta x^a + \Delta t^b)$, where Δx and Δt are the spatial grid size and time step, respectively. The powers a and b are, respectively, the asymptotic orders of spatial and temporal accuracy. In order to evaluate the asymptotic order of temporal accuracy, the spatial component of the error must be kept considerably smaller than the temporal component to make sure it does not affect the error analysis results. When the errors are evaluated with respect to an exact (analytical) solution ($\mathcal{X}_e$), it means the spatial grid should be sufficiently fine to assure a small spatial error. Ideally one may have $\Delta x \ll \Delta t^{b/a}$ so that the error could be approximated as $O(\Delta t^b)$. In our numerical tests, reference numerical results ($\mathcal{X}_r$) are used instead of an exact solution, due to the lack of exact solution for the realistic problems considered. The reference solution is obtained using a certain spatial and temporal grid size (Δx_r and Δt_r), thus itself contains an error with respect to the exact solution, $\epsilon_r = \mathcal{X}_r - \mathcal{X}_e$. Test solutions ($\mathcal{X}_t$) are obtained by using different (and increasingly larger) time step sizes but an identical spatial grid. Therefore, In comparing the test results to the reference results, the error due to the spatial discretization is discarded since it is identical for the test and reference solutions.

To further clarify the aforementioned explanation, let us assume the global error (with respect to the exact solution) to be of the form $O(\Delta x^a + \Delta t^b) = O_1(\Delta x^a) + O_2(\Delta t^b)$. We expect $a = b = 2$ for the method in this work. The error of the reference solution with respect to the exact solution is then $\epsilon_r = O_1(\Delta x_r^a) + O_2(\Delta t_r^b)$. Similarly, the error (with respect to the exact solution) of the test solutions using test time step

size Δt and identical spatial grid is $\epsilon_t = O_1(\Delta x_r^a) + O_2(\Delta t^b)$. When we compare the test results to the reference results and evaluate the error with respect to the reference solution, the error for each test solution becomes $\epsilon = \mathcal{X}_t - \mathcal{X}_r = O_2(\Delta t^b) - O_2(\Delta t_r^b)$, because the spatial error is identical for test and reference solutions. In the numerical tests, we have used a much smaller time step size for the reference results ($\Delta t_r \ll \Delta t$). Therefore, the error with respect to the reference solution becomes $\epsilon \approx O_2(\Delta t^b)$. Thus, by evaluating the error with respect to the reference solution, for different time step sizes, one is able to appropriately asses the temporal accuracy of the method. In the numerical tests, graphs of error against time step size ($\epsilon - \Delta t$) are presented that show a slope of two, which suggest a second-order temporal accuracy according to the above explanation.

It should be also noted that our numerical tests would have revealed a mixed error term of the form $O(\Delta x \Delta t)$. If there existed such a term in the global error (in this work it would be $O_1(\Delta x^2) + O_2(\Delta t^2) + O_3(\Delta x \Delta t)$), the error with respect to the reference solution would become $\epsilon = O_2(\Delta t^b) - O_2(\Delta t_r^b) + O_3(\Delta x_r \Delta t) - O_3(\Delta x_r \Delta t_r)$. Since we have kept the spatial grid constant for the analysis, and the fact that lower order error is the dominant term, we would have $\epsilon \approx O_3(\Delta t)$. This means the slope of the error graphs in the numerical tests would become unity and reveal if a mixed error term existed in the solution.

3.4 Numerical Tests

In this section, numerical results on benchmark FSI problems are presented to study the accuracy of the proposed method. Three FSI test cases are considered, as well as a test case that contains only fluid flow. The three FSI cases vary widely, featuring an internal flow contained by a deformable membrane, an external flow over a blunt body with rigid-body motion, and a cavity flow with a deformable bottom. Obtained results are compared to experimental and numerical results from the literature in order to validate the solution method. An error analysis is provided for each test case that demonstrates second-order temporal accuracy of the solution.

3.4.1 Lid-driven cavity (only fluid flow)

As the first numerical test case, we study the flow inside a classical lid-driven cavity at $Re = 400$. The problem is a 1m $\times$ 1m cavity with the lid moving at a constant velocity of $u_x = 1\text{m/s}$. The fluid density and viscosity are $\rho_f = 1.0\ \text{kg/m}^3$ and $\mu_f = 0.0025$ Pa.s, respectively. All the walls are rigid and there is no fluid-structure interaction.

A classical 31×31 mesh is used to solve the problem. A finer grid of 61×61 is also used for comparison. Spatial discretization is carried out using a finite volume method with second-order central difference schemes. Figure 3.2 compares the steady

state solution for horizontal velocity on the vertical mid-line ($x = 0.5$), against the classical results of Kim and Moin (1985) [34]. The steady state was reached after roughly 30 seconds ($t > 30$s). As seen in figure 3.2, the results obtained by the coarse and fine mesh are almost indistinguishable and they agree very well with those of Kim and Moin (1985) [34].

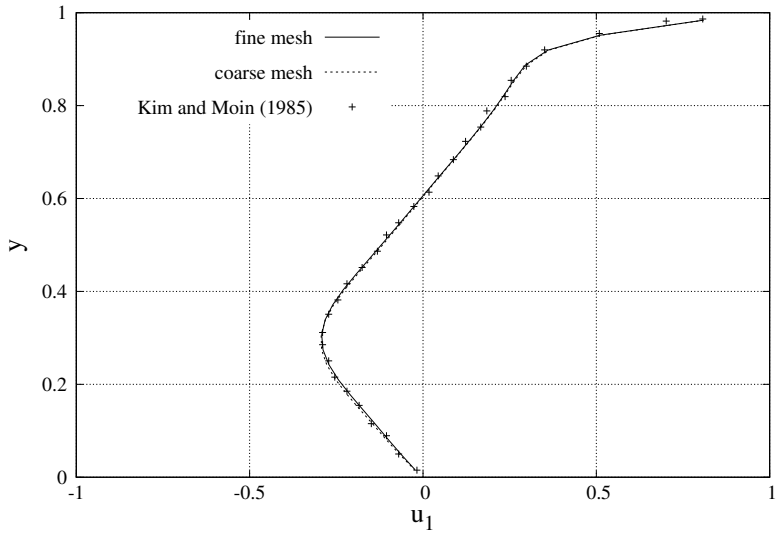

Figure 3.2: Horizontal component of velocity on the vertical mid-line of a lid-driven cavity (NO FSI) in the steady state.

To evaluate the temporal accuracy of the method, four different time step sizes are used to solve the problem from $t = 0$ till $t = 1$s. The time step ranges from 5×10^{-4} to 4×10^{-3}s. In the lieu of an exact solution, reference results are generated using a much smaller time step size of $\Delta t = 5 \times 10^{-5}$s. The reference time step size is an order of magnitude smaller than the smallest test time step. The spatial grid is kept constant for the tests. The error with respect to the reference solution is evaluated at every grid point and presented in both L_2 and L_∞ norms in figure 3.3, on a logarithmic scale. A solid line with a slope of two (Δt^2) is also plotted to compare the slope of the error graphs. A slope of two of the error graphs means the rate of convergence of the error by time step size is two, i.e. a second-order accuracy.

As seen in figure 3.3, both velocity and pressure are clearly second-order accurate. To monitor closely the error of the pressure on the domain boundaries, the error on the boundaries is evaluated separately and plotted in figure 3.3. It demonstrates that

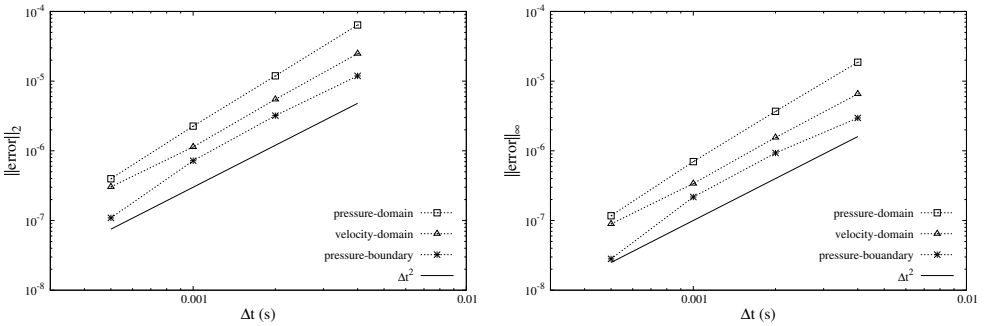

Figure 3.3: Variation of relative error inside the domain and on the boundaries by time step size, driven cavity case (NO FSI). Left: L_2 norm; Right: L_∞ norm.

the proposed method solves the fluid equations with a second-order accuracy up to the domain boundary.

3.4.2 Driven cavity with deformable bottom

Numerical tests are carried out on a benchmark problem studied in [2,37], among others. The test case is a 2-D driven cavity of 1m $\times$ 1m with a flexible bottom wall. The top boundary of the cavity is moving with an oscillatory speed of $u_x(t) = 1 - \cos(\omega t)$ m/s, with $\omega = 2\pi/5$. There are two openings of 0.1m length on the sidewalls that allow the fluid to enter to and exit from the domain. Figure 3.4(left) shows a schematic description of the problem. The fluid density and viscosity are $\rho_f = 1.0$ kg/m^3 and $\mu_f = 0.01$ Pa.s, respectively. The flexible structure at the bottom has a thickness of $h = 0.05$m, the structural density is $\rho_s = 5$ kg/m^3 and the Young modulus $E = 250$ N/m^2.

The flexible bottom is modeled as an Euler-Bernoulli beam, governed by the following equation:

$$\rho_s A \frac{\partial^2 \mathbf{d}}{\partial t^2} + EI \frac{\partial^4 \mathbf{d}}{\partial x^4} = q(x,t) \tag{3.46}$$

where $\mathbf{d} = [0, y, 0]^T$ in a Cartesian coordinate (x, y, z), A is the cross section area of the beam, I the second moment of area, and q is the load per unit length.

A classical 31 $\times$ 31 spatial grid is used to solve the problem. Spatial discretization is carried out using a finite volume method with second-order central difference schemes. The structure is a thin membrane so the fluid mesh elements on the interface are also used as the computational grid for the structural equations. Thus, the

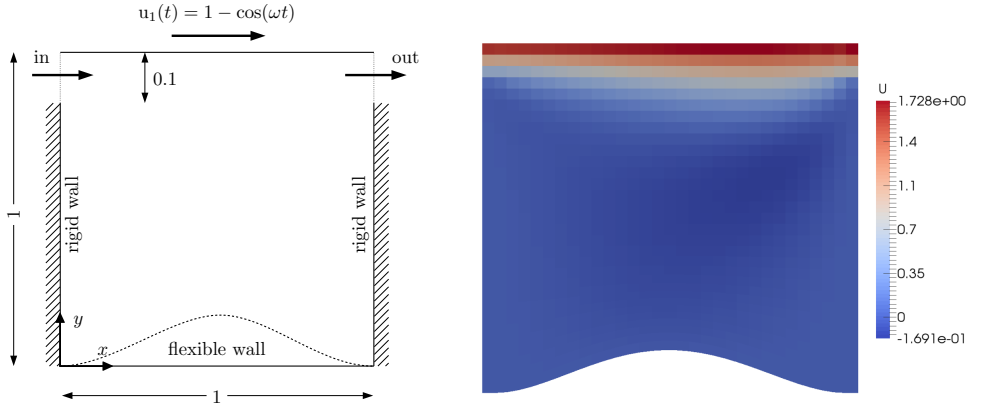

Figure 3.4: Driven cavity with flexible bottom. Left: schematic view of the domain and problem setup; Right: contour plot of horizontal velocity u_1(m/s) inside the deformed domain at $t = 7s$.

structural grid nodes match the fluid mesh on the interface and there is no need for interpolation of parameters between the domains. Figure 3.4(right) shows the flow field inside the domain with structural deformation at the bottom, at $t = 7$s.

To evaluate the temporal accuracy of the method, the problem is solved using four different time step sizes ranging from 5×10^{-4} to 4×10^{-3}s. Reference results are obtained using a much smaller time step of $\Delta t = 5 \times 10^{-5}$s. The spatial grid is kept constant for the tests. The simulations are carried out from $t = 0$ until $t = 1$s, when the results for different time steps are saved and compared to the reference solution. The error is evaluated at every grid node and presented in both L_2 and L_∞ norms.

Figures 3.5 and 3.6 represent the variation of the relative error with time step size Δt in logarithmic scales. Figure 3.5 represents the error of fluid velocity and pressure inside the fluid domain, while Figure 3.6 represents the error of fluid pressure and structural deformation on the fluid-structure interface. Results demonstrate a clear second-order temporal accuracy for all the variables of interest in both L_2 and L_∞ norms.

3.4.3 Wave propagation in a 2-D deformable channel

This test case is to simulate the propagation of pressure waves inside a 2D straight channel with elastic walls. The problem is similar to blood flow inside large arteries and was studied, among others, in [10, 38–40]. The problem represents a strong added-mass effect, as the densities of the fluid and the structure are very similar and the domain is slender.

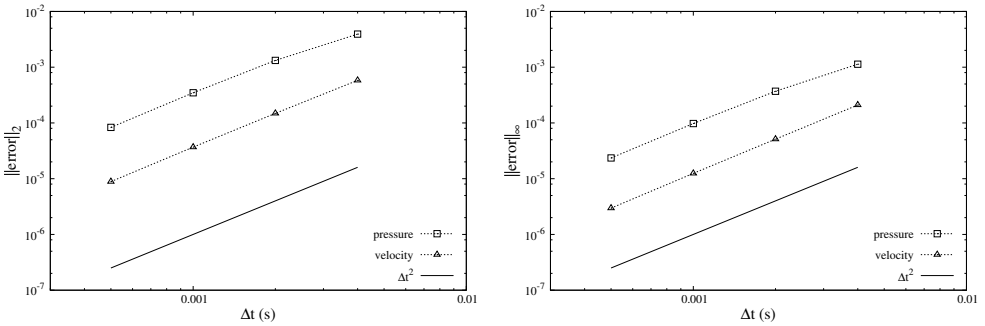

Figure 3.5: Variation of relative error inside the fluid domain by time step size, driven cavity with flexible bottom. Left: L_2 norm; Right: L_∞ norm.

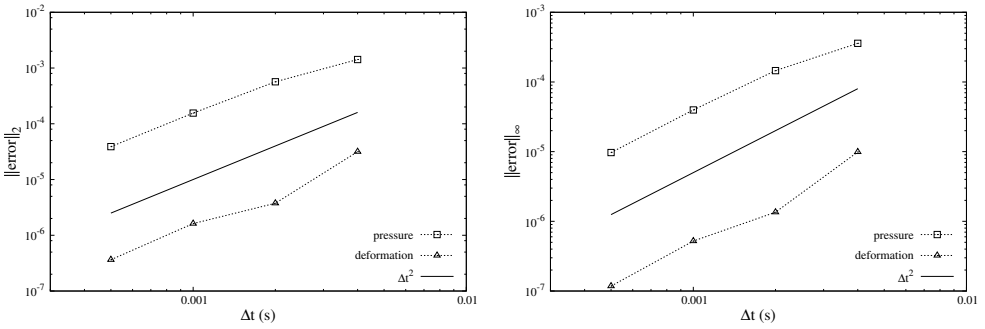

Figure 3.6: Variation of relative error on the fluid-structure interface by time step size, driven cavity with flexible bottom. Left: L_2 norm; Right: L_∞ norm.

The initial fluid domain is $[0, L] \times [0, R_0]$, where $L = 6$cm is the length of the channel and $R_0 = 0.5$cm is its height. The top boundary is an elastic wall interacting with the fluid, while the bottom boundary is a slip wall (axis of symmetry). Fluid pressure is specified at both inlet and outlet boundaries, while Neumann boundary condition is used for velocity. Pressure at the outlet boundary is set to zero while the inlet pressure represents a time-dependent pulse:

$$p_{inlet}(t) = \begin{cases} P_{pulse}\sin(\pi t / t_{pulse}) & 0 \le t \le t_{pulse} \\ 0 & t > t_{pulse} \end{cases}$$

where $P_{pulse} = 2000$Pa and $t_{pulse} = 0.005$s are, respectively, the amplitude and duration of the pressure pulse.

The deformable wall is modeled using the generalized string model [41]:

$$\rho_s h \frac{\partial^2 \mathbf{d}}{\partial t^2} - \frac{Eh}{2(1+\nu)} \frac{\partial^2 \mathbf{d}}{\partial x^2} + \frac{Eh}{1-\nu^2} \frac{\mathbf{d}}{R_0^2} + \alpha_0 \rho_s h \frac{\partial \mathbf{d}}{\partial t} - \frac{\alpha_1 Eh}{2(1+\nu)} \frac{\partial^3 \mathbf{d}}{\partial x^2 \partial t} = q(x,t) \quad (3.47)$$

where the solid density is $\rho_s = 1100\text{kg/m}^3$, the Young modulus $E = 7.5 \times 10^4 \text{N/m}^2$, the wall thickness $h = 0.1$cm, the Poisson ratio is $\nu = 0.5$, and the damping parameters $\alpha_0 = 1$ and $\alpha_1 = 0.001$. Fluid density and viscosity are $\rho_f = 1000\text{kg/m}^3$ and $\mu_f = 0.0035$Pa.s, respectively. The system is at rest at $t = 0$ and the simulations are carried out till $t = 0.015$s.

Three different spatial and temporal grid resolutions (table 3.1) are used to solve the problem, in order to ensure the grid independency of the results. Fluid mesh is refined near the top boundary (F-S interface) using a hyperbolic function. The boundary nodes for the fluid mesh coincide with the grid nodes for the solid domain, thus there is no need for further interpolations between the domains.

Table 3.1: Three grid resolutions used for deformable channel case.

Mesh name	No. of nodes		time step
	length	height	Δt (s)
coarse	100	20	2e-5
medium	150	30	1e-5
fine	200	40	5e-6

Figure 3.7 depicts the location of the interface at $t = 0.015$s, evaluated using the three grid resolutions mentioned in table 3.1. As seen in the figure 3.7, the results with medium and fine meshes are almost indistinguishable, which means the mesh convergence is achieved.

Figure 3.8 contains contour plots of pressure at three different instants, demonstrating the propagation of the wave inside the channel. The deformation of the channel wall (although small) is also visible in the figure. Despite the fluid being incompressible, the pressure wave propagates with a finite velocity, which is an important feature of this problem.

To evaluate the accuracy of the results and verify the presented numerical methodology, a comparison to other numerical results in [39, 40] has been carried out. Figure 3.9 shows the location of the interface at $t = 0.015$s evaluated with the medium mesh and $\Delta t = 1 \times 10^{-5}$s, together with the results from Fernandez et al. [39] and Li et al. [40]. As seen in the figure, our results agree fairly well with those of the other reports. There is a slight difference between the three sets of results which is

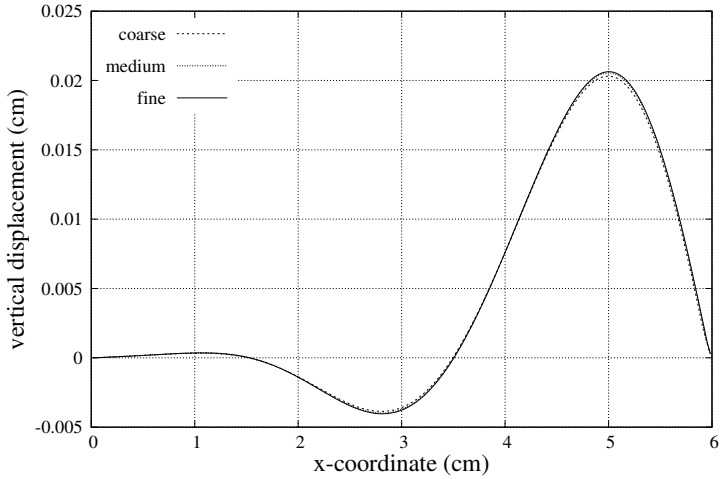

Figure 3.7: Displacement of the elastic boundary at $t = 0.015$s, evaluated with different grid resolutions.

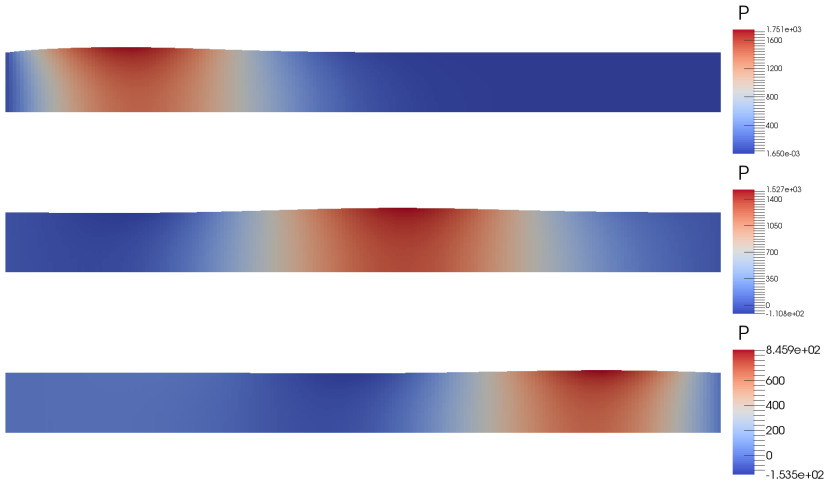

Figure 3.8: Propagation of pressure wave inside the deformable channel, contour plots of pressure at $t = 0.005$s, $t = 0.01$s and $t = 0.015$s.

acceptable considering they use different discretizations and FSI coupling techniques.

It is worth to mention that results in [39] are evaluated using a first-order method and a very small time step $\Delta t = 1 \times 10^{-6} s$. Results of simulations with different time step sizes (and different spatial mesh) are also reported in [39] to show the convergence to a limiting solution (similar to figure 3.7). These results show that convergence is achieved at $\Delta t = 1 \times 10^{-6}$s and for larger time steps there is a considerable discrepancy (see section 4.1 in [39]). However, our method provides time-step-independent results for $\Delta t = 1 \times 10^{-5}$s (figure 3.7). This fact highlights the advantage of using higher-order methods. Unfortunately in [40] the time step size for this test case is not mentioned.

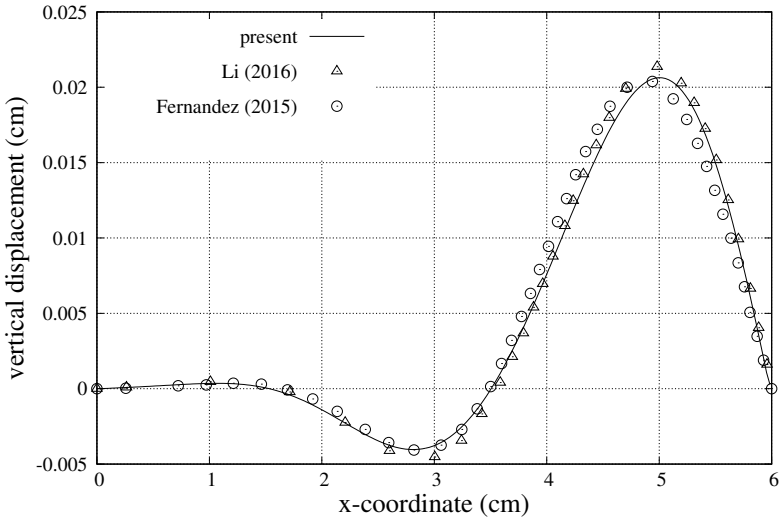

Figure 3.9: Displacement of the elastic boundary at $t = 0.015$s, comparison with numerical results at [39, 40].

To evaluate the temporal accuracy of the method, the problem was solved with four different time step sizes between 5×10^{-5} to 4×10^{-4}s. Reference results were generated using $\Delta t = 5 \times 10^{-6}$s, which is an order of magnitude smaller than the smallest test time step size. The spatial grid is kept constant for the analysis. The error for each time step size was evaluated and is presented in figures 3.10 and 3.11. These results demonstrate a clear second-order temporal accuracy for all the variables of interest, in both L_2 and L_∞ norms.

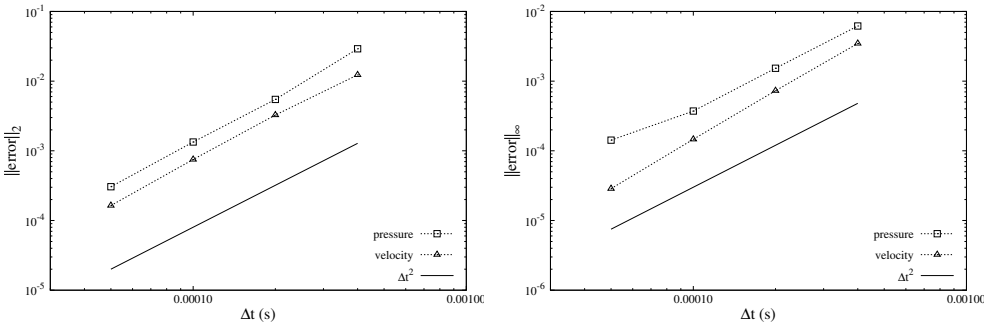

Figure 3.10: Variation of relative error in the fluid domain by time step size, elastic channel case. Left: L_2 norm; Right: L_∞ norm.

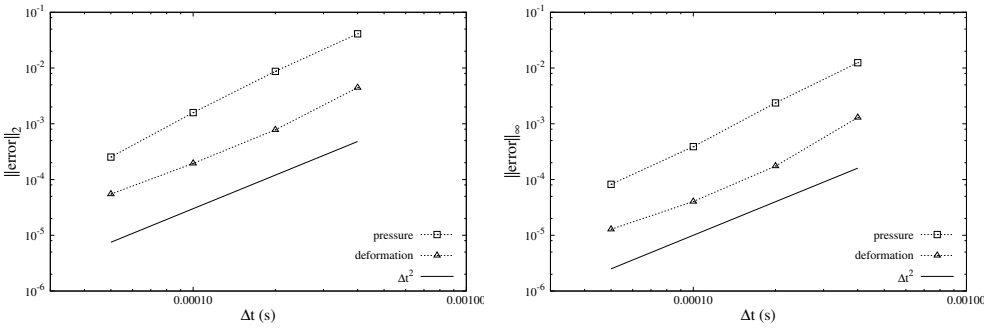

Figure 3.11: Variation of relative error on the fluid-structure interface by time step size, elastic channel case. Left: L_2 norm; Right: L_∞ norm.

3.4.4 Vortex-induced vibration of a circular cylinder

For this test case, the fluid flow over an elastically-mounted cylinder is solved in order to study the structural vibration due to the flow vortices. Vortex-induced vibration (VIV) is an important class of FSI problems with a wide range application. The cylinder is elastically mounted and it can move as a rigid-body around its reference position. Due to the vortex-shedding over the blunt body, the flow exerts an oscillating force on the cylinder, which causes it to vibrate. Generally, the vortex-shedding over an elastically-mounted cylinder occurs at the same frequency as for a fixed cylinder (the Strouhal frequency). However, there is a certain range of Reynolds number where the vortex-shedding frequency changes to match the natural structural frequency of the cylinder. This range of Reynolds number is called *lock-in* region as the vortex-

shedding no longer occurs at the Strouhal frequency, but at the natural frequency of the solid.

A series of VIV simulations are carried out to numerically reproduce the experimental results of Anagnostopoulos and Bearman [42]. A rectangular fluid domain is considered around a circular cylinder with a diameter D. The size of the domain is chosen based on previous experience of the authors and guidance from other VIV studies in the literature. Figure 3.12 depicts the layout of the domain and the problem setup. The flow enters the domain with a uniform velocity U_∞, while the pressure is set to zero at the outlet. For the sake of computational efficiency, the ALE formulation and the dynamic mesh is limited to a circular zone around the cylinder with a diameter of $6D$. The mesh is not moving at the rest of the domain and an Eulerian formulation is used. The cylinder motion is constrained to the cross-flow direction only, as per experiments.

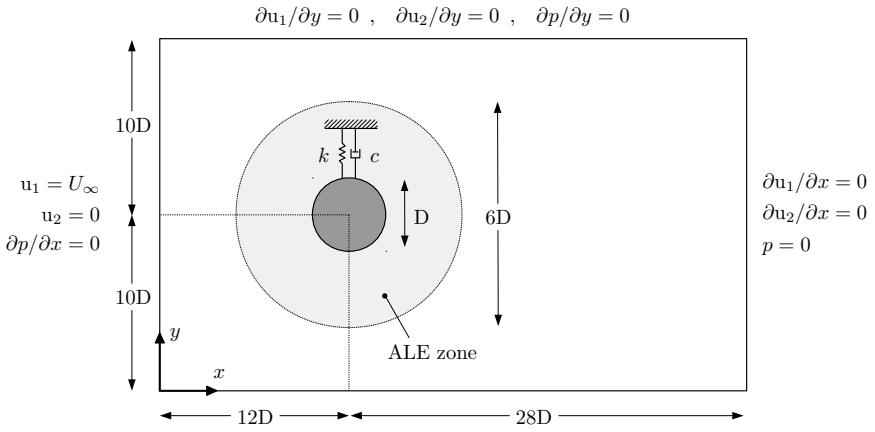

Figure 3.12: Schematic view of the domain and problem setup for the VIV test case.

The rigid-body motion of the cylinder is modeled as a system of spring and damper:

$$m\frac{\partial^2 \mathbf{d}}{\partial t^2} + c\frac{\partial \mathbf{d}}{\partial t} + k\mathbf{d} = q(t) \tag{3.48}$$

where $\mathbf{d} = [0, y, 0]^T$, y being the vertical location of the center of the cylinder. The cylinder mass is shown by m, c is the damping coefficient, k the spring stiffness, and $q(t)$ stands for the vertical component of the time-variant forces exerted on the cylinder by the fluid flow. The natural frequency of the vibration system is

$f_n = \frac{1}{2\pi}\sqrt{\frac{k}{m}}$. Table 3.2 shows the definition of the relevant non-dimensional numbers for this problem and their respective values. All non-dimensional numbers are equal to those of the experiments [42]. The Reynolds number varies between 90 to 140 and the associated reduced velocity between 5.01 to 7.80. The variable l in the definition of mass ratio is the length of the cylinder.

Table 3.2: Relevant non-dimensional numbers of the VIV problem and their value.

Name	Definition	Value
Reynolds (Re)	$\rho_f U_\infty D/\mu_f$	90-140
Reduced velocity (U_r)	$U_\infty/(f_n D)$	5.01-7.80
Mass ratio	$\rho_f D^2 l/2m$	0.00427
Damping ratio	$c/2\sqrt{km}$	0.0012

To assess the grid-independency of the results, three different grids are used to solve the problem at Re=110, which lies in the lock-in zone as seen later. Table 3.3 contains the information of the mesh and three representative values of the results, i.e. the normalized amplitude of the vibrations $A^* = y_{max}/D$, vortex-shedding frequency f, in non-dimensional form fD/U_∞, and the drag coefficient C_d.

Table 3.3: mesh-independency of VIV results at Re=110.

Mesh name	No. of cells	A^*	fD/U_∞	C_d
coarse	7195	0.417	0.165	1.43
medium	13685	0.408	0.163	1.53
fine	27091	0.410	0.163	1.56

Results in Table 3.3 show that the medium and fine grids yield similar results. We have used the medium mesh to conduct further simulations, for the sake of both accuracy and computational efficiency. Grid-independence study at one Reynolds number is sufficient for this test case because the range of studied Reynolds number is small ($90 \leq Re \leq 140$).

Figure 3.13 demonstrates contour plots for pressure inside the domain at two different instants while the cylinder is locked-in ($Re = 110$) and undergoes large-amplitude vibrations. The structure of the wakes and vortices could be seen at the rear of the cylinder. Results of the simulations for the range of Reynolds number in table 3.2 are presented in figure 3.14, along with experimental data from [42] and other numerical results from [21, 43, 44]. The compared results are the normalized amplitude of the vibrations (A^*), and the vortex-shedding frequency divided by the

natural frequency of the structure (f/f_n).

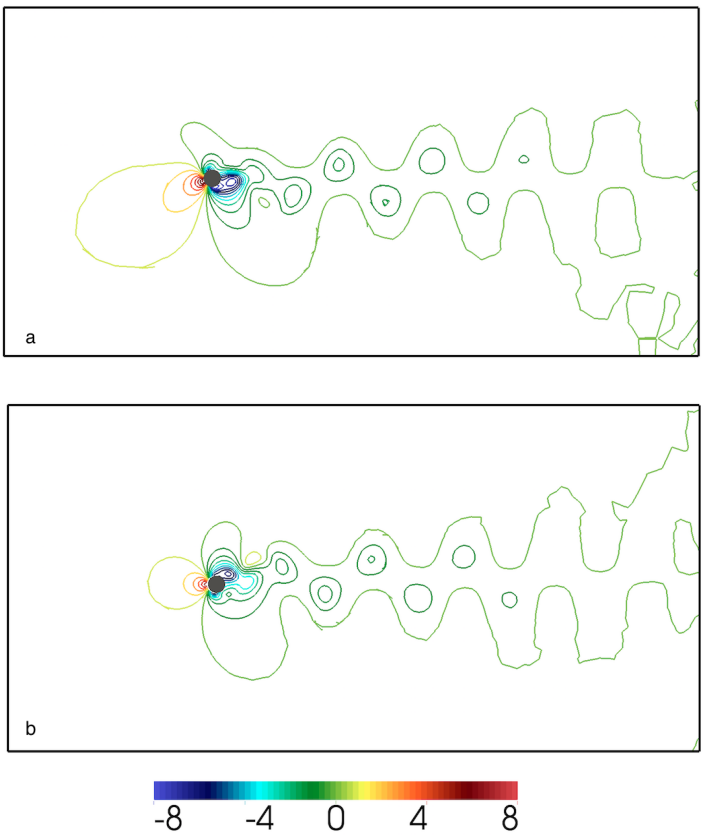

Figure 3.13: Contour plots of pressure (color legend in Pa) inside the domain at two instants while the cylinder undergoes large-amplitude vibration. a: cylinder approximately at the equilibrium point ($y = 0$); b: cylinder approximately at the maximum displacement ($y = -0.4D$)

It is seen in figure 3.14 that the lock-in phenomenon is well captured. For Reynolds numbers $Re < 95$ and $Re > 115$ the vortex shedding occurs at the Strouhal frequency (frequency of vortex-shedding over a fixed cylinder at the same Reynolds number). In this range of Reynolds number, the amplitude of the vibration is small. The lock-in region is at $95 < Re < 115$, where the amplitude of the vibration is significantly larger. In the lock-in zone, the vortex shedding no longer occurs at the Strouhal frequency,

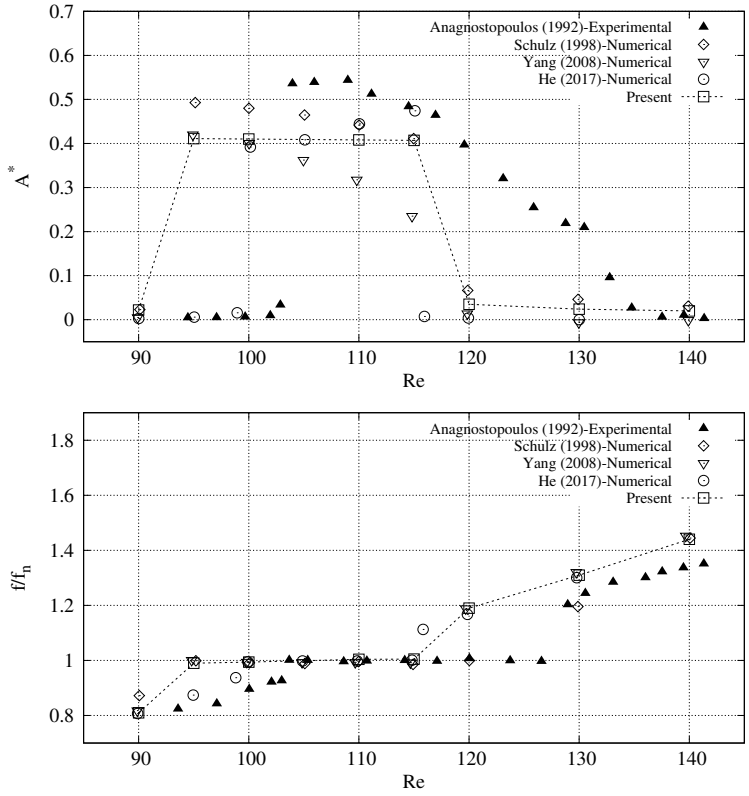

Figure 3.14: Comparison of the VIV simulation results against experimental and numerical results from the literature. Up: normalized amplitude of vibration; Down: vortex-shedding frequency divided by natural frequency of the cylinder.

but at the natural frequency of the cylinder.

Results in figure 3.14 agree fairly well with the experimental and numerical results from the literature. Different numerical results in the figure differ fairly amongst each other, which is due to the use of different numerical methods and complexity of the problem. Assessing the present results, it is seen that the predicted amplitude of vibration is smaller than the experiments but similar to other numerical results. Moreover, there is a slight shift of the location of the lock-in zone, i. e. numerical lock-in region starts and ends at lower Reynolds numbers than its experimental counterpart. This shift is seen in other numerical results as well [21, 43, 44]. The discrepancies may originate from the 3D effects in the experiments, as also noted

in [43]. In the experimental study, the authors mention that no end plates were used on the cylinder [42]. This would possibly introduce some 3-D effects that a 2-D simulation like the current work is not able to capture. Moreover, the Reynolds number is close to the region of transition to 3-D (transition to 3-D occurs at about Re=180 for a fixed cylinder). Therefore, it is possible that the Reynolds number locally exceeded the transition range and some 3-D effects were introduced.

Similar to the previous test cases, we have used four time step sizes, ranging from 8×10^{-3} to 1×10^{-3}s, and a much smaller time step of 1×10^{-4}s for the reference results. The variation of the error by time step size is presented in figures 3.15 and 3.16. These results are similar to the previous test cases and further confirm the second-order accuracy of the method.

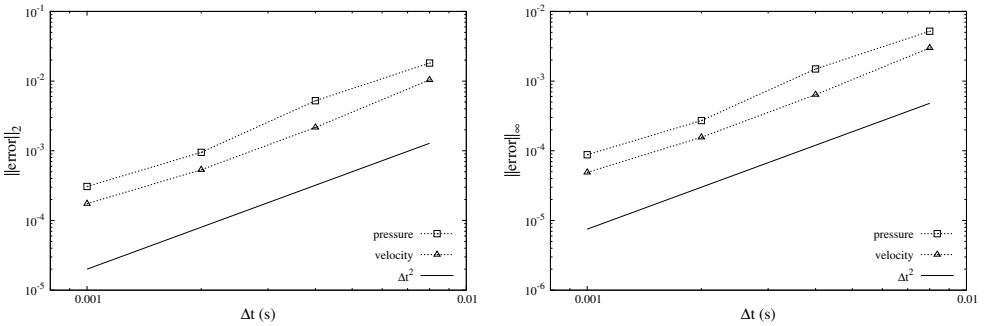

Figure 3.15: Variation of relative error in the fluid domain by time step size, VIV case. Left: L_2 norm; Right: L_∞ norm.

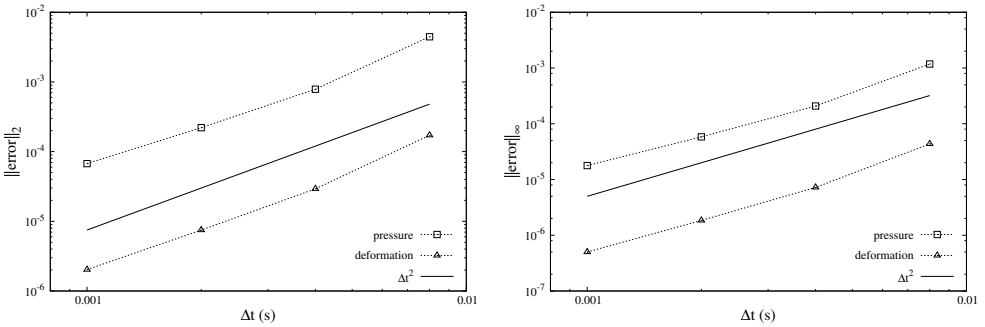

Figure 3.16: Variation of relative error on the fluid-structure interface by time step size, VIV case. Left: L_2 norm; Right: L_∞ norm.

3.5 Conclusions

A second-order semi-implicit method for partitioned solution of fluid-structure interaction problems is proposed. The method uses a second-order projection method to solve the fluid equations and also as a framework for the FSI coupling. The fluid pressure term is effectively segregated using the projection method and is strongly coupled to the structure via Newton iterations. Implicit treatment of the fluid pressure provides for the stability of the method for FSI problems with strong added-mass effect. The remaining fluid terms and the geometrical nonlinearities (moving mesh) are treated explicitly and thus evaluated only once per time step.

An ALE formulation with a conforming mesh technique is used to solve the fluid flow in a moving domain. A dynamic mesh technique based on radial basis function interpolation method is used to adapt the fluid mesh to the structural displacement. The geometrical terms arisen from the ALE formulation are evaluated with a second-order temporal accuracy. Consistent boundary conditions are developed for the intermediary fields encountered when solving the fluid equations with a projection method. Particular attention is paid to second-order accuracy of the fluid pressure up to the moving boundary.

Second-order accuracy of the method for fully coupled non-linear FSI problems is demonstrated through rigorous numerical tests. Three FSI test cases are considered, including internal flow contained by a deformable membrane, external flow over an elastically-mounted blunt body, and cavity flow with deformable bottom wall. Simulation results are validated against experimental and numerical results from the literature. Different time step sizes are used to solve the problems and the error is evaluated with respect to a reference numerical solution. Second-order temporal accuracy for all the variables of interest (fluid velocity and pressure, and structural displacement) is clearly demonstrated.

References

[1] 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*, 194:4506–4527, 2005.

[2] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196:1278–1293, 2007.

[3] Joris Degroote. Partitioned simulation of fluid-structure interaction. *Archives of Computational Methods in Engineering*, 20:185–238, 2013.

[4] Gene Hou, Jin Wang, and Anita Layton. Numerical methods for fluid-structure interaction - A review. *Communications in Computational Physics*, 12(2):337–377, 2012.

[5] Patrick Le Tallec and Jean Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190(24-25):3039–3067, 2001.

[6] Jean Frédéric Gerbeau and Marina Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:631–647, 2003.

[7] E Kuhl, S Hulshoff, and R De Borst. An arbitrary lagrangian eulerian finite-element approach for fluid–structure interaction phenomena. *International Journal for Numerical Methods in Engineering*, 57(1):117–142, 2003.

[8] Fabio Nobile and Christian Vergara. An effective fluid-structure interaction formulation for vascular dynamics by generalized robin conditions. *SIAM Journal on Scientific Computing*, 30(2):731–763, 2008.

[9] Santiago Badia, Fabio Nobile, and Christian Vergara. Fluid-structure partitioned procedures based on Robin transmission conditions. *Journal of Computational Physics*, 227(14):7027–7051, 2008.

[10] Miguel A Fernández, Jimmy Mullaert, and Marina Vidrascu. Explicit robin–neumann schemes for the coupling of incompressible fluids with thin-walled structures. *Computer Methods in Applied Mechanics and Engineering*, 267:566–593, 2013.

[11] 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. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.

[12] A. J. Chorin. Numerical solution of the Navier-Stokes equations. *Mathematics of Computation*, 22:745–762, 1968.

[13] M. Breuer, G. De Nayer, M. Münsch, T. Gallinger, and R. Wüchner. Fluid-structure interaction using a partitioned semi-implicit predictor-corrector coupling scheme for the application of large-eddy simulation. *Journal of Fluids and Structures*, 29:107–130, 2012.

[14] M. Astorino, F. Chouly, and M. A. Fernández. Robin based semi-implicit coupling in fluid-structure interaction: Stability analysis and numerics. *SIAM Journal on Scientific Computing*, 31(6):4041–4065, 2009.

[15] A. Naseri, O. Lehmkuhl, I. Gonzalez, and A. Oliva. Partitioned semi-implicit methods for simulation of biomechanical fluid-structure interaction problems. *Journal of Physics: Conference Series*, 745(3):032020, 2016.

[16] A. Naseri, O. Lehmkuhl, I. Gonzalez, E. Bartrons, C. D. Pérez-Segarra, and A. Oliva. A semi-implicit coupling technique for fluid–structure interaction problems with strong added-mass effect. *Journal of Fluids and Structures*, 80:94–112, 2018.

[17] I González, O Lehmkuhl, A Naseri, J Rigola, and A Oliva. Fluid-structure interaction of a reed type valve. In *International Compressor Engineering Conference*, page 2490. Purdue University Libraries, 2016.

[18] I. González, A. Naseri, J. Rigola, C. D. Pérez-Segarra, and A. Oliva. A fluid-structure interaction solver for the fluid flow through reed type valves. *IOP Conference Series: Materials Science and Engineering*, 232(1):012032, 2017.

[19] A Amani, A Naseri, C D Perez-Segarra, and A Oliva. A method for fluid-structure interaction problems with non-newtonian fluid. In *6th European Conference on Computational Mechanics (ECCM-ECFD 2018)*. ECCOMAS, Glasgow, UK, June 2018.

[20] Tao He. A cbs-based partitioned semi-implicit coupling algorithm for fluid–structure interaction using mcibc method. *Computer Methods in Applied Mechanics and Engineering*, 298:252–278, 2016.

[21] T. He, K. Zhang, and T. Wang. Ac-cbs-based partitioned semi-implicit coupling algorithm for fluid-structure interaction using stabilized second-order pressure scheme. *Communications in Computational Physics*, 21(5):1449–1474, 2017.

[22] Soyibou Sy and Cornel Marius Murea. A stable time advancing scheme for solving fluid–structure interaction problem at small structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 198(2):210–222, 2008.

[23] Fabio Nobile, Matteo Pozzoli, and Christian Vergara. Time accurate partitioned algorithms for the solution of fluid–structure interaction problems in haemodynamics. *Computers & Fluids*, 86:470–482, 2013.

[24] Jie Liu, Rajeev K. Jaiman, and Pardha S. Gurugubelli. A stable second-order scheme for fluid–structure interaction with strong added-mass effects. *Journal of Computational Physics*, 270:687–710, 2014.

[25] D. L. Brown, R. Cortez, and M. L. Minion. Accurate projection methods for the incompressible navier–stokes equations. *Journal of Computational Physics*, 168(2):464–499, 2001.

[26] S. Armfield and R. Street. An analysis and comparison of the time accuracy of fractional-step methods for the navier–stokes equations on staggered grids. *International Journal for Numerical Methods in Fluids*, 38(3):255–282, 2002.

[27] J. L. Guermond, P. Minev, and J. Shen. An overview of projection methods for incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 195(44-47):6011–6045, 2006.

[28] 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.

[29] Charbel Farhat, Kristoffer G. van der Zee, and Philippe Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195:1973–2001, 2006.

[30] Oyekola Oyekole, Catalin Trenchea, and Martina BukacÌ. A second-order in time approximation of fluid-structure interaction problem. *SIAM Journal on Numerical Analysis*, 56(1):590–613, 2018.

[31] J. Van Kan. A second-order accurate pressure-correction scheme for viscous incompressible flow. *SIAM Journal on Scientific and Statistical Computing*, 7(3):870–891, 1986.

[32] J. B. Bell, P. Colella, and H. M. Glaz. A second-order projection method for the incompressible navier-stokes equations. *Journal of Computational Physics*, 85(2):257–283, 1989.

[33] R Verzicco and P Orlandi. A finite-difference scheme for three-dimensional incompressible flows in cylindrical coordinates. *Journal of Computational Physics*, 123(2):402–414, 1996.

[34] John Kim and Parviz Moin. Application of a fractional-step method to incompressible navier-stokes equations. *Journal of Computational Physics*, 59(2):308–323, 1985.

[35] O. Estruch, O. Lehmkuhl, R. Borrell, C. D Pérez Segarra, and A. Oliva. A parallel radial basis function interpolation method for unstructured dynamic meshes. *Computers and Fluids*, 80:44–54, 2013.

[36] Holger Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4(1):389–396, 1995.

[37] U. Küttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43:61–72, 2008.

[38] L. Formaggia, J. F. Gerbeau, F. Nobile, and A. Quarteroni. On the coupling of 3D and 1D Navier-Stokes equations for flow problems in compliant vessels. *Computer Methods in Applied Mechanics and Engineering*, 191(6-7):561–582, 2001.

[39] Miguel A Fernández, Mikel Landajuela, and Marina Vidrascu. Fully decoupled time-marching schemes for incompressible fluid/thin-walled structure interaction. *Journal of Computational Physics*, 297:156–181, 2015.

[40] L. Li, W. D. Henshaw, J. W. Banks, D. W. Schwendeman, and A. Main. A stable partitioned FSI algorithm for incompressible flow and deforming beams. *Journal of Computational Physics*, 312:272–306, 2016.

[41] Alfio Quarteroni, Massimiliano Tuveri, and Alessandro Veneziani. Computational vascular fluid dynamics: Problems, models and methods. *Computing and Visualization in Science*, 2:163–197, 2000.

[42] P Anagnostopoulos and PW Bearman. Response characteristics of a vortex-excited cylinder at low reynolds numbers. *Journal of Fluids and Structures*, 6(1):39–50, 1992.

[43] Karl W Schulz and Yannis Kallinderis. Unsteady flow structure interaction for incompressible flows using deformable hybrid grids. *Journal of Computational Physics*, 143(2):569–597, 1998.

[44] J Yang, S Preidikman, and E Balaras. A strongly coupled, embedded-boundary method for fluid–structure interactions of elastically mounted rigid bodies. *Journal of Fluids and Structures*, 24(2):167–182, 2008.

4

A scalable framework for solution of FSI problems

Main content of this chapter is being considered for publication in:

A. Naseri, A. Totounferoush, I. Gonzalez, M. Mehl and C. D. Pérez-Segarra. A Scalable Framework for Partitioned Solution of Fluid-Structure Interaction Problems. *preprint*.

Abstract. In this chapter, we present a scalable and efficient parallel framework for the partitioned solution of FSI problems through multi–code coupling. Two instances of an in–house parallel software, *TermoFluids*, are used to solve the fluid and the structural sub–problems, coupled together on the interface via the *preCICE* coupling library. For fluid flow, the ALE form of the Navier-Stokes equations is solved on an unstructured grid using a finite–volume discretization and second–order numerical schemes. A parallel dynamic mesh method for unstructured meshes is used to track the moving boundary. For the structural problem, the nonlinear elastodynamics equations are solved on an unstructured grid using a finite–volume method and second–order numerical schemes. A semi–implicit FSI coupling method is used which segregates the fluid pressure term and couples it strongly to the structure, while the remaining fluid terms and the geometrical nonlinearities are only loosely coupled. A robust and advanced multi–vector quasi–Newton method is used for the coupling iterations between the solvers. Both fluid and structural solvers use distributed–memory parallelism. The intra–solver communication required for data update in the solution process is carried out using non–blocking point–to–point communicators. The inter–code communication is fully parallel and point–to–point, avoiding any central communication unit. Inside each single–physics solver, the load is balanced by dividing the computational domain into fairly equal blocks for each process. Additionally, a load balancing model is used at the inter–code level to minimize the overall idle time of the processes. Practical test cases in the context of biological flow (blood flow inside deformable vessels) are studied and the scalability of the coupled solver is evaluated. Results confirmed a very good scalability for up to 3920 cores.

4.1 Background

In a partitioned approach to solve FSI problems, separate solvers are used for fluid and structural sub-problems and a coupling technique is adopted to account for the interaction of the domains. One of the big advantages of the partitioned methods is the possibility to use the most adapted and well-validated numerical methods for each sub-problem. Moreover, it allows using previously developed and optimized fluid and structural solver codes, thus saving excessive software development effort [1,2]. However, the partitioned approach introduces a new challenge to the problem which is the coupling between the separate solvers. This challenge has two different aspects. One aspect concerns the methodology, i.e., coupling distinct sets of (discretized) partial differential equations and ensuring the physical equilibrium conditions on the interface. The second aspect is related to the implementation, i.e., coupling two parallel codes with different modules and structures, with the aim of achieving an efficient and scalable overall software.

Modern scientific and engineering problems are often very complex and require a huge computational effort. Therefore, any simulation software must be able to efficiently run on massively parallel computers. The parallel efficiency of a solver code is crucial in order to be able to use the available resources adequately and perform a complex calculation. During the recent years, efficient parallel codes have been developed for many single-physics problems, particularly fluid and structural systems. While the monolithic approach to solve FSI problems requires developing a new solver and implementing a software, following a partitioned approach creates the opportunity to exploit the previously-developed efficient codes for FSI simulations. As far as single-physics simulation codes are concerned, a great advancement has been achieved in parallel efficiency. Nevertheless, using efficient single-physics codes does not automatically guarantee achieving a good parallel efficiency for a coupled multi-physics simulation. Multi-code coupling introduces several new challenges. One particular difficulty is the data exchange between separate codes which often use different data structures and could even be written in different languages. An efficiently parallel mechanism for data exchange between the codes is crucial for achieving parallel efficiency on the coupled framework. Moreover, by coupling two parallel codes, a new level of load balancing is introduced to the problem, as each code would be responsible for a different amount of calculations on a different number of CPUs.

Recent efforts have been made to develop efficient multi-physics simulation codes (particularly for FSI problems) using either monolithic [3] or partitioned [4,5] approaches. In this work, we focus on the partitioned approach in order to create a scalable and efficient framework using existing single–physics solver codes.

In our multi-code coupled framework, the communication between the separate codes is managed by using a communication library. The communication library receives the data from each code via an *adapter* and contains functions to facilitate the exchange of data. It also accelerates the coupling iterations between the solvers in the strongly–coupled configuration. The adapter is used to readjust the data structure when transferring data from one solver, through the communication library, to the other solver. This approach leads to a robust and powerful scheme to couple different codes for multi-physics simulations. An immediate advantage of this method is that one or both of the single-physics solvers could be replaced

by other codes with relatively small changes in the code, which would be often limited to the adapter. Examples of such communication libraries can be found in [4, 6–8]. Examples of multi-physics simulation software using communication libraries can be found in [4, 9]. There are other examples in the literature where the data exchange is handled directly and the communication functions are included in the single-physics codes themselves. This approach is shown to be efficient when two instances of the same code are coupled (e.g. [5]). However it is not as robust and powerful as the first approach, especially in cases where two different codes are coupled.

At the coupling methodology level, partitioned methods are generally divided into explicit (or loosely coupled) and implicit (or strongly coupled) schemes. Explicit methods solve the fluid and structural equations only once per time step, using data from the previous solution of the partner solver. Therefore, explicit methods do not satisfy the exact equilibrium conditions on the interface, which causes instability issues in a range of FSI problems (the so-called *added-mass* instability). The added-mass instability is particularly strong in FSI problems with incompressible flow, a slender interface, and similar densities of fluid and solid [10, 11]. Implicit methods, on the other hand, enforce the equilibrium condition on the interface through coupling iterations between fluid and structural solvers. These methods are stable for problems with strong added-mass effect, however, their computational cost is generally high due to the repetitive solution of the governing equations at each time step [1, 2]. In the recently introduced semi-implicit coupling approach [12–15], the fluid pressure term is segregated and strongly coupled to the structure, while the remaining fluid terms are only loosely coupled. Strong coupling of the fluid pressure and structural deformation eliminates the added-mass instability issue, while loose coupling of the remaining fluid terms helps avoiding excessive computational cost [12, 14].

In this work, two instances of a parallel in-house code, *TermoFluids* [16], are used to solve the fluid and the structural problem. TermoFluids is a robust general-purpose software for fluid and structural problems, using state-of-the-art methods for turbulent flow [17, 18], multiphase flow [19, 20] and complex thermal systems [21, 22], with a high computational efficiency and parallel scalability [23–25]. It presents a conservative discretization of the governing equations on unstructured grids based on a finite-volume method. It is also equipped with dynamic-mesh schemes to track the moving boundary. The coupling of the codes is carried out using the *preCICE* coupling library [8]. The preCICE library provides communication, data mapping and equation coupling for surface coupled multi-physics applications in a modular manner. It uses a fully parallel point-to-point communication, advanced quasi-Newton iterative coupling scheme and various advanced mapping methods (both consistent and conservative). A semi-implicit FSI coupling approach proposed in [14] is applied, which effectively segregates the fluid pressure term and couples it strongly to the structure. The remaining fluid terms and the geometrical non-linearities are treated explicitly, reducing the computational cost of the numerical solution. Practical test cases in the context of biological flow (flow inside deformable vessels) are studied and the scalability of the overall framework is evaluated.

The remainder of this chapter is organized as follows. In Section 2, the governing equations for each sub-problem and the coupling conditions are presented. Section 3 describes the proposed numerical methods. Section 4 presents the parallelization method for each single-physics solver, as well as the inter-code communications. Numerical tests are presented in

Section 5, while Section 6 summarizes and concludes the chapter.

4.2 Governing Equations

In this section, we present the fluid and structural governing equations and the coupling conditions on their common interface. The fluid and the structural domain are referred to as $\Omega_f(t) \subset \mathbb{R}^3 \times (0,T)$ and $\Omega_s(t) \subset \mathbb{R}^3 \times (0,T)$, respectively, where $t \in (0,T)$ denotes time. The fluid-structure interface is the common boundary of the domains, denoted by $\Gamma(t) = \partial\Omega_f(t) \cap \partial\Omega_s(t)$. An Arbitrary Lagrangian-Eulerian (ALE) formulation together with a conforming mesh technique is used to solve the fluid flow in a moving domain. A Lagrangian formulation is used for the structural equations.

The unsteady flow of an incompressible viscous fluid is governed by the Navier-Stokes equations. An ALE formulation of these equations in a moving domain is given by:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{c} \cdot \nabla \mathbf{u} = \frac{1}{\rho_f} \nabla \cdot \boldsymbol{\sigma}_f \tag{4.1}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{4.2}$$

where $\mathbf{u}$ is the fluid velocity and ρ_f the fluid density. Vector $\mathbf{c}$ is the ALE convective velocity $\mathbf{c} = \mathbf{u} - \mathbf{w}$, which is the fluid velocity relative to a domain moving with a velocity $\mathbf{w}$. The stress tensor $\boldsymbol{\sigma}_f$ is defined for an incompressible Newtonian fluid as:

$$\boldsymbol{\sigma}_f = -p\mathbf{I} + \mu_f(\nabla \mathbf{u} + \nabla \mathbf{u}^T) \tag{4.3}$$

where p is the fluid pressure, $\mathbf{I}$ the unit tensor and μ_f is the dynamic viscosity of the fluid.

The structural domain is governed by the conservation laws of mass and momentum, whose Lagrangian form is given by:

$$\rho_s^0 = \rho_s J \tag{4.4}$$

$$\frac{\partial}{\partial t}\left(\rho_s \frac{\partial \mathbf{d}}{\partial t}\right) = \nabla \cdot \boldsymbol{\sigma}_s \tag{4.5}$$

where superscript 0 refers to the reference material configuration (undeformed) of the body, ρ_s is the structural density and $\mathbf{d}$ is the displacement from the reference configuration. The tensor $\boldsymbol{\sigma}_s$ is the Cauchy stress tensor, which can be related to the displacement field by the hyperelastic constitutive model of Saint Venant-Kirchhoff:

$$\boldsymbol{\sigma}_s = \frac{\mathbf{B}}{2J}[2\mu_s(\mathbf{B} - \mathbf{I}) + \lambda_s \mathrm{tr}(\mathbf{B} - \mathbf{I})] \tag{4.6}$$

where $\mathbf{B}$ is the left Cauchy-Green deformation tensor $\mathbf{B} = \mathbf{F} \cdot \mathbf{F}^T$, and μ_s and λ_s are the Lamé's parameters. The material deformation tensor $\mathbf{F}$ is evaluated as $\mathbf{F} = \mathbf{I} + \nabla \mathbf{d}$ and its determinant is shown by $J = \det(\mathbf{F})$.

The physical equilibrium on the fluid-solid common boundary (kinematic and dynamic equilibrium) constitutes the coupling conditions on the interface. For a non-slip type interface they read:

$$\mathbf{u}_\Gamma = \frac{\partial \mathbf{d}_\Gamma}{\partial t} \tag{4.7}$$

$$\boldsymbol{\sigma}_s \mathbf{n}_\Gamma = \boldsymbol{\sigma}_f \mathbf{n}_\Gamma \tag{4.8}$$

for any $\mathbf{x} \in \Gamma$, where $\mathbf{n}_\Gamma$ is the unit normal vector on the interface.

4.3 Numerical Methods

In this section we present the numerical methods for discretization and solution of the single-physics problems, as well as the coupling method. Throughout the development of our coupled framework and the single-physics solvers, an emphasis was made on making the software simple, modular and, as much as possible, matrix-free. Moreover, the fluid solver was mainly developed for turbulent flow simulations. Therefore, many aspects of the discretization and the numerical methods correspond to the particular considerations of turbulent flows.

4.3.1 Fluid solver

For fluid flow, a fractional-step projection method along with an explicit time advancement is used to solve the velocity-pressure coupling of the momentum equation. This leads to a three step solution of the fluid governing equations from time step n to $n+1$, with a time increment of Δt

$$\mathbf{u}^p = \mathbf{u}^n - \Delta t[\frac{3}{2}(\mathbf{c}^n \cdot \nabla \mathbf{u}^n - \frac{\mu_f}{\rho_f}\Delta \mathbf{u}^n) - \frac{1}{2}(\mathbf{c}^{n-1} \cdot \nabla \mathbf{u}^{n-1} - \frac{\mu_f}{\rho_f}\Delta \mathbf{u}^{n-1})] \tag{4.9}$$

$$\frac{\Delta t}{\rho_\mathrm{f}} \Delta p^{\mathrm{n+1}} = \nabla \cdot \mathbf{u}^{\mathrm{p}} \tag{4.10}$$

$$\mathbf{u}^{\mathrm{n+1}} = \mathbf{u}^{\mathrm{p}} - \frac{\Delta t}{\rho_\mathrm{f}} \nabla p^{\mathrm{n+1}} \tag{4.11}$$

for $\mathbf{x} \in \Omega_f^{n+1}$, where $\mathbf{u}^p$ is a predicted velocity field which does not satisfy the incompressibility condition (Eq. (4.2)). This intermediate velocity field is then projected onto a divergence-free field through the correction at Eq. (4.11). An explicit Adams-Bashforth method is used for the convective and diffusive terms in Eq. (4.9). Using an explicit method is particularly preferred in turbulent flow simulations where small time steps are indispensable. From a computational point of view, it avoids solving a nonlinear system for the fluid velocity field. It also offers advantages in parallelization as only the information from the previous time step is required (only one episode of data update between nodes is required at each time step).

A finite-volume method is used for the spatial discretization of the fluid equations on a collocated, unstructured mesh with second-order symmetry-preserving schemes. Symmetry-preserving schemes conserve the kinetic energy of the flow at the discrete level which is crucially important in turbulent flow simulations [26,27]. A Jacobi–preconditioned conjugate

gradient solver is used to solve the Poisson equation for pressure. More details of the numerical methods for fluid flow equations could be found in [27,28].

We use a conforming mesh technique to track the moving boundary, thus the fluid mesh needs to move in order to adapt to the new location of the interface. The translated mesh and the evaluated domain velocity, $\mathbf{w}$, must satisfy the *Geometric Conservation Law* (GCL) [29,30]. The GCL guarantees that no volume is lost while moving the grid, and a constant field is preserved by the ALE scheme. For any control volume (CV) in the fluid domain, the GCL is stated as:

$$\frac{\partial v}{\partial t} - \int_s \mathbf{w} \cdot \mathrm{d}\mathbf{A} = 0 \tag{4.12}$$

where v and s stand for the volume and the boundary surface of the CV, respectively, and $\mathbf{A}$ is the area vector pointing outward.

A parallel moving mesh technique based on the radial basis function interpolation method [31] is used to move the fluid grid in accordance to the new location of the interface and define the discretized fluid domain at the new time step Ω_f^{n+1}. The method uses the known displacement on the interface to evaluate an interpolated value for the interior vertices of the fluid grid. A great advantage of this method is that it does not need the connectivity of the mesh elements and can be applied to both structured and unstructured grids. Moreover, it only requires solving a linear system of equations whose size is limited to the number of vertices on the fluid-solid interface. A detailed description of the moving mesh method could be found in [14,31].

After the fluid mesh is moved, the domain velocity is evaluated at the surfaces of each control volume. We evaluate the surface velocities based on the GCL law in order to exactly satisfy it. Time rate of change of volume of a CV is equal to the sum of volumes swept by its each face. In this work we evaluate the domain velocity at each face, $\mathbf{w}_{face}$, based on the volume swept by that face. With a second-order backward discretization it reads:

$$\mathbf{w}_{face}^{n+1} = \frac{3}{2}\left(\frac{\delta v}{A\Delta t}\mathbf{n}\right)^{n+1} - \frac{1}{2}\left(\frac{\delta v}{A\Delta t}\mathbf{n}\right)^{n} \tag{4.13}$$

where A is the surface area, $\mathbf{n}$ the unit normal vector of the face, Δt the time step and δv is the volume swept by the face at one time step. A more detailed description of the evaluation of the domain velocity field and satisfaction of the geometric conservation law could be found in chapter 3 and Appendix A.

4.3.2 Solid solver

Aiming at flow-induced deformations and oscillations of the structure, the solid equations are solved using an implicit time integration, where both the inertial and surface forces in Eq. (4.5) are evaluated at the current time instant t^{n+1}. A cell-centered finite volume formulation with a total Lagrangian approach is used for the spatial discretization. The momentum equation is integrated on the undeformed configuration. After applying the Gauss theorem on the stress divergence and relating the undeformed and current area vectors with Nanson's formula, the momentum balance can be written as:

$$\int_{v^0} \rho_s^0 \frac{\partial^2 \mathbf{d}}{\partial t^2}\, dv^0 = \int_{s^0} \boldsymbol{\sigma}_s (J\mathbf{F}^{-T}\mathbf{n}^0)\, ds^0 \tag{4.14}$$

The acceleration of the inertial term is computed according to the trapezoidal rule, as a function of the new displacement $\mathbf{d}^{n+1}$ and past kinematics in the following manner:

$$\frac{\partial^2 \mathbf{d}}{\partial t^2} = \frac{4}{\Delta t^2}\mathbf{d}^{n+1} - \frac{4}{\Delta t^2}\mathbf{d}^n - \frac{4}{\Delta t}\left(\frac{\partial \mathbf{d}}{\partial t}\right)^n - \left(\frac{\partial^2 \mathbf{d}}{\partial t^2}\right)^n \tag{4.15}$$

The dependencies between different directions of the displacement and the geometrical and material non-linearities found in the right-hand side of Eq. (4.14) are deferred to the source term of the system. Therefore, outer fixed-point iterations are needed to obtain a converged solution. This segregated algorithm is typically adopted within finite volume procedures for structures due to its robustness and low memory requirements [32]. It divides the surface force into an implicit diffusive component and a deferred correction as:

$$\begin{aligned}\int_{v^0} \rho_s^0 \frac{\partial^2 \mathbf{d}}{\partial t^2}\, dv^0 = &\int_{s^0} \mathcal{K}_{\text{imp}} (\nabla \mathbf{d})\mathbf{n}^0\, ds^0 \\ &+ \int_{s^0} \boldsymbol{\sigma}_s (J\mathbf{F}^{-T}\mathbf{n}^0)\, ds^0 - \int_{s^0} \mathcal{K}_{\text{imp}} (\nabla \mathbf{d})\mathbf{n}^0\, ds^0\end{aligned} \tag{4.16}$$

The diffusive term is approximated by a central difference scheme with a non-orthogonal correction, presented as the over-relaxed scheme in [33]. The convergence of the iterative process is improved by selecting an optimal value of $\mathcal{K}_{\text{imp}} = (2\mu_s + \lambda_s)$, and using an Aitken Δ^2 acceleration technique for multidimensional problems [34].

Similar to the methods in [35,36], the gradient of the displacement (and hence, the strain and stress tensors) are evaluated directly on the cell face centers. To do so, the gradient is decomposed into the normal and the tangential derivatives to the face surface, and a specific numerical scheme is used for each one. The former uses the same central difference scheme dedicated to the implicit term whereas the latter follows the surface Gauss theorem to express the tangential derivative as a function of the displacement in the face vertices. Therefore, a method to interpolate the displacement from the cell centers to the vertices of the grid has to be defined. The second-order accurate interpolation technique described in [37] is used for this purpose. Additional details on the numerical method for solid mechanics are presented in [38].

4.3.3 Fluid-structure coupling

A Dirichlet-Neumann (DN) domain decomposition method is used to solve the coupled FSI problem. In the DN decomposition, the fluid equations are solved for a known location of the interface and the kinematic equilibrium condition (Eq. (4.7)) is used as a Dirichlet boundary condition for fluid flow. The structural equations are solved for a known traction on the interface, thus subject to a Neumann boundary condition derived from the dynamic equilibrium condition (Eq. (4.8)). A great advantage of the DN decomposition method is its consistency with the most common solvers for fluid and structural equations.

A semi-implicit coupling method similar to the ones proposed in [14, 15] is followed in this work. The principal idea is to segregate the fluid pressure term and couple it strongly to the structure. It is argued that fluid pressure term is the main contributor to the added mass effect and coupling it loosely will cause numerical instabilities [10, 12]. Segregation of the fluid pressure is naturally achieved by using a projection method to solve the fluid equations. Therefore, only the Poisson equation for pressure in the discretized fluid equations (Eq. (4.10)) is strongly coupled to the structure. The remaining fluid equations (Eq. (4.9) and (4.11)) are solved only once per time step. The geometrical nonlinearities are also treated explicitly and the fluid mesh is only moved once per time step. The location of the interface for each time step is evaluated as an extrapolation of the location in the previous instants. This semi-implicit coupling strategy is concisely described in Algorithm 1. For a detailed description of the method see [14, 15]. Therefore, the coupled problem on the interface could be represented as:

$$\mathcal{S} \circ \mathcal{P}(\mathbf{d}_\Gamma) = \mathbf{d}_\Gamma \tag{4.17}$$

where $\mathcal{S}$ represents the structural block solver as a function of the traction on the interface $\mathcal{S} = \mathcal{S}(\boldsymbol{\sigma}_\Gamma)$ and $\mathcal{P}$ is the part of the fluid solver that is strongly coupled to the structure (the pressure equation). Both $\mathcal{S}$ and $\mathcal{P}$ are nonlinear operators. Also note that the discretized operators $\mathcal{S}$ and $\mathcal{P}$, as well as the interface deformation field $\mathbf{d}_\Gamma$, are in the current time step t^{n+1}.

Algorithm 1 Semi-implicit FSI coupling method

start $t = t^{n+1}$

1: Predict the location of the interface by extrapolating from previous time steps.
2: Move the mesh and evaluate the surface velocities.
3: Solve the fluid ALE convective-diffusive equation for the predicted velocity field (Eq. (4.9)).

4: **while** coupling not converged **do** ▷ $\mathcal{S} \circ \mathcal{P}(\mathbf{d}_\Gamma) = \mathbf{d}_\Gamma$
5: Solve the Poisson's equation for pressure (Eq. (4.10)).
6: Solve the structural equations for the deformation.
7: Update boundary values for predicted velocity using the new solid deformation.
8: **end while**

9: Evaluate the corrected velocity field using converged pressure field (Eq. (4.11)).
10: Apply the boundary condition on the corrected velocity using converged deformation.

The coupled interface problem in Eq. (4.17) needs to be solved at each time step similar to the interface problem in a fully implicit partitioned method. Fixed-point iterations with Aitken's relaxation (e.g. [1, 5, 39]) or Newton-based methods (e.g. [40–42]) are commonly used to solve

the interface problem. The execution of the fluid and structural solvers could be carried out sequentially (staggered solution) or simultaneously (parallel solution). In the staggered method, the fluid and structural systems are solved sequentially (one after another), corresponding to a Gauss-Seidel-type problem. On the other hand, in simultaneous or parallel solution, the systems are solved at the same time (using different processors), corresponding to a Jacobi-type problem. Staggered methods are commonly used, although they have a very limited parallel efficiency due to a severe load imbalance [43, 44]. Although fixed-point iterations with Aitken's relaxation are seen to be efficient and robust in a staggered solution (see e.g. [39]), they show poor performance in a parallel (simultaneous) solution method. On the other hand, parallel execution together with a quasi-Newton method results in a very efficient and robust method [43, 44].

In this work, we take a parallel solution approach where the fluid and structural systems are solved at the same time (on different processors) using inputs from the previous iteration of the partner solver. One iteration of this Jacobi-like fixed-point problem can be written in matrix-like notation as:

$$\begin{pmatrix} 0 & \mathcal{S} \\ \mathcal{P} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{d}_\Gamma^k \\ \boldsymbol{\sigma}_\Gamma^k \end{pmatrix} = \begin{pmatrix} \widetilde{\mathbf{d}}_\Gamma^{k+1} \\ \widetilde{\boldsymbol{\sigma}}_\Gamma^{k+1} \end{pmatrix} \tag{4.18}$$

where k indicates the iteration count in the current time step $t^n \rightarrow t^{n+1}$, and the tilde sign in $\widetilde{\mathbf{d}}_\Gamma^{k+1}$ and $\widetilde{\boldsymbol{\sigma}}_\Gamma^{k+1}$ means these new values are yet to be modified in a subsequent Newton step. In order to describe the quasi-Newton method, we denote the vector form of the underlying fixed-point interface problem of Eq. (4.18) as:

$$\mathcal{H}(\mathbf{x}) = \mathbf{x} \tag{4.19}$$

where $\mathbf{x} = \begin{pmatrix} \mathbf{d}_\Gamma \\ \boldsymbol{\sigma}_\Gamma \end{pmatrix}$. To solve Eq. (4.19), we accelerate the fixed-point iteration (Eq. (4.18)) by a subsequent Newton step:

$$\mathbf{x}^{k+1} = \mathcal{H}(\mathbf{x}^k) - \mathcal{J}^{-1}\mathbf{R}(\mathbf{x}^k) \tag{4.20}$$

where k denotes the iteration count, $\mathbf{R}$ is the residual function $\mathbf{R}(\mathbf{x}^k) = \mathcal{H}(\mathbf{x}^k) - \mathbf{x}^k$, and $\mathcal{J}^{-1}$ is the inverse Jacobian of $\mathbf{R}$. Since calculation of the inverse Jacobian is not feasible, it is approximated based on secant equation:

$$\hat{\mathcal{J}}_k^{-1}\mathcal{V}_k = \mathcal{W}_k \tag{4.21}$$

in which the hat sign indicates the approximation and $\mathcal{V}_k$ and $\mathcal{W}_k$ are two vectors which include increments of $\mathbf{x}$ and the residual $\mathbf{R}$ in the previous iterations within the same time step:

$$\mathcal{W}_k = [\Delta\mathbf{x}^k, \Delta\mathbf{x}^{k-1}, ..., \Delta\mathbf{x}^1] \tag{4.22}$$

$$\mathcal{V}_k = [\Delta\mathbf{R}^k, \Delta\mathbf{R}^{k-1}, ..., \Delta\mathbf{R}^1] \tag{4.23}$$

In this work, we use a multi-vector quasi-Newton method (MVJ) as proposed in [8, 42]. In this approach, instead of minimizing the Forbenius norm of the Jacobian, the distance between

the current $\hat{\mathcal{J}}_k^{-1}$ and the previous time step approximated Jacobian $\hat{\mathcal{J}}^{-1,(n)}$ is minimized ($min(||\hat{\mathcal{J}}_k^{-1} - \hat{\mathcal{J}}^{-1,(n)}||)$. This results in the following approximation for the Jacobian inverse:

$$\hat{\mathcal{J}}_k^{-1} = \hat{\mathcal{J}}^{-1,(n)}(\mathcal{W}_k - \hat{\mathcal{J}}^{-1,(n)}\mathcal{V}_k)(\mathcal{V}_k^T\mathcal{V}_k)^{-1}\mathcal{V}_k^T \tag{4.24}$$

where it should be noted that k indicates the iteration number at the current time step, while n refers to the previous time step t^n. The estimated Jacobian can be used to find the increment of $\mathbf{x}$. This process must be repeated until the convergence criterion is met and then we can move to the next time step. This quasi–Newton method is part of the preCICE coupling library and can be consulted in details in [8].

4.4 Parallelization

The current framework is based on multi-code coupling of separate solvers for the fluid and the structural domain. Therefore, the parallelization aspects can be considered in two levels, namely the parallelization of each single-physics solver, and the inter-code communication level.

4.4.1 Single-physics solvers

Two instances of the same software are used for the single-physics sub-problem domains in this work. Therefore, the fluid and the structural solver share the same parallelization methods and programming model which are described in this chapter.

The overall computational efficiency of any code depends on both its sequential and parallel performance. Computational fluid dynamics and solid mechanics are memory-bounded applications, meaning that the sequential performance of the solver is limited by the cost of fetching data from the memory rather than the cost of computations on the CPU. This is a characteristic feature of these applications since their discretized systems of equations are represented by sparse matrices and the computations have low arithmetic intensity. Therefore, the single–core performance of the code in this work is optimized by minimizing memory transfers and using SIMD operations whenever possible, depending on the operation and type of data.

For parallelization of the computations at each single-physics solver, a distributed–memory model is used and communication between processes are established using the Message Passing Interface (MPI) standard. This configuration was seen in our previous tests to be equally or more efficient than a hybrid MPI–openMP arrangement for many CFD applications on a homogeneous CPU-only machine (see [45] for details). Moreover, it makes our code portable across all the distributed-memory systems, enabling us to efficiently use different clusters without the need to adapt the parallel implementation of the code to the specific architecture of the machine. This distributed-memory parallelism is complemented by a vectorization (SIMD) level within each core. The code also has a multi-threading option with CUDA or openCL to use GPUs as co–processors on a hybrid machine [46–48]. This option was not activated in this work since the system used for tests is a CPU-only cluster.

The distributed-memory parallelization is carried out based on a spatial domain decomposition. The computational grid Ω (either Ω_f or Ω_s) is divided into n non-overlapping subdomain blocks, $\Omega_0, ..., \Omega_{n-1}$, and each block is assigned to a different CPU core (MPI rank). Each block contains many cells (control volumes) that are *owned* by the assigned core, and several *halo* cells that represent the neighbours of the owned cells that are themselves owned by a different process. Neighbour cells are defined based on the notion of shared vertices. Figure 4.1 schematically represents the discretized domain and its decomposition into two subdomain blocks. The decomposition of the computational domain is carried out using the METIS library [49]. METIS divides the computational mesh into roughly equal partitions using a parallel multilevel *k-way* graph-partitioning method, minimizing the number of halo cells [49]. Roughly equal sizes of the blocks results in a roughly balanced load on different processes and a minimized number of halo cells means minimized required communications.

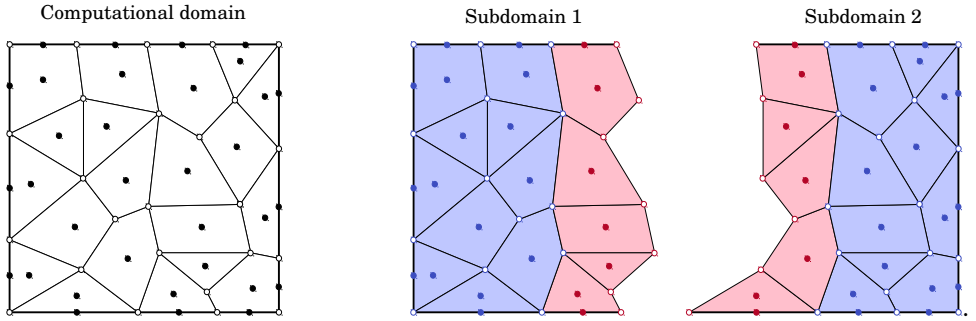

Figure 4.1: Schematic illustration of a discretized domain (left) and its decomposition into two subdomain blocks (right). Cell and boundary nodes are represented as filled circles and vertices as empty circles. The owned elements (cells, nodes and vertices) of each process are shown in blue while the halo elements are shown in red.

At the parallel level, the efficiency is mainly limited by the inter-process communications. Two types of inter-process communication are present in our implementation:

(i) global reduction operations which are used in the evaluation of norms, dot products, and for obtaining global extrema (e.g., in time step evaluation),

(ii) point-to-point communications between the processes in order to update the data in the halo cells.

The first type of communication is not required very frequently in the code and is carried out by simply calling the corresponding `MPI` collective operations. On the other hand, communication for data updates in the halo cells is carried out via non–blocking functions `MPI_Isend` and `MPI_Irecv` in order to avoid unnecessary synchronization. The synchronization is deferred to a later time when the `MPI_Waitall` function is called. An efficient and sparse scheme is implemented where each process stores the list of other processes with which it needs to

communicate. This is carried out only once at the initialization stage. In the communication episode, a loop is created over this small list to invoke the respective communication.

Another relevant aspect in computational performance of the code is the checkpointing and input and output operations (IO). Checkpoints are saved instants of data which are used for post-processing or to restart the simulations from an specific point. The IO operations in this work are managed via the HDF5 library [50]. Our implementation of the IO operations for checkpointing is seen in our previous tests to have a good parallel performance and to generate a reasonably low overhead [25].

4.4.2 Inter-code communication

Efficient inter–code communication is a key element in parallel partitioned coupled simulations. For this purpose, we use the preCICE coupling library [8]. Inter–code communication can be either based on MPI ports (MPI-2.0) or on lower level TCP/IP sockets. In the current work we have used TCP/IP sockets, because the MPI ports functionality is missing in the implemented MPI versions on the supercomputer that we used for our tests (SuperMUC supercomputer at the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences). To establish communication channels between the participants' ranks, the fluid and the solid mesh partitions are initially analyzed to find logical connections between them. Once the communication channels are established, data are exchanged in an asynchronous way to avoid unnecessary blocking. Note that establishing the channels is carried out only once at the initialization stage. For the rest of the run-time, the same channels are used for data exchange. Since using a central communication instance can degrade the scalability of our framework, we use a fully parallel point-to-point communication scheme and no central server–like unit is used. This way, data exchange happens locally between the connected ranks. It is obvious that, in case of having a big mesh or a high number of ranks and using a central communication instance, a single master rank or server would be a severe bottleneck.

Figure 4.2 schematically shows the parallel structure of the coupled framework and the different levels of communication.

4.4.3 Load balancing

Load imbalance can be a serious source of inefficiency in parallel simulation codes. Inside each single–physics solver, the load is balanced by dividing the computational domain into fairly equal blocks for each process, as explained in Section 4.4.1. However, in a partitioned FSI simulation, there exists a new level of load balancing between the single–physics solvers. In such simulations, the solvers need to communicate with each other and the output of one solver is used as the input for the other. In this work with a simultaneous execution of the solvers (Jacobi-type problem described in Section 4.3.3), both solvers must finish an iteration and send the output to the partner solver before the next iteration. Therefore, until the fluid solver has not received the data from the structural solver, it can not start the computations for the next iteration (and vice versa). This means, in case the available CPUs are not distributed optimally among the solvers, one solver will be waiting for the partner to finish its own computations. In

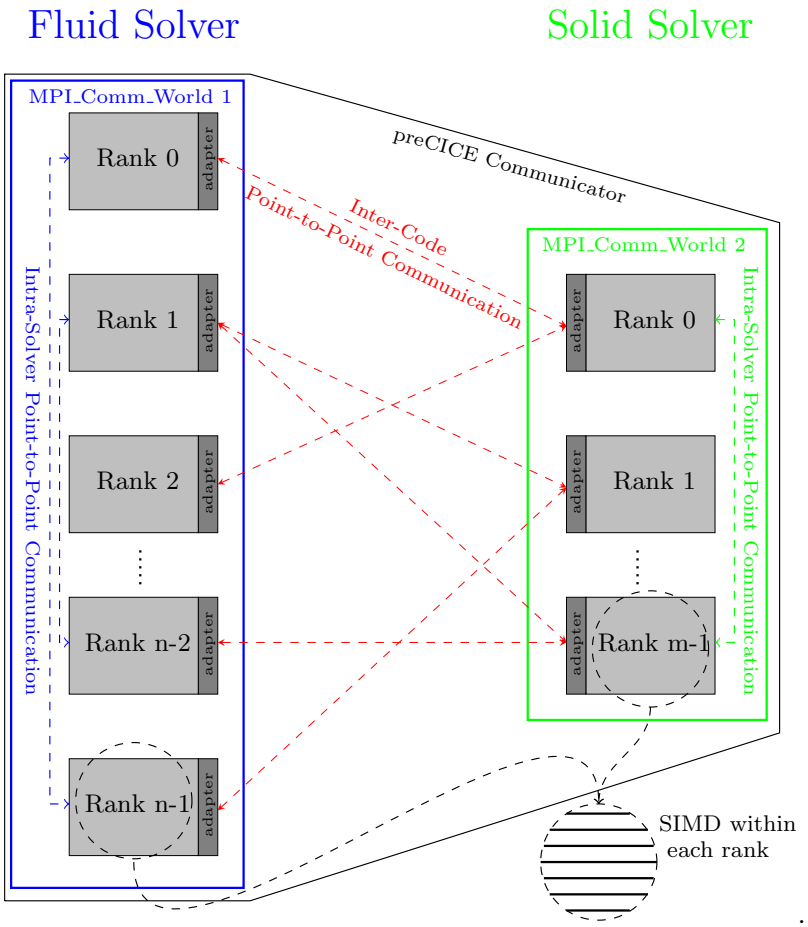

Figure 4.2: Parallelization model of the coupled framework and the different levels of communications.

contrast to the load balancing within a single–physics solver, load balancing across the codes is more difficult since we do not know a priori the relation between the work per cell of different solvers.

In this work, we follow the approach proposed in [51] to address this issue. We first model the solver performance against the number of ranks for each domain, and then solve an integer optimization problem to find the appropriate pair of core numbers for the domains that minimizes the waiting time. Since analytical modeling of the solvers' performance is very complex, we use an empirical approach instead, aiming to find an appropriate performance

model.

Assuming we have a given set of m data points, consisting of pairs (n, f_n) mapping the number of ranks n to the run-time f_n, we want to find a function $f(n)$ for each solver, which predicts the run-time for any n. We use the Performance Model Normal Form (PMNF) [52], defined in Eq. (4.25), as a basis for our prediction model:

$$f(n) = \sum_{k=1}^{q} c_k n^{i_k} \log_2^{j_k}(n) \tag{4.25}$$

where n is the number of ranks used by the solver, q is the number of terms used for the empirical run-time approximation, c_k is the weight of each term in the approximation function, and i_k and j_k are empirical coefficients. Calotoiu et al. [52] suggest, that the search space given by $q = 2$, $i_k \in \{\frac{0}{4}, \frac{1}{4}, \ldots, \frac{12}{4}\}$ and $j_k \in \{0, 1, 2\}$ is suitable for many applications. To find the optimal model, we simply check all the combinations within the search space, calculate a cross-validation-based loss for each combination (the accumulative error on the validation data set) and pick the combination with the smallest loss. By applying a PMNF regression, we are able to generate performance models $f(n)$ for each solver involved in the simulation.

As mentioned before, our goal is to find an optimal assignment of cores for each solver for a limited number of total available cores P, such that the overall run-time $F(n_f, n_s)$ is minimized (n_f and n_s are number of ranks used by the fluid and the solid solver respectively). This can be expressed by the following optimization problem:

$$\begin{aligned} &\underset{n_f, n_s}{\text{minimize}} \quad F(n_f, n_s) \qquad \text{with } F(n_f, n_s) = \max(f_s(n_s), f_f(n_f)) \\ &\text{subject to} \quad n_f + n_s \leq P. \end{aligned}$$

If the functions f_s and f_f are approximated by the PMNF regression, this optimization problem is a nonlinear, possibly non-convex integer optimization. We assume that f_s and f_f are both monotonically decreasing, i.e., assigning more cores to a solver never increases the run-time. With this, we can simplify the constraints to:

$$P = n_f + n_s. \tag{4.26}$$

The optimization problem can then be solved by brute-force checking all possible values for n_s and n_f in order to choose the pair that minimizes the total run-time. This pair of rank counts is then used to divide the available processors between the fluid and the structural solver for the FSI simulation. For more details, please refer to [51].

4.5 Numerical Tests

Numerical tests are provided in this section to demonstrate the parallel scalability of the coupled framework in solving practically relevant FSI problems. The test cases are in the context of hemodynamics. The first test case is a simplified benchmark problem representing the propagation of pressure waves inside a 3D deformable tube. The second test case is the blood flow inside a patient-specific aorta considering the elastic deformation of the aorta

wall. The scalability tests are carried out on the SuperMUC supercomputer at the Leibniz Supercomputing Centre of the Bavarian Academy of Sciences in Garching [53].

4.5.1 3D flow inside a deformable tube

This benchmark problem was proposed by [54] and studied, among others, by [12,39,55]. The problem is a 3D incompressible flow inside a straight tube with a deformable wall, motivated by the type of problems encountered in hemodynamics. The tube has a length of $l = 0.05$m, an inner radius of $R_0 = 0.005$m and a wall thickness of $h = 0.001$m. The fluid density and viscosity are $\rho_f = 1000\text{kg/m}^3$ and $\mu_f = 0.003\text{Pa} \cdot \text{s}$, respectively. The structural density is $\rho_s = 1200\text{kg/m}^3$, the Young modulus $E = 3 \times 10^5\text{N/m}^2$ and the Poisson ratio $\nu = 0.3$.

The tube is clamped at both ends and the fluid is initially at rest. An overpressure of 1333.2Pa is applied at the tube inlet during a period of 0.003s and a constant pressure of 0Pa afterwards. Pressure at the outlet is 0Pa during the whole simulation. A Neumann boundary condition is used for the velocity at both inlet and outlet boundaries.

Unstructured 3D meshes are used for both fluid and structural domains. Two meshes (M1 and M2) are used to solve the problem from $t = 0$ until $t = 0.02$s with constant time step sizes of $\Delta t = 10^{-4}$ and $\Delta t = 5 \times 10^{-5}$s, respectively. The information of the meshes are provided in Table 4.1. Figure 4.3 presents the radial displacement at the mid-length of the tube during the simulation time. As seen in the figure, the results obtained by mesh M1 and M2 are fairly close. Figure 4.4 shows the velocity vectors inside the deformed tube at two instants $t = 0.005$ and $t = 0.01$s. The color contour in the solid domain is the von Mises equivalent stress. The deformation of the wall is magnified by a factor of 10 to be visible better. The propagation of the pressure wave with a finite velocity inside the tube is seen in the figure.

Table 4.1: Computational grids used for the deformable tube test case.

Mesh name	No. of cells	
	Fluid	Structure
M1	60K	20K
M2	120K	40K
M3	9M	6M

For the scalability tests we used a much finer grid of M3 (Table 4.1). A complete simulation is not carried out for these tests and only the first 10 time steps are solved. Figure 4.5 shows the average run–time per time step for different numbers of CPU cores. The number of cores indicates the total number of cores divided between the fluid and solid solvers. The division of the available cores between the solvers is based on the load balancing model in Section 4.4.3. As seen in the figure, a good reduction in computational time is achieved by increasing the number of cores. Figure 4.6 shows the corresponding speed–ups in the calculation time and compares them to the ideal (linear) speed–up. Results show a very good scalability for up to 1400 cores. For higher core numbers the parallel efficiency degrades. This limit appears to be mostly determined by the size of the mesh used for the tests. The size of the computational

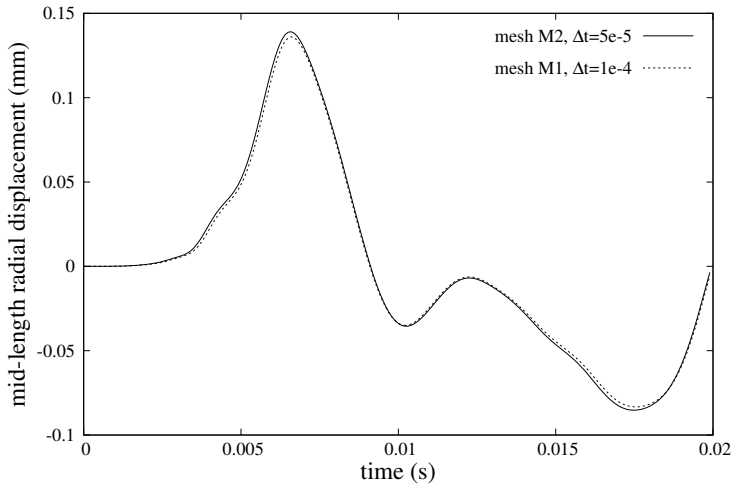

Figure 4.3: Radial displacement at the mid-length of the tube with two different meshes and time step sizes.

grid (total of 15M cells in mesh M3) is relatively small to be divided among over a thousand processes. For instance, dividing the mesh among 1400 cores means each process owns roughly 10K cells, which is too few calculations and corresponds to a very low arithmetic intensity (compared to communication). We believe, by repeating the scalability tests using finer grids, a greater scalability is achievable on higher number of cores. Such extended tests are left for expected future works.

At every time step, an average of roughly 17 coupling iterations were required to achieve convergence on the coupled problem (using $\epsilon_{FSI} = 10^{-5}$). This number remained fairly constant by increasing the number of processes, as seen in Table 4.2. This shows that the overall methodology for the coupled FSI problem is mathematically scalable. Table 4.2 also contains the data for the initialization time of the coupled framework. The reported time includes the initialization times of the single–physics solvers, as well as the time for establishing the inter–code communication channels and initializing the coupling library modules. The data in Table 4.2 shows that the initialization time remains fairly constant and only slightly increases by increasing the number of processes.

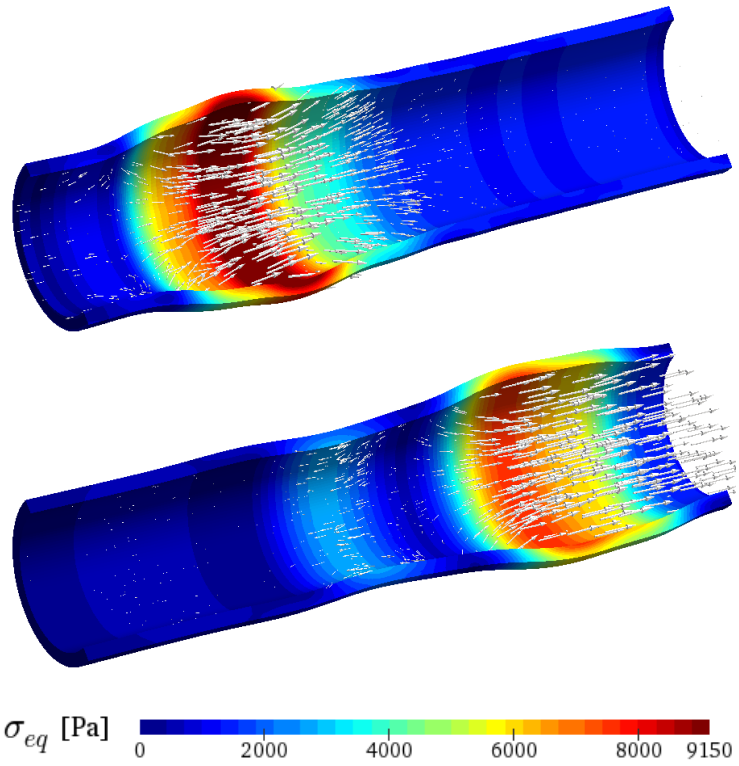

Figure 4.4: Propagation of the pressure wave inside the deformable tube. Fluid velocity vectors inside the deformed domain and the von Mises equivalent stress at the wall. Top: at $t = 0.005$s; Bottom: at $t = 0.01$s. Deformations magnified by a factor of 10.

Table 4.2: Average number of coupling iterations per time step, and the total initialization time of the coupled framework for test case 1 (deformable tube).

No. of cores	280	420	560	700	840	980	1120	1260	1400
coupling iterations	17.4	17.6	16.8	17.4	16.8	16.8	17.2	17.4	17.2
initialization time [s]	329.6	335.8	339.2	345.6	351.7	353.9	359.2	365.4	370.6

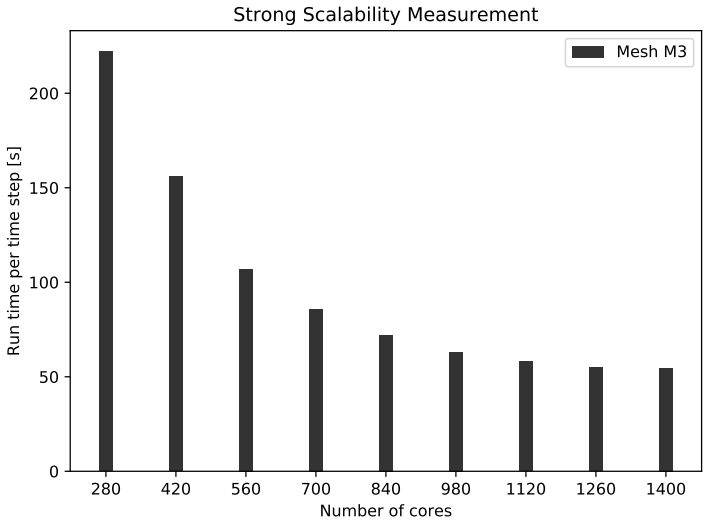

Figure 4.5: Average run–time per time step for different numbers of cores for test case 1 (deformable tube).

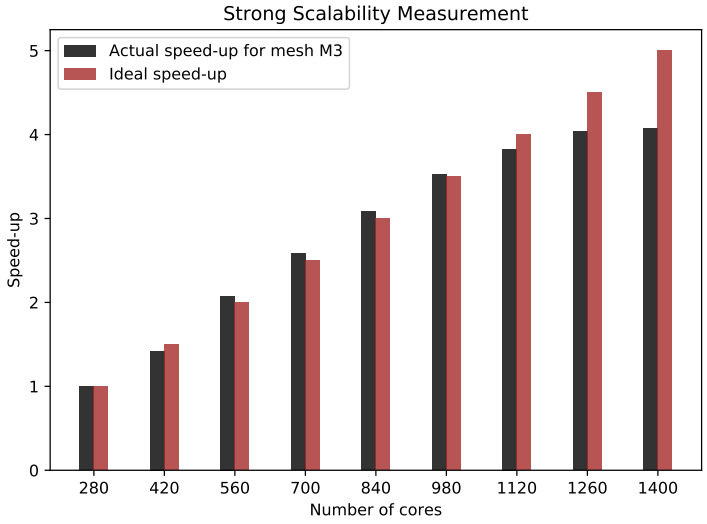

Figure 4.6: Speed–up in the calculations by increasing the number of cores for test case 1 (deformable tube).

4.5.2 Patient–specific aorta

The second test case is simulation of blood flow inside a patient-specific aorta with a mild thoracic aortic coarctation. The 3D geometry of the aorta is obtained by contrast agent magnetic resonance angiography, provided by the 2nd CFD challenge of the STACOM 2013 conference [56]. Figure 4.7 shows the provided geometry and the location of the boundaries. The thickness of the aorta wall and its mechanical properties were not provided in the challenge data. Therefore, we assume values in the typical physiological range. A uniform thickness of $h = 2\text{mm}$ is assumed for the wall along with density $\rho_s = 1200\text{kg/m}^3$, Young modulus $E = 3 \times 10^5\text{N/m}^2$ and Poisson ratio $\nu = 0.3$. The density and viscosity of the blood are considered to be $\rho_f = 1000\text{kg/m}^3$ and $\mu_f = 0.004\text{Pa} \cdot \text{s}$. These values for the wall thickness and the properties of the wall and blood are similar to the values used in [57].

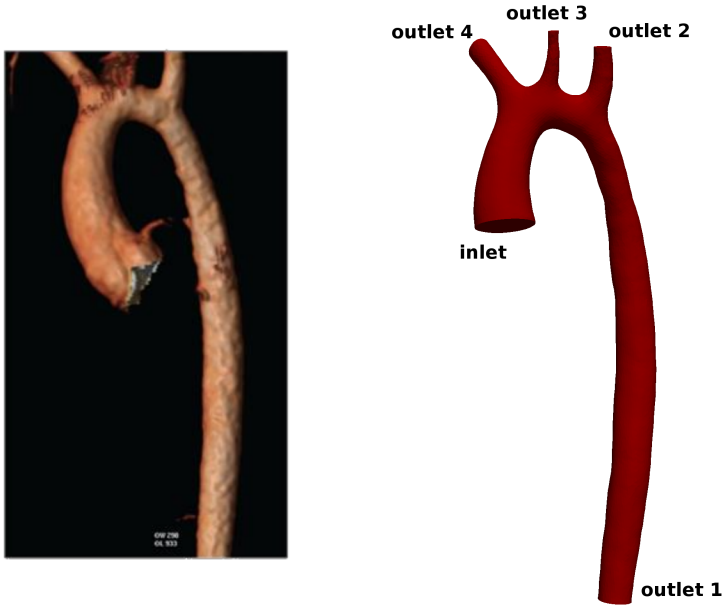

Figure 4.7: 3D geometry of a patient-specific aorta provided in [56].

For the inlet boundary of the aorta, a Dirichlet boundary condition is used for velocity, using measured physiological flow rate data provided in [56] for the rest condition. A Neumann boundary condition is used for the fluid pressure at the inlet. For the outlet boundaries, explicit RCR Windkessel boundary conditions [58] are used to model the effect of the rest of the vascular network. The Windkessel parameters are chosen as those reported in [59]. For the solid, a zero–displacement (clamped) boundary condition is set at the inlet and at the outlets, while a traction-free boundary condition is used on the outer surface of the wall.

The objective of the current work is not a deep study of the biophysical phenomena featured in this test case, but rather to show the capability of the developed framework to solve such a complex problem on a massively-parallel configuration. Nevertheless, the obtained fluid velocity and pressure fields, as well as the structural displacements were seen to be in the reasonable physiological range. Figure 4.8 demonstrates the solution at two instants, $t = 0.05$s and $t = 0.1$s. The figure contains the velocity vector plot inside the deformed aortic wall. The color contours in the structural domain correspond to the von Mises equivalent stress.

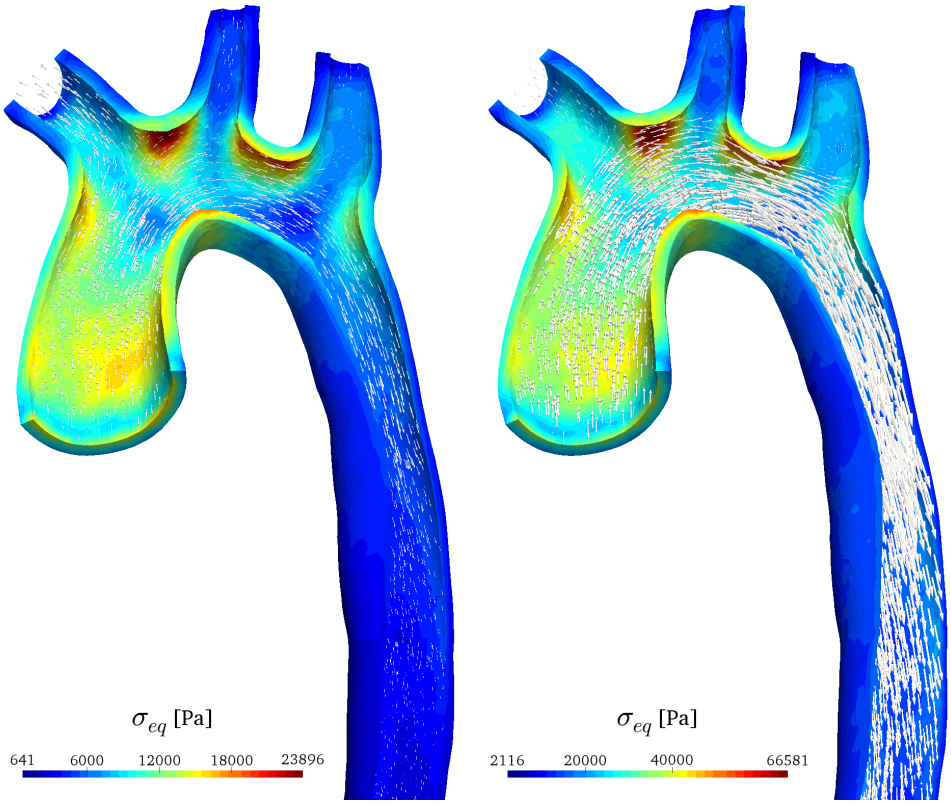

Figure 4.8: Fluid velocity vectors inside the deformed aortic wall and the von Mises equivalent stress at the wall. Left: at $t = 0.05$s; Right: at $t = 0.1$s.

The strong scalability of the developed framework is evaluated using the computational grid M1 (Table 4.3). Both fluid and structural grids are unstructured tetrahedral meshes. The first 10 time steps are solved for the scalability tests. Figure 4.9 shows the average run–time per time step for different numbers of cores. Similar to the previous test case, the number of

cores indicates the total number of processes of the fluid and solid solvers. The division of the available cores between the solvers is based on the load balancing model in Section 4.4.3. As seen in the figure, a very good reduction in computational time is achieved by increasing the number of cores. Figure 4.10 presents the corresponding speed–ups and compares them to the ideal speed–up. Results show a very good scalability for up to 3360 cores and still adequate for up to 3920 processes. For higher core numbers the parallel efficiency degrades. Similar to the previous test case, this limit appears to be mostly determined by the size of the mesh used for the scalability tests. The size of the computational grid (total of 29M cells in mesh M1) is not large enough to be divided among four thousand processes. We believe, by repeating the scalability tests using finer grids, a greater scalability is achievable on higher number of cores. Such extended tests are left for expected future works.

Table 4.3: Computational grid used for the patient-specific aorta test case.

Mesh name	No. of cells	
	Fluid	Structure
M1	20M	9M

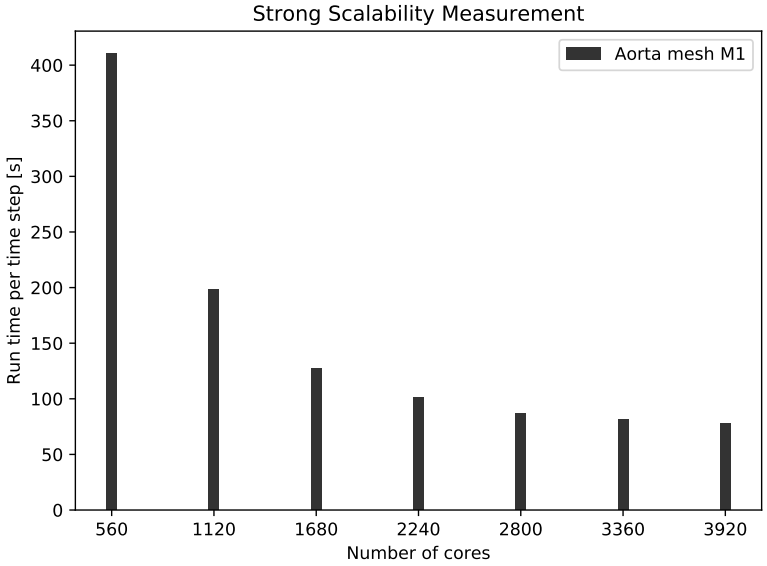

Figure 4.9: Average run–time per time step for different numbers of cores for test case 2 (patient–specific aorta).

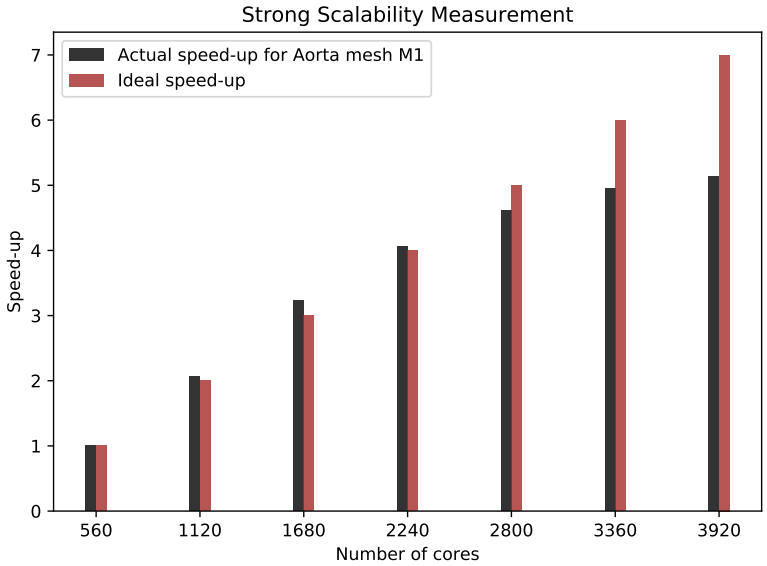

Figure 4.10: Speed–up in the calculation time by increasing the number of cores for test case 2 (patient–specific aorta).

At every time step, an average of roughly 11 coupling iterations were required to achieve convergence on the coupled problem (using $\epsilon_{FSI} = 10^{-3}$). This number remained constant by increasing the number of processes, as seen in Table 4.4. This shows that the overall methodology for the coupled FSI problem is mathematically scalable. Table 4.4 also contains the data for the initialization time of the coupled framework. Similar to the previous test case, the reported time includes the initialization times of the single–physics solvers, as well as the time for establishing the inter–code communication channels and initializing the coupling library modules. The data in Table 4.4 shows that the initialization time remains fairly constant and only slightly increases by increasing the number of processes.

Table 4.4: Average number of coupling iterations per time step, and the total initialization time of the coupled framework for test case 2 (patient–specific aorta).

No. of cores	560	1120	1680	2240	2800	3360	3920
coupling iterations	10.6	10.6	10.6	10.6	10.6	10.6	10.6
Initialization time [s]	263.9	261.3	274.8	308.8	309.7	328.8	352.5

4.6 Conclusions

An efficient and scalable parallel framework is presented for partitioned solution of fluid–structure interaction problems. The framework is developed through multi–code coupling, using separate fluid and structural solvers. Both single–physics solvers use distributed–memory parallelism based on spatial domain decomposition. The intra–solver communications required for data update in the solution process are carried out using non–blocking point–to–point communicators. The inter–code communications are also fully parallel and point–to–point, avoiding any central communication unit. Both the intra–solver and inter–code communication channels are established at the initialization stage, before the simulation begins. This configuration is seen to produce very rapid and efficient communications.

Inside each single–physics solver the load is balanced by dividing the computational domain into fairly equal blocks for each process. The available processor cores are divided between the fluid and solid solvers based on a load balancing model that minimizes the total idle time of the processes.

Both fluid and structural solvers discretize and solve the corresponding equations on unstructured 3D meshes. A semi–implicit coupling method is used, in which the fluid pressure term is segregated and strongly coupled to the structure. The remaining fluid terms and the geometrical nonlinearities are only loosely coupled. Strong coupling of the fluid pressure and structural deformation provides for the stability of the method in FSI problems with strong added–mass effect. On the other hand, loose coupling of the remaining terms considerably reduces the computational time of the simulations. An efficient multi–vector quasi–Newton method is used to solve the coupled interface problem.

Two numerical test cases in the context of hemodynamics are considered and the strong scalability of the coupled framework is evaluated. The first test case is a benchmark FSI problem, solving an incompressible flow inside a deformable tube. The second test case solves the blood flow inside a patient–specific aorta. Results confirmed a very good scalability for up to 3920 cores. The test results also suggest that by using a finer mesh for the scalability tests, a greater scalability on higher number of cores is achievable.

References

[1] Joris Degroote. Partitioned simulation of fluid-structure interaction. *Archives of Computational Methods in Engineering*, 20:185–238, 2013.

[2] Gene Hou, Jin Wang, and Anita Layton. Numerical methods for fluid-structure interaction - A review. *Communications in Computational Physics*, 12(2):337–377, 2012.

[3] Fande Kong and Xiao-Chuan Cai. A scalable nonlinear fluid–structure interaction solver based on a schwarz preconditioner with isogeometric unstructured coarse spaces in 3d. *Journal of Computational Physics*, 340:498–518, 2017.

[4] Shunji Kataoka, Satsuki Minami, Hiroshi Kawai, Tomonori Yamada, and Shinobu Yoshimura. A parallel iterative partitioned coupling analysis system for large-scale acoustic fluid–structure interactions. *Computational Mechanics*, 53(6):1299–1310, 2014.

[5] JC Cajas, G Houzeaux, M Vázquez, M GarciÌa, E Casoni, H Calmet, A Artigues, R Borrell, O Lehmkuhl, D Pastrana, et al. Fluid-structure interaction based on hpc multicode coupling. *SIAM Journal on Scientific Computing*, 40(6):C677–C703, 2018.

[6] Fraunhofer Institute for Algorithms and Scientific Computing (SCAI). MpCCI coupling environment, 2018. www.mpcci.de.

[7] Jay Larson, Robert Jacob, and Everest Ong. The model coupling toolkit: a new fortran90 toolkit for building multiphysics parallel coupled models. *The International Journal of High Performance Computing Applications*, 19(3):277–292, 2005.

[8] Hans-Joachim Bungartz, Florian Lindner, Bernhard Gatzhammer, Miriam Mehl, Klaudius Scheufele, Alexander Shukaev, and Benjamin Uekermann. precice–a fully parallel library for multi-physics surface coupling. *Computers & Fluids*, 141:250–258, 2016.

[9] Hans-Joachim Bungartz, Florian Lindner, Miriam Mehl, Klaudius Scheufele, Alexander Shukaev, and Benjamin Uekermann. Partitioned fluid–structure–acoustics interaction on distributed data: Coupling via precice. In Hans-Joachim Bungartz, Philipp Neumann, and Wolfgang E. Nagel, editors, *Software for Exascale Computing - SPPEXA 2013-2015*, pages 239–266, Cham, 2016. Springer International Publishing.

[10] 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*, 194:4506–4527, 2005.

[11] C. Förster, W. A. Wall, and E. Ramm. Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows. *Computer Methods in Applied Mechanics and Engineering*, 196:1278–1293, 2007.

[12] 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. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.

[13] M. Astorino, F. Chouly, and M. A. Fernández. Robin based semi-implicit coupling in fluid-structure interaction: Stability analysis and numerics. *SIAM Journal on Scientific Computing*, 31(6):4041–4065, 2009.

[14] A. Naseri, O. Lehmkuhl, I. Gonzalez, E. Bartrons, C. D. Pérez-Segarra, and A. Oliva. A semi-implicit coupling technique for fluid–structure interaction problems with strong added-mass effect. *Journal of Fluids and Structures*, 80:94–112, 2018.

[15] Alireza Naseri, Ignacio Gonzalez, Ahmad Amani, Carlos David Pérez-Segarra, and Assensi Oliva. A second–order time accurate semi–implicit method for fluid–structure interaction problems. *Journal of Fluids and Structures*, 86:135–155, 2019.

[16] O Lehmkuhl, CD Perez-Segarra, R Borrell, M Soria, and A Oliva. TermoFluids: A new parallel unstructured CFD code for the simulation of turbulent industrial problems on low cost PC cluster. In *Parallel Computational Fluid Dynamics 2007*, pages 275–282. Springer, 2009.

[17] Ivette Rodriguez, Ricard Borell, Oriol Lehmkuhl, Carlos D. Perez Segarra, and Assensi Oliva. Direct numerical simulation of the flow over a sphere at Re = 3700. *Journal of Fluid Mechanics*, 679:263–287, 2011.

[18] I. RodrÃguez, O. Lehmkuhl, J. Chiva, R. Borrell, and A. Oliva. On the flow past a circular cylinder from critical to super-critical reynolds numbers: Wake topology and vortex shedding. *International Journal of Heat and Fluid Flow*, 55:91 – 103, 2015.

[19] Néstor Balcázar, Lluís Jofre, Oriol Lehmkuhl, Jesús Castro, and Joaquim Rigola. A finite-volume/level-set method for simulating two-phase flows on unstructured grids. *International journal of multiphase flow*, 64:55–72, 2014.

[20] E Gutiérrez, F Favre, Nestor Balcazar, Ahmad Amani, and J Rigola. Numerical approach to study bubbles and drops evolving through complex geometries by using a level set–moving mesh–immersed boundary method. *Chemical Engineering Journal*, 349:662–682, 2018.

[21] PA Galione, O Lehmkuhl, J Rigola, and A Oliva. Fixed-grid numerical modeling of melting and solidification using variable thermo-physical properties–application to the melting of n-octadecane inside a spherical capsule. *International Journal of Heat and Mass Transfer*, 86:721–743, 2015.

[22] Eduard Bartrons, Carles Oliet, Enrique Gutierrez, Alireza Naseri, and Carlos David Pérez-Segarra. A finite volume method to solve the frost growth using dynamic meshes. *International Journal of Heat and Mass Transfer*, 124:615–628, 2018.

[23] Guillem Colomer, Rick Borrell, F Xavier Trias, and I Rodríguez. Parallel algorithms for Sn transport sweeps on unstructured meshes. *Journal of Computational Physics*, 232(1):118–135, 2013.

[24] Lluís Jofre, Ricard Borrell, Oriol Lehmkuhl, and Assensi Oliva. Parallel load balancing strategy for volume-of-fluid methods on 3-d unstructured meshes. *Journal of Computational Physics*, 282:269–288, 2015.

[25] R Borrell, J Chiva, O Lehmkuhl, G Oyarzun, I Rodríguez, and A Oliva. Optimising the Termofluids CFD code for petascale simulations. *International Journal of Computational Fluid Dynamics*, 30(6):425–430, 2016.

[26] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187(1):343–368, 2003.

[27] F. X. Trias, O. Lehmkuhl, A. Oliva, C. D. Pérez-Segarra, and R.W.C.P. Verstappen. Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids. *Journal of Computational Physics*, 258:246–267, 2014.

[28] Lluís Jofre, Oriol Lehmkuhl, Jordi Ventosa, F Xavier Trias, and Assensi Oliva. Conservation properties of unstructured finite-volume mesh schemes for the navier-stokes equations. *Numerical Heat Transfer, Part B: Fundamentals*, 65(1):53–79, 2014.

[29] PD Thomas and CK Lombard. Geometric conservation law and its application to flow computations on moving grids. *AIAA journal*, 17(10):1030–1037, 1979.

[30] Michel Lesoinne and Charbel 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(1-2):71–90, 1996.

[31] O. Estruch, O. Lehmkuhl, R. Borrell, C. D Pérez Segarra, and A. Oliva. A parallel radial basis function interpolation method for unstructured dynamic meshes. *Computers and Fluids*, 80:44–54, 2013.

[32] Philip Cardiff and Ismet Demirdžić. Thirty years of the finite volume method for solid mechanics. *arXiv preprint arXiv:1810.02105*, 2018.

[33] Hrvoje Jasak. *Error analysis and estimation for the finite volume method with applications to fluid flows*. PhD thesis, Imperial College London (University of London), 1996.

[34] A. J. Macleod. Acceleration of vector sequences by multiâdimensional Δ^2 methods. *Communications in Applied Numerical Methods*, 2(4):385–392, 1986.

[35] Ž. Tuković, A. Ivanković, and A. Karač. Finite-volume stress analysis in multi-material linear elastic body. *International Journal for Numerical Methods in Engineering*, 93(4):400–419, 2013.

[36] P. Cardiff, Tuković, H. Jasak, and A. Ivanković. A block-coupled Finite Volume methodology for linear elasticity and unstructured meshes. *Computers and Structures*, 175:100–122, 2016.

[37] P. Chandrashekar and A. Garg. Vertex-centroid finite volume scheme on tetrahedral grids for conservation laws. *Computers & Mathematics with Applications*, 65(1):58–74, 2013.

[38] Ignacio González, Alireza Naseri, Jorge Chiva, Joaquim Rigola, and Carlos D. Pérez-Segarra. An enhanced finite volume based solver for thermoelastic materials in fluid-structure coupled problems. In *6th European Conference on Computational Mechanics (ECCM-ECFD 2018)*. ECCOMAS, Glasgow, UK, June 2018.

[39] U. Küttler and W. A. Wall. Fixed-point fluid-structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43:61–72, 2008.

[40] Jean Frédéric Gerbeau and Marina Vidrascu. A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37:631–647, 2003.

[41] C Michler, E H Van Brummelen, and R De Borst. An interface Newton-Krylov solver for fluid-structure interaction. *International Journal for Numerical Methods in Fluids*, 47(10-11):1189–1195, 2005.

[42] Klaudius Scheufele and Miriam Mehl. Robust multisecant quasi-Newton variants for parallel fluid-structure simulations–and other multiphysics applications. *SIAM Journal on Scientific Computing*, 39(5):S404–S433, 2017.

[43] Hans-Joachim Bungartz, Florian Lindner, Miriam Mehl, and Benjamin Uekermann. A plug-and-play coupling approach for parallel multi-field simulations. *Computational Mechanics*, 55(6):1119–1129, 2015.

[44] Miriam Mehl, Benjamin Uekermann, Hester Bijl, David Blom, Bernhard Gatzhammer, and Alexander Van Zuijlen. Parallel coupling numerics for partitioned fluid–structure interaction simulations. *Computers & Mathematics with Applications*, 71(4):869–891, 2016.

[45] Ricard Borrell Pol. *Parallel algorithms for computational fluid dynamics on unstructured meshes*. PhD thesis, Universitat Politècnica de Catalunya (UPC), 2012.

[46] G Oyarzun, R Borrell, A Gorobets, and A Oliva. MPI-CUDA sparse matrix–vector multiplication for the conjugate gradient method with an approximate inverse preconditioner. *Computers & Fluids*, 92:244–252, 2014.

[47] Guillermo Oyarzun, Ricard Borrell, Andrey Gorobets, and Assensi Oliva. Portable implementation model for CFD simulations. application to hybrid CPU/GPU supercomputers. *International Journal of Computational Fluid Dynamics*, 31(9):396–411, 2017.

[48] Guillermo Oyarzun, Ricard Borrell, Alexander Gorobets, Filippo Mantovani, and Assensi Oliva. Efficient CFD code implementation for the ARM-based Mont-Blanc architecture. *Future generation computer systems*, 79:786–796, 2018.

[49] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

[50] The HDF Group. Hierarchical Data Format, version 5, 1997-2019. http://www.hdfgroup.org/HDF5/.

[51] A Totounferoush. Fast multi-parameter performance modeling. IEEE, 2019.

[52] Alexandru Calotoiu, David Beckinsale, Christopher W Earl, Torsten Hoefler, Ian Karlin, Martin Schulz, and Felix Wolf. Fast multi-parameter performance modeling. In *2016 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 172–181. IEEE, 2016.

[53] SuperMUC Petascale System, Leibniz Supercomputing Centre , 2019. https://www.lrz.de/services/compute/supermuc/.

[54] L. Formaggia, J. F. Gerbeau, F. Nobile, and A. Quarteroni. On the coupling of 3D and 1D Navier-Stokes equations for flow problems in compliant vessels. *Computer Methods in Applied Mechanics and Engineering*, 191(6-7):561–582, 2001.

[55] Miguel Angel Fernández and Marwan Moubachir. A Newton method using exact Jacobians for solving fluid–structure coupling. *Computers and Structures*, 83:127–142, 2005.

[56] 2nd CFD challenge predicting patient-specific hemodynamics at rest and stress through an aortic coarctation, 2013. http://www.vascularmodel.org/miccai2013/.

[57] Miguel A Fernández, Mikel Landajuela, and Marina Vidrascu. Fully decoupled time-marching schemes for incompressible fluid/thin-walled structure interaction. *Journal of Computational Physics*, 297:156–181, 2015.

[58] Nico Westerhof, Jan-Willem Lankhaar, and Berend E Westerhof. The arterial Windkessel. *Medical & biological engineering & computing*, 47(2):131–141, 2009.

[59] Sanjay Pant, Benoit Fabrèges, J-F Gerbeau, and IE Vignon-Clementel. A methodological paradigm for patient-specific multi-scale cfd simulations: from clinical measurements to parameter estimates for individual analysis. *International journal for numerical methods in biomedical engineering*, 30(12):1614–1648, 2014.

5

Concluding remarks

This thesis was dedicated to developing numerical methods and tools for simulation of complex fluid–structure interaction problems. In the first chapter of the thesis we have provided an introduction to the FSI problems and an extensive literature review on the numerical methods for their solution. The current work was focused on the semi–implicit partitioned approach, due to its promising features. Moreover, the study was focused on complex and challenging FSI problems, including ones with very strong added–mass effect and very large deformations. The research work carried out in this thesis could be divided into three major parts which are presented in the second, third and the fourth chapters of this document.

In the second chapter, a simple and robust semi–implicit coupling method is proposed for FSI problems with strong added–mass effect. A projection method is used to split the pressure stress term of the fluid and implicitly couple it to the structure. The remaining fluid terms as well as the geometrical nonlinearities are only explicitly coupled. An ALE formulation and conforming mesh technique is used to solve the fluid flow in a moving domain. A parallel radial basis function method is used to move the computational grid. The main advantages of the proposed method are its simplicity, modularity, robustness and computational efficiency. Extensive numerical tests were carried out on three widely different test cases, which demonstrated adequate stability, accuracy and computational efficiency of the proposed method. Results of the simulations were validated against experimental data and other numerical results from the literature, as well as domestic results obtained by using a fully–implicit coupling method. Results demonstrated that the proposed semi–implicit method significantly reduces the computational cost of the simulations without undermining either stability or accuracy of the results. In this chapter we have also evaluated the effect of implicit or explicit treatment of the geometrical nonlinearities on the overall performance and accuracy of the semi–implicit method. Results showed that implicit treatment of the geometrical nonlinearities considerably increases the computational cost while only slightly improving the accuracy. This conclusion was shown to stand even for problems with large structural deformations. Finally, we have compared the performance of fixed–point and Newton–Krylov methods to solve the coupled interface problem. Results showed that the Newton–Krylov solver outperforms the fixed–point method in a problem that requires many iterations to converge. On the other hand, in a FSI problem which does not require many coupling iterations, fixed–point method with Aitken's relaxation is probably a better choice, considering its simplicity and good performance.

In the third chapter, a second–order time accurate semi–implicit method is proposed for solution of FSI problems. The method uses a second–order projection method to solve the fluid equations and also as a framework for the FSI coupling. The fluid pressure term is effectively segregated using the projection method and is strongly coupled to the structure. The remaining fluid terms and the geometrical nonlinearities are treated explicitly. An ALE formulation with a conforming mesh technique was used to solve the fluid flow and the arisen geometrical terms were evaluated with a second–order temporal accuracy. Consistent boundary conditions were developed for the predicted velocity field which is encountered when solving the fluid equations with a projection method. Particular attention was paid to second–order accuracy of the fluid pressure, up to the moving boundary. Second–order accuracy of the method for coupled nonlinear FSI problems was demonstrated through rigorous numerical tests. Simulation results were validated against experimental and numerical results from the literature. Different time step sizes were used to solve the problems and the error was evaluated with respect to a reference numerical solution. Second–order temporal accuracy for all the variables of interest was clearly demonstrated.

In the fourth chapter, an efficient and scalable parallel framework was presented for partitioned solution of FSI problems, through multi–code coupling. Two instances of our in–house software were used to solve the single–physics sub–problems and an external coupling library was used for inter–code communications. Both fluid and structural solvers used distributed–memory parallelism. The intra–solver communications required for data update in the solution process were carried out using non–blocking point–to–point communicators. The inter–code communications were also fully parallel and point–to–point, avoiding any central communication unit. Both the intra–solver and inter–code communication channels were established at the initialization stage, before the simulations began. This configuration was seen to produce very rapid and efficient communications. Load balancing was considered in two levels, inside each single–physics solver and in the inter–code level. The semi–implicit FSI coupling methods developed in the previous chapters were used here. An efficient multi–vector quasi–Newton method was used to solve the coupled interface problem. Parallel efficiency and scalability of the coupled framework was demonstrated on practical FSI problems.

Overall, this thesis presents several original contributions to the field of numerical methods for FSI problems. The proposed methods improve the existing methods of the literature in many aspects, including accuracy and computational efficiency. These improvements directly reflect in the application of the proposed methods to real–life engineering problems. Moreover, the methods and tools developed in this work are capable of efficiently running on massively–parallel machines, which is a crucial aspect in the modern computational engineering.

6

Future work

This work could be continued in different ways. The most interesting extension, in the opinion of the author, is to use the developed methods and tools to study complex physical and engineering systems that feature FSI. Since this work was mostly focused on developing new methods, the numerical tests mainly served to demonstrate the capabilities of the proposed methods rather than deep study of the physical problems. A very interesting application of FSI is the blood flow analysis inside deformable vessels in human cardiovascular system. The methods and tools in this study are particularly developed for such problems and their capabilities are demonstrated. However, a deep study of the biophysical phenomena in such systems was beyond the scope of the current work and could be a very interesting extension for future.

Another possible extension is to improve the efficiency and accuracy of the developed methods. More accurate, more robust, higher-order and more cost-efficient methods could be achieved by carefully analyzing the current methods and detecting their potential weak points to improve.

Moreover, the scalability tests of the developed coupled framework in chapter 4 could be further extended. The tests in chapter 4 showed a very good strong scalability on thousands of processors. However, they do not represent the limits of the scalability of the framework. We believe by using finer meshes for the tests, a good scalability on even more cores could be achieved and demonstrated. Also the weak scalability of the framework was not evaluated in this work and could be studied in the future.

Appendix A

Integral form of governing equations and discretization in space

Throughout this thesis the governing equations are presented in their differential form and the semi-discretization of the equations in time is described, independent of the choice for discretization in space. This time-discretized equations could then be used with different spatial discretizations. For the sake of completeness, the spatial discretization method used in this work and the integral form of the conservation equations are described here. We only explain the discretization for the fluid equations in ALE form, as the structural equations are used in their standard Lagrangian form. This annex is complementary to the main text and follows the same notation and nomenclature. The subscript f used in the main text to refer to the fluid properties is dropped here (e.g. ρ_f in the main text is simply shown by ρ here).

We use a finite-volume method with collocated mesh arrangement for space discretization. This method is based on the discretization of the computational domain into a finite number of non-overlapping control volumes (CVs). Each CV has an associated grid node P located at its centroid where the equations are solved. Each CV has a volume v, surrounded by a control surface s, which consists of an arbitrary number of well-defined neighbouring faces. Quantities associated to a grid node (or cell center) are indicated by capital subscripts (e.g. P or N), while the values at the faces are indicated by lower-case subscripts (e.g. pn to refer to the face located between cell nodes P and N). The area vector of a face is referred to as $\mathbf{A}_{pn}$ ($\mathbf{A}_{pn} = A_{pn}\mathbf{n}_{pn}$ where A_{pn} is the surface area of the face and $\mathbf{n}_{pn}$ is the normal vector pointing outwards). In this work we use an ALE method on a moving mesh which means the shape and volume of the CVs are varying in time ($v = v(t)$ and $s = s(t)$). Therefore, the equations of conservation of mass and momentum are integrated over time-varying CVs. The integral form of the governing equations are as follows:

$$\frac{d}{dt}\int_v \rho dv + \int_s \rho(\mathbf{u} - \mathbf{w}) \cdot d\mathbf{A} = 0 \tag{A.1}$$

$$\frac{d}{dt}\int_v \rho \mathbf{u} dv + \int_s \rho \mathbf{u}(\mathbf{u}-\mathbf{w})\cdot d\mathbf{A} = \int_s \boldsymbol{\sigma}\cdot d\mathbf{A} \tag{A.2}$$

The time change of the CV is taken into account in both mass and momentum equations. The change of volume of each CV is present in the transient terms (first term in both equations) while the movement of the surface of the CV is reflected as additional mass and momentum fluxes (second term in both equations).

This set of equations have an extra unknown which is the velocity of the domain ($\mathbf{w}$). Another conservation law to close the system is the space conservation law (see sections 2.2.5 and 3.3.3), which guarantees the conservation of space (volume) in the moving domain:

$$\frac{d}{dt}\int_v dv - \int_s \mathbf{w}\cdot d\mathbf{A} = 0 \tag{A.3}$$

Comparing Eq. (A.1) and (A.3) (and assuming incompressibility) we realize that the equation of conservation of mass on the moving domain is identical to that equation in a fixed domain:

$$\int_s \rho \mathbf{u}\cdot d\mathbf{A} = \sum_{pn\in s} \dot{m}_{pn} = 0 \tag{A.4}$$

where $\dot{m}_{pn}$ is the mass flux at the face pn and the summation $\sum_{pn\in s}$ is over all the faces of the CV. Surface integrals at the faces are approximated using the mid-point rule, evaluating the mass flux at a given face as $\dot{m}_{pn} = \rho_{pn}\mathbf{u}_{pn}\cdot\mathbf{A}_{pn}$, where the density and velocity are evaluated at the centroid of the face (here, the density is constant as we consider incompressible flow).

The momentum equation is solved using a projection method as described in sections 2.2.4 and 3.3.1. In chapter 3, the ALE convection-diffusion equation for the predicted velocity field is discretized in time using the second-order method in Eq. (3.12). In the finite volume method used in this work, the volume integrals are evaluated at their own associated time levels:

$$\frac{1}{\Delta t}\Big(\int_{v^{n+1}} \rho \mathbf{u}^* dv - \int_{v^n} \rho \mathbf{u}^n dv\Big) =$$

$$-\Big[\frac{3}{2}\int_{s^n} \rho \mathbf{u}^n(\mathbf{u}^n - \mathbf{w}^n)\cdot d\mathbf{A} - \frac{1}{2}\int_{s^{n-1}} \rho \mathbf{u}^{n-1}(\mathbf{u}^{n-1} - \mathbf{w}^{n-1})\cdot d\mathbf{A}\Big] \tag{A.5}$$

$$+\frac{\mu}{2}\Big(\int_{s^{n+1}} \nabla \mathbf{u}^*\cdot d\mathbf{A} + \int_{s^n} \nabla \mathbf{u}^n\cdot d\mathbf{A}\Big) - \int_{s^{n-1/2}} p^{n-1/2} d\mathbf{A}$$

which is a consistent second-order discretization of the ALE convective-diffusive equation on a moving mesh. The intermediate mesh $s^{n-1/2}$ for the pressure term is obtained as an average between s^n and s^{n-1}:

$$\int_{s^{n-1/2}} p^{n-1/2} d\mathbf{A} = \sum_{pn\in s} p_{pn}^{n-1/2}\mathbf{A}_{pn}^{n-1/2} = \sum_{pn\in s}\frac{1}{2}p_{pn}^{n-1/2}(\mathbf{A}_{pn}^{n} + \mathbf{A}_{pn}^{n-1}) \tag{A.6}$$

The convective term is evaluated as some of the fluxes on the faces:

$$\int_s \rho \mathbf{u}(\mathbf{u}-\mathbf{w}) \cdot d\mathbf{A} = \sum_{pn \in s} (\dot{m}_{pn} - \dot{m}_{mesh_{pn}}) \mathbf{u}_{pn} \tag{A.7}$$

where $\dot{m}_{mesh_{pn}}$ refers to the additional mass flux due to the movement of the face:

$$\dot{m}_{mesh_{pn}} = \rho_{pn} \mathbf{w}_{pn} \cdot \mathbf{A}_{pn} \tag{A.8}$$

which will be evaluated using the space conservation law. The velocity on the face centroid to evaluate the mass flux is obtained using a distance-weighted average between the velocities at the grid nodes on either side of the face:

$$\dot{m}_{pn} = \rho \frac{\mathbf{u}_N \delta x_P + \mathbf{u}_P \delta x_N}{\delta x_P + \delta x_N} \cdot \mathbf{A}_{pn} \tag{A.9}$$

where δx_P and δx_N are the distance between the face centroid and the grid nodes P and N, respectively. The convected velocity on the face is evaluated using a symmetry-preserving scheme which is a non-weighted central scheme

$$\mathbf{u}_{pn} = \frac{\mathbf{u}_P + \mathbf{u}_N}{2} \tag{A.10}$$

as in [1,2]. The diffusive term is also evaluated on the faces as:

$$\int_s \nabla \mathbf{u} \cdot d\mathbf{A} = \sum_{pn \in s} \nabla \mathbf{u}_{pn} \cdot \mathbf{A}_{pn} \tag{A.11}$$

and the gradients are evaluated using the neighbouring grid node values. The approximations in equations (A.9)–(A.11) and their relation with the overall discretization are similar to the case of a constant domain problem (Eulerian form) and are described in more details in [2,3].

The space conservation law (Eq. (A.3)) is used to evaluate the additional fluxes due to the mesh velocity in order to be used in the momentum equation (Eq. (A.2), (A.5) and (A.7)). In the discretized form, the change of volume of the CV could be represented by the sum of the volumes swept by the faces of that CV. A first-order discretization of this equation reads:

$$\frac{v^{n+1} - v^n}{\Delta t} = \sum_{pn \in s} \frac{\delta v_{pn}^{n+1}}{\Delta t} \tag{A.12}$$

which is used in chapter 2. In this equation δv_{pn}^{n+1} indicates the volume swept by the face pn (as in figures 2.1 and 3.1). Thus, the additional flux in the momentum equation could be approximated as $\dot{m}_{mesh_{pn}}^{n+1} = \rho \frac{\delta v_{pn}^{n+1}}{\Delta t}$, which is first-order in time (as used in chapter 2).

Alternatively, a second-order discretization of this equation is used in chapter 3:

$$\frac{3v^{n+1} - 4v^n + v^{n-1}}{2\Delta t} = \sum_{pn \in s} \frac{3\delta v_{pn}^{n+1} - \delta v_{pn}^n}{2\Delta t} \tag{A.13}$$

which uses the information of the swept volume by face pn in the two consecutive time steps to obtain second-order accuracy. Thus, the additional flux in the momentum equation is approximated by:

$$\dot{m}^{n+1}_{mesh_{pn}} = \rho \frac{3\delta v^{n+1}_{pn} - \delta v^{n}_{pn}}{2\Delta t} \tag{A.14}$$

to be used in Eq. (A.7). This second-order approximation is used in the methods in chapters 3 and 4.

References

[1] R.W.C.P. Verstappen and A.E.P. Veldman. Symmetry-preserving discretization of turbulent flow. *Journal of Computational Physics*, 187(1):343–368, 2003.

[2] F. X. Trias, O. Lehmkuhl, A. Oliva, C. D. Pérez-Segarra, and R.W.C.P. Verstappen. Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured grids. *Journal of Computational Physics*, 258:246–267, 2014.

[3] Lluís Jofre, Oriol Lehmkuhl, Jordi Ventosa, F Xavier Trias, and Assensi Oliva. Conservation properties of unstructured finite-volume mesh schemes for the navier-stokes equations. *Numerical Heat Transfer, Part B: Fundamentals*, 65(1):53–79, 2014.