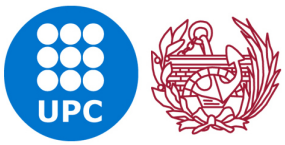
Towards the Virtual Wind Tunnel for civil engineering applications

Rubén Zorrilla Martínez

Ph.D. thesis in Civil Engineering

Thesis submitted as a compendium of publications



Universitat Politècnica de Catalunya

Departament d'Enginyeria Civil i Ambiental

Supervisors: Prof. Eugenio Oñate Ibáñez de Navarra

Prof. Riccardo Rossi

Barcelona, July 2020

A mi padre.

Abstract

This thesis develops a numerical tool (the Virtual Wind Tunnel, VWT) for the resolution of problems involving fluid flow around structures. Due to the limitations that traditional methods may have in this context, the VWT is based on the use of fixed mesh technologies (CutFEM-type) combined with an implicit representation of the embedded bodies.

One of the main contributions of the thesis is the use of such fixed mesh methods to solve lightweight thin-walled structures problems. Hence, two embedded formulations capable of representing the flow around bodies with and without internal volume are proposed. The first one results in a simpler implementation and lower computational effort but can only represent a slip behavior of the wall. The second one gets rid of such limitation by including a Nitsche imposition of the Navier-slip condition, thus allowing modelling any wall behavior as a wall law would do.

The applicability range of the VWT includes the fluid–structure interaction problem (FSI). To that purpose an improvement for the boundary condition imposition of the FM-ALE algorithm mesh motion problem is also proposed. Moreover, the implementation, which has been conceived to be easily extended to any other coupled problem, is also treated.

The validation of the technologies within the VWT includes multiple theoretical test cases as well as feasible industrial applications. Among these, the FSI analysis of a 4-point tent during a strong wind episode deserves to be highlighted as it showcases the achievement of the initial objective of the thesis.

Resumen

En esta tesis se desarrolla una herramienta numérica (el Virtual Wind Tunnel, VWT) para la resolución de problemas que involucran el flujo de un fluido alrededor de una estructura. Debido a las limitaciones que los métodos tradicionales pueden tener en este contexto, el VWT se basa en el empleo de técnicas de malla fija (tipo CutFEM) combinadas con una descripción implícita de los cuerpos embebidos.

Una de las principales contribuciones de la tesis es el empleo de dichos métodos de malla fija para resolver problemas de estructuras ligeras de pared delgada. Así pues, se proponen dos formulaciones embebidas capaces de representar el flujo alrededor de un cuerpo con o sin volumen interno. La primera de ellas resulta en una implementación más sencilla así como en un menor coste computacional pero únicamente puede reprentar un comportamiento deslizante de la pared. La segunda elimina esta limitación incluyendo una imposición mediante el método de Nitsche de la condición de Navier-slip, permitiendo así modelar cualquier comportamiento del mismo modo que lo haría una ley de pared.

El rango de aplicabilidad del VWT incluye el problema de interacción fluido–estructura (FSI). A tal propósito se plantea una mejora para la imposición de las condiciones de contorno del problema de movimiento de la malla del algoritmo FM-ALE. Asimismo, también se hace especial hincapié en la implementación, que ha sido concebida para ser fácilmente extensible a cualquier otro problema acoplado.

La validación de las tecnologías implementadas en el VWT incluyen múltiples casos teóricos así como posibles aplicaciones industriales. Entre éstas se destaca el análisis FSI de una *4-point tent* durante un episodio de viento severo ya que demuestra la consecución del objetivo inicial de la tesis.

Acknowledgements

First of all, I would like to acknowledge the Spanish Minister of Science, Innovation and Universities (Ministerio de Ciencia, Innovación y Universidades) and the International Center for Numerical Methods in Engineering (CIMNE) for the financial support during these four years. In this regard, the International Graduate School of Science and Engineering (IGSSE) of the Technische Universität München (TUM) is also acknowledged. Furthermore, I also thank the Oskar von Miller Forum (OvMF) for the accommodation during my international stay.

My gratitude extends to my two directors. I thank Prof. Eugenio Oñate for allowing me to work in such an interesting topic and for the confidence in me to achieve the objectives of this thesis. This work would not have been completed without the help of Prof. Rossi. Riccardo, many thanks for always having time to invest on this. I hope this is just the very fist one of many upcoming research successes.

The entire Kratos team deserve to be acknowledged by its own merits. Among all the developers, I particularly thank Dr. Jordi Cotela, Dr. Eduardo Soudah, Dr. Antonia Larese, Dr. Pooyan Dadvand, Carlos A. Roig (aka Charlie), Philipp Bucher and (soon Dr.) Vicente Mataix. Jordi, thanks for the valuable discussions (possibly lessons) about fluid dynamics and turbulence as well as for the coding advises. I also thank you for the fun during my stay in Munich. Edu, thanks for giving me the chance of applying my research in the biomedical field and for constantly try to find new opportunities for the group (by the way, you still owe me some hours of conversation). Antonia, thanks for all the help since I became your student for the first time. I want to also thank the KratosCore-man Pooyan for the enormous help while developing the level set algorithms as well as for the C++ and design patterns guidelines. The guy who silently makes everything work must be acknowledged too. Charlie, thanks for all the help with

the compilation and debugging weird issues. I also thank the guy who has most probably reviewed the entire Kratos many times to force all of us to do things properly. Philipp, thanks for all the code reviews and brilliant suggestions. At this point, I must also thank the coolest office mate I ever had. Vicente, thanks for allowing me to constantly tease you, for the laughs during the tough moments and for being always disposed to help. I want to also thank the rest of CIMNE staff, specially the GiD team and the rookies that brought fresh air to the office a few years ago.

My gratitude also goes for the Bavarian colleagues in the Statik Chair. I personally thank Prof. Roland Wüchner for all the help and the guys of the Raum 0101.Z1.026, which I can confirm is the loudest office in the Chair. Thanks Andreas, Altug, Íñigo and Tobi for making me feel like at home during my stay.

Furthermore, I would like to express my gratitude to Prof. Miguel Cervera for the confidence in me to lecture one of his subjects. In this regard, I also acknowledge my teaching colleague Alejandro Cornejo as well as my colleague in the Department Gabriel Barbat.

There are people that has contributed to make this thesis possible without realizing. Here I want to thank my friends Toni (aka Kaborza), Ana, Enric (aka Epi), Eva María (aka Kpo), Dani Rey, Laura Cuni and Carlos (aka Carlutus). I also thank my long term teammates Víctor, Ortuño, Samu (aka Chino), Fer and Javi, as well as the last seasons joins Nachete, Nacho (aka Bibi), Andreu, Antonio and Lluís. These lists lack the important former teammates Álex, Adri, Joaquín (aka Maka), Bistu, Aitor and Manolo. All of them made me (temporally) forget about the troubles that raised during the prosecution of this thesis.

Last but not least, I want to also thank my family. Gracias Elena por acompañarme durante estos años y conseguir siempre hacerme reír, especialmente en los momentos más duros. Agradecer a mi madre María Luisa Martínez por su amor incondicional y por constantemente cuidar de nosotros. Gustaríame tamén dedicar unas palabras a o recordo da miña avoa Hortensia. Finalmente, mi más sentido agradecimiento es para mi padre Juan Francisco Zorrilla. Gracias Papa por enseñarme como luchar frente a las adversidades, por siempre confiar ciegamente en mí y por constantemente empujarme a dar un paso más allá. Esta tesis es una consecuencia de todo lo que he aprendido de ti.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Fluid dynamics. A brief historical survey

Since the ancient era, humans have tried to tame the fluids for their own benefit. In the very beginning, this was limited to take profit of the wind thrust for navigation purposes. The earliest evidence of the use of a sail appears on a painted ceramic disk dating between 5500 and 5000 BC which was found in the current Kuwait. This discovery suggests that primitive wind sailing was already known in the Ubaid 3 period of the Mesopotamian empire [15]. Later on, the watermill appeared during the Hellenistic period (early 3rd century BC) in Byzantium and Alexandria to harness power from water [79, 109].

These technologies kept evolving during centuries. Hence, the advances in sail navigation made possible the goods commerce as well as the culture exchange among all the civilizations in the Mediterranean sea and, after the 15th century, around the world. Watermills also evolved, to be more efficient or to become windmills, which appeared in the current Iran in the mid 9th century to generate power from the wind [68]. Although these are only two examples it is clear that these and other fluid-related technologies helped to write the history of mankind.

Furthermore, the technological advances nourished the interest on the theoretical comprehension of the mechanics of fluids. In this context, the Archimedes' principle (250 BC) can be considered as the very first theoretical milestone, which became the base for the hydrostatics studies of the ancient Greek philosophers. A few centuries later, the middle age Islamic physicists took the lead on this. This allowed them to reach a deep comprehension about the hydraulics of many engineering structures.

The breakthrough into modern fluid dynamics came in the 16th century from the hand of Leonardo da Vinci (1452-1519). Based on the observation of nature, da Vinci achieved an empirical understanding of some of the most complex fluid dynamics phenomena. Some of these are the first description of a pressure lift force, which can be understood as the first theory of flight, or the first explanation of turbulence, which he did from the observation of the water flow patterns that generate behind obstacles in rivers.

Nonetheless, the 17th and 18th centuries can be considered as the most prolific times for the theory of fluid dynamics thanks to the contributions of many brilliant mathematicians. Among all of them, Isaac Newton (1643-1727), Daniel Bernoulli (1700-1782), Leonhard Euler (1707-1783) and Jean le Rond d'Alembert (1717-1783) stand out. Isaac Newton settled the foundations with his well-known laws of motion and studies about the friction and viscosity (*Principia*, 1687). Later on, Daniel Bernoulli proved that the gradient of pressure is proportional to the acceleration of the fluid in his theory of the motion of fluids (*Hydrodynamica*, 1738). In 1757 Leonhard Euler posed his famous Euler differential equations for inviscid flows. Just one year later, Jean le Rond d'Alembert proved that the drag of a body of any shape moving through a fluid with no viscosity is zero (1758). This phenomenon was known from then on as the d'Alembert paradox.

During the 19th century the effort was put on adding a friction term to Euler's equations in order to overcome the d'Alembert paradox and obtain the results observed in nature. This was achieved in 1822 by Claude-Louis Navier (1785-1836), who proposed the addition of an extra molecular viscosity term. Although several authors published works in this regard, it took two decades to find the first mathematical reversible derivation of the new viscous Euler equations, which was presented by George Gabriel Stokes (1819-1903) in 1845. Hereafter, these equations were known as the Navier-Stokes equations. In 1851 George Gabriel Stokes also derived the well-known Stokes' law for the calculation of the drag force exerted on spherical objects in high viscous flows.

Viscosity and turbulence have been possibly the major theoretical concerns of contemporary fluid dynamics. In 1883 Osborne Reynolds (1842-1912) demonstrated with his famous experiment how the transition from laminar to turbulent flow occurs. 20 years later, Ludwig Prandtl (1875-1953) presented his viscous boundary layer theory (1904) which has been the basis for most of the aerospace technological advances of the past century [69]. Besides these, the contributions of Theodore von Karman (1881-1963), Geoffrey Ingram Taylor (1886-1975) and Andrey Nikolaevich Kolmogorov (1903-1987) aiming at understanding the turbulence also deserve to be mentioned.

The eruption of the World Ward in the beginning of the 20th century came with an astonishing evolution of the aerospace engineering, which promoted the construction of experimental wind tunnel facilities. The wind tunnels proliferation continued until the end of the World War II and allowed scientists and engineers to properly visualize and accurately measure the flow passing through objects for first time in history. This capability turned into a deep empirical knowledge that made possible the amazing aerodynamic advances of the past century.

Although numerical methods were used from long time ago, it was the apparition of modern computers during the second half of the 20th century what effectively enabled the virtual resolution of real-life problems by using computational techniques. Hence, the constantly growing computational power paced the apparition of a widespread range of numerical techniques since the 50s. It was however during the 60s and the 70s when the aircraft industry, mainly Boeing and NASA, effectively drove the Computational Fluid Dynamics (CFD) forward by developing numerical techniques to solve simplified inviscid models, namely the potential and Euler equations. Thanks to this, CFD methods could be applied for the first time to the resolution of real engineering problems. During the 80s and the 90s CFD technologies advanced further on and achieved the resolution of the complete Navier-Stokes equations, helping CFD methods to populate other engineering fields such as the automotive, biomedical or civil engineering.

In addition, during the early 80s commercial CFD codes came into the open market to become one of the daily used Computer-Aided Engineering (CAE) tools. Nowadays, commercial and open-source CFD codes provide technicians, designers and researchers with a virtual wind tunnel toolbox on their own laptops. Thus, CFD have become in an indispensable part of any design process involving fluids, making possible the development of technologies that have helped to achieve the all-times most rapid mankind evolution.

1.2 The fluid–structure interaction problem

From the previous section, it can be concluded that the comprehension of the interaction mechanisms that appear after the immersion of a body into a fluid has been one of the most important scientific challenges of the past centuries. In a wide range of applications, such as the hydrostatic analysis of massive civil engineering structures or the laminar flow analysis in open water channels, the fluid dynamics analysis can be efficiently

achieved without considering the deformation of the "wet" bodies. However, there are lots of engineering and natural phenomena in which such simplification is no longer valid, leading to the so called Fluid–Structure Interaction (FSI) problems.

At this point, it is important to remark that the resolution of a fluid dynamics problem with and without considering such fluid–structure interaction mechanisms can lead to a completely different response. Hence, the FSI problem can be somehow considered as the union of a fluid dynamics and a solid mechanics problems such that the solution of each one of them is linked to the one of the other. This mutual dependency comes from the fact that the structure is deformable under the action of the fluid load. At the same time, the structure deformation implies a modification in the fluid problem geometry, which turns into a change in the load distribution over the structure.

Nowadays this might seem kind of obvious, but uncontrolled FSI is behind some of the most famed engineering failures of the modern era. For instance, fluttering, which is a dynamic instability that causes self-oscillation of fluid-surrounded slender bodies, is understood to be the main cause of the Tacoma Narrows bridge collapse in 1940 as well as the Braniff International Airways flight 542 crash in 1959.

Furthermore, there also exists lots of natural phenomena, such as the flapping of animal wings or the hydrodynamics of fish fins that are indeed FSI problems [37, 105, 113]. The study of such animal motion mechanisms from a computational perspective [76, 116] is of great value for the development of novel technologies such as newfangled unmanned vehicles. Such natural phenomena also include many of the human body processes, mainly those related to the circulatory and respiratory systems. The numerical simulation of these has helped the clinicians in the decision making in front of cardiovascular [41, 93, 104] and airways [62] diseases, in the development of new prosthetic devices such as artificial heart valves [29, 31, 58], and more recently, in the study of the mechanics of blood cell motion [16].

More specifically, there are also lots of civil engineering structures that are affected by the FSI problem. These can be basically summarized in thin-walled structures [48], namely membranes and shells, and slender structures such as bridge decks [99], chimneys, elevated tanks that suffer from sloshing during earthquakes [56, 81] or the blades of wind energy harvesting mills [95].

In some cases, the FSI effects can be efficiently reproduced using reduced order models [65] (e.g. water hammer in fluid ducts). However, such simplifications are no longer possible in the vast majority of applications since the interaction mechanisms involve

a three-dimensional behavior. This thesis focuses on the latter, particularly in those problems that involve lightweight thin-walled structures.

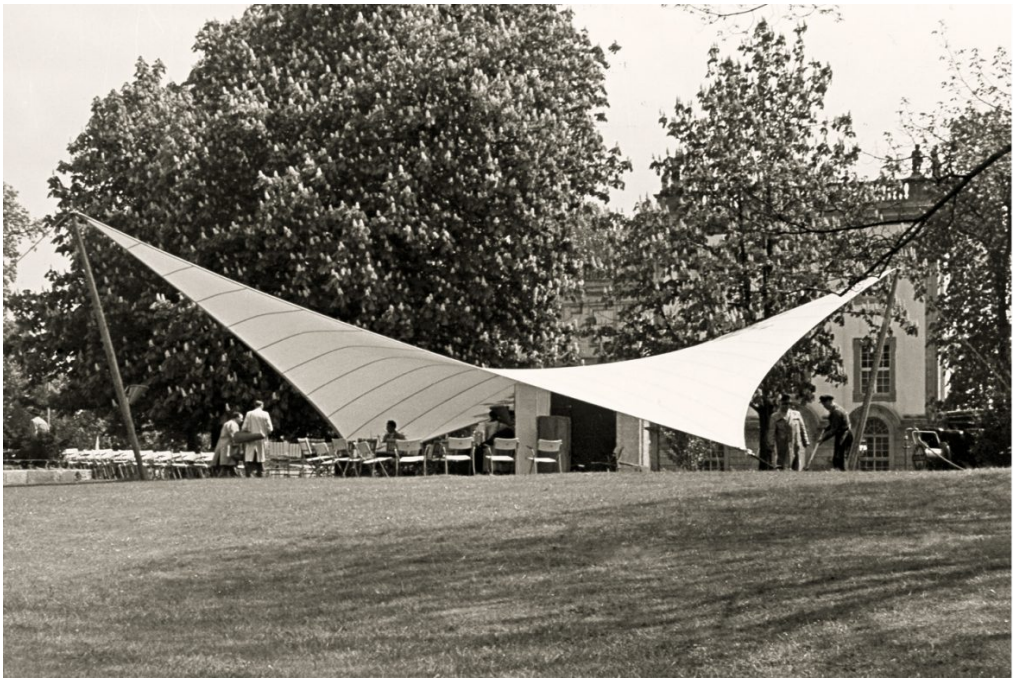
1.3 Lightweight thin–walled structures

Since their appearance in the mid past century, the interest in the use of highly flexible thin–walled structural solutions has been constantly growing. In this regard, the German architect and engineer Frei Otto (1925-2015) must be acknowledged as the pioneer in the design of lightweight structures. Under the idea of creating more sustainable structures in a world with resources shortages, Frei Otto abandoned the traditional structural design in favour of more efficient and creative solutions based on the use of metal reinforced tensile membranes and cables [82]. Although this new technology was initially used for basic structures, such as the Musikpavillon in the 1955 Kassel Federal Garden exhibition (Fig. 1.1a), their advantages rapidly became apparent and encouraged their use as permanent solutions for the roof covers of the 1967 Montreal World Expo German pavilion and the 1972 Munich Olympic Stadium (Olympiastadion München) (Fig. 1.1b).

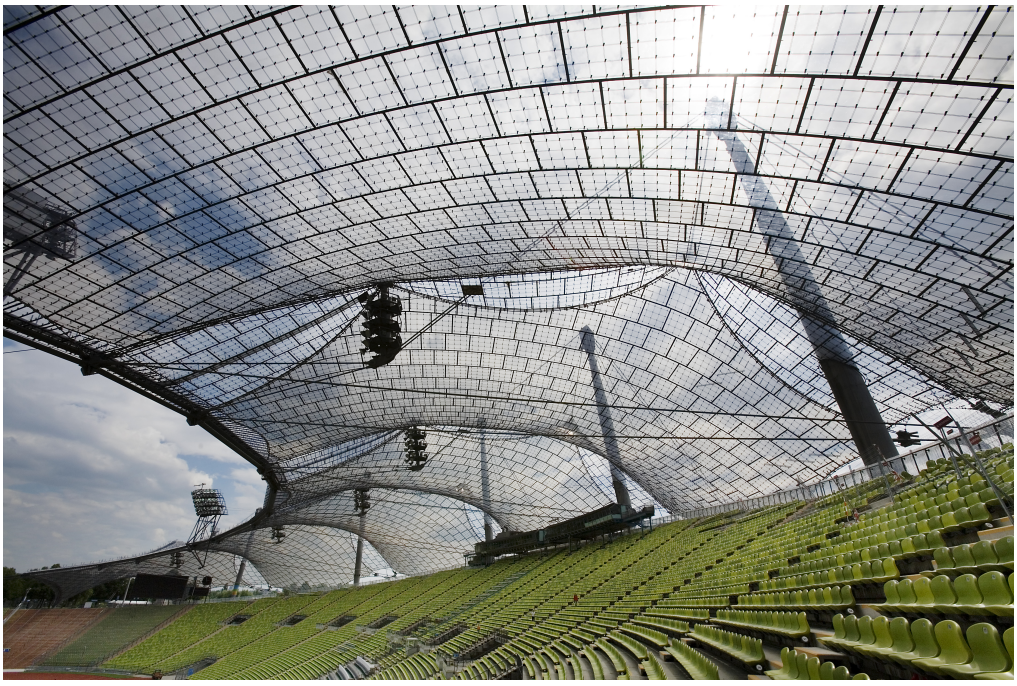
The pioneer understanding of how structures must be conceived that Frei Otto had impinged the current structural design criteria, which have evolved to use of much more efficient materials such as reinforced textile composites. Even though such extremely lightweight structures can be also employed as roof covers or shadowing structures (Fig. 1.2), their versatility and resource-saving manufacturing opens a wide variety of novel applications.

For instance, these can be used for the creation of temporary rapid deployment inflatable structures that can serve as emergency shelters, exposition pavilions or hangars (Fig. 1.3) [80]. Furthermore, they can also be used for the creation of foldable or moving structures capable of adapting their shape according to the environmental conditions (Fig. 1.4). Last but not least, there are other problems out of the civil engineering scope, such as the optimization of sail navigation or the design of parachutes, that could directly benefit from the research on this field.

It is clear that the structural safety analysis of this family of structures depends hugely on the characterization of the mechanical properties of the textile materials they are composed of. Their response is also strongly influenced by the wind load they are subjected to.



(a) Kassel Federal Garden exhibition Musikpavillon (1955).



(b) Olympiastadion München interior view (1972).

Figure 1.1: Evolution of lightweight structures by Frei Otto.

Figure 1.2: Textile roof cover in the airport of München.

Normally, the structural integrity under the action of wind loads is ensured by the accomplishment of the design codes requirements. Unfortunately, these newfangled structural solutions are commonly out of the scope of the standards, and thus require alternative ways to guarantee safety and reliability. Moreover, they are prone to suffer from aeroelastic effects, which should definitively be considered in the response assessment, owing to their extremely flexible nature.

Even though wind tunnel facilities have been of great benefit to characterize the flow around flexible objects, they have some inherent drawbacks. On the one hand, wind tunnels involve enormous construction and operation costs that compromise their economic viability. This turns into an increased cost per experiment, which restricts their use to reference research centers or big private companies. On the other hand, depending on the dimensions of the problem and of the ones of the wind tunnel, the experiment may require the use of a reduced scale model. In some applications this might compromise the quality of the results as it is not always possible to properly reproduce the scale effects in the reduced experiment in such a way that the aeroelastic behavior of the real structure is represented. In this context, numerical simulation arises as a feasible alternative to overcome these limitations, which surely appear in the FSI wind analysis of lightweight structures.

(a) Exterior view.

(b) Interior view.

Figure 1.3: Inflatable 75m span hangar in Saudi Arabia (courtesy of Buildair).

(a) Shading structure closed.

(b) Shading structure opened.

Figure 1.4: Convertible shading roof in the Prophet's Mosque in Medina. In summer the umbrellas open to provide shade while in winter they do so to prevent the day warmth from escaping (courtesy of SL Rash GmbH).

1.4 Objectives

The main objective of this thesis is the development of a numerical tool, henceforth the Virtual Wind Tunnel (VWT), for the reliable resolution of FSI numerical experiments. It has to be emphasized that the VWT must be capable of dealing with any type of structure, namely volumetric and membrane-like bodies.

Considering that the target application is the resolution of extremely flexible structures under the action of severe wind loads, the VWT must be robust enough to cope with extremely large displacements and/or rotations. On top of this, there might be cases in which the analysed bodies can suffer from self-contact and wrinkling.

In this context, the traditional Arbitrary Lagrangian-Eulerian (ALE) approaches that have been traditionally used for similar purposes are prone to suffer from element deterioration. Hence, rather than focusing on the Computational Solid Mechanics (CSM) problem, this work focuses on the developing of embedded Finite Element (FE) methods for the robust resolution of CFD and FSI problems involving thin-walled structures. It is pointed out that efficiency is crucial for the effective technology transfer of the VWT to real engineering applications, in which the computational times must be in accordance to the project development ones. In consequence, all the techniques in this work take this as an essential requirement.

At the end of this thesis, the VWT tool will help to

- extend the use of CFD and FSI simulations to the civil engineering field as it already occurs in other engineering disciplines (e.g. aerospace engineering)
- study the aeroelasticity phenomena that occur in extremely flexible structures that cannot be tested in real wind tunnel facilities. The better understanding of these during the design phase would allow optimizing the geometry and materials of the structure, thus reducing the construction overall cost and material consumption.
- optimize and simplify the design process by reducing the amount of required wind tunnel experiments, something that should turn into a more efficient project development.

Last but not least, it is due mentioning that all the implementations of this work must remain as open source software. In order to fulfil this requirement the VWT will be developed within the Kratos Multiphysics open source framework [27, 28].

1.5 Contents

First of all, it has to be clearly stated that this document is conceived as a compendium of publications. Therefore, three of the chapters are made as a brief introduction to the corresponding paper and the postprint submitted to the journal.

Chapter 2 presents the state of the art on fixed mesh CFD methods and FSI coupling. In Chapter 3 the quasi–incompressible Navier–Stokes stabilized formulation used throughout the thesis is derived. Chapter 4 is the first paper of the compendium and presents a discontinuous embedded formulation for the CFD analysis of slip thin-walled bodies. This formulation is enhanced in Chapter 5 to deal with both stick and slip thin-walled bodies. Chapter 6 is the second paper of the compendium and validates the formulation described in Chapter 5 with data from *in vitro* biomedical experiments. The extension of the embedded CFD solver to FSI problems is studied in the third paper of the compendium, which is collected in Chapter 7. In Chapter 8 embedded CFD methods are exploited to efficiently deal with extremely large wind engineering problems. Finally, the conclusions and further work lines are discussed in Chapter 9.

Chapter 2

State of the art

This chapter reviews the state of the art on the use of fixed mesh techniques in CFD and FSI problems. The first section describes how this family of numerical methods can help to efficiently solve the VWT target applications. Secondly, the implicit representation of geometries is briefly discussed, particularly focusing on the discontinuous level set representation of membrane-like bodies. The third section reviews the characteristic features of each one of the literature fixed mesh approaches in order to choose the suitable one for the problems at hand. The forth section reviews the literature on the chosen alternative. The fifth section presents the common issues that appear when fixed mesh methods are applied to moving interface problems. Finally, the last section reviews the coupling techniques to approach the FSI problem, specially that one applied in this thesis.

2.1 Fixed mesh methods. Motivation

This thesis aims at creating a framework for the FSI analysis of any type of bodies, with a particular emphasis on the lightweight thin-walled structures. Taking into account that these are highly deformable structures, the traditional Arbitrary Lagrangian-Eulerian (ALE) methods [36] might lack robustness since the mesh motion problem tends to yield degenerated elements, namely highly distorted or even inverted, when large displacements and rotations of the boundaries appear. Moreover, problems involving thin-walled bodies might also fall into topology changes if self-contact or wrinkling appears. Having said this, it can be easily guessed that the selection of the fluid-structure in-

terface tracking technique becomes crucial for the robust and efficient resolution of the problems at hand. Taking this into account, mesh-based approaches can be classified according to how the analysed bodies are represented in body conforming (also known as body fitted) and non-conforming approaches.

Even though body conforming purely Lagrangian methods such as the Particle Finite Element Method (PFEM) [43, 94] are a robust alternative for the interface tracking in moving boundary problems, they rapidly become prohibitive as remeshing is required at each time step. Alternatively, the CFD domain can be meshed regardless the analysed objects to decouple the computational mesh (henceforth denoted as background mesh) from the immersed geometries, which are no longer represented by the boundaries of the mesh. Instead, the analysed bodies are implicitly characterized on the background mesh from their skin representation by using techniques such as the level set method [101].

Although this introduces an extra complexity to the problem, the implicit representation of boundaries becomes advantageous for the interface tracking. Hence, all the potential mesh updating issues that may appear under the presence of large boundary displacements and rotations are bypassed since the ALE mesh motion problem is substituted by a simple update of the implicit representation of the bodies. On top of that, topology changes are naturally dealt with without requiring any operation in the background fixed mesh.

Furthermore, the use of an implicit representation also optimizes the simulation pipeline since the preprocess becomes much more efficient. On the one hand, it avoids all the manual repairing operations such as the removal of holes, overlapped surfaces or duplicated entities that are required to create a volume mesh from a "dirty" input geometry. These operations, which can amount up to the 50% of the total simulation time for real engineering cases [26], can be completely bypassed as the geometrical defects are filtered by the level set calculation algorithm, which is a more robust and effective operation [66, 67]. On the other hand, it is also known that the creation of a volume mesh from a thin-walled geometry is a rather troublesome operation, specially when complex curved geometries are involved. This is because the original membrane geometry needs to be duplicated in order to generate the volume mesh. These two geometrical entities tend to intersect each other when working in the machine precision limit, thus leading to a failure of the meshing operation. Using an implicit representation of the volumeless immersed bodies automatically avoids such limitation.

2.2 Level set representation of immersed geometries

At this point, it is clear that using a level set representation is of great utility in some applications. This section briefly describes the basics of implicit representations and how they can be applied depending on the volumetric nature of the analysed geometries.

The main idea behind any level set method relies on the calculation of a signed distance function such that its zero isosurface matches, as closely as possible, the skin of the analysed bodies. Even though all the level set algorithms are based on the intersection between the background and overlapped skin meshes, they can be classified according to their treatment of the volumetric nature of the bodies. Hence, those bodies that feature a well defined internal volume are typically represented using a continuous level set algorithm (Fig. 2.1). In this case the distance is computed node-by-node and the inside/outside concept is treated using computer graphics techniques such as the raycasting [91]. These methods always return a smooth and continuous representation of the immersed bodies.

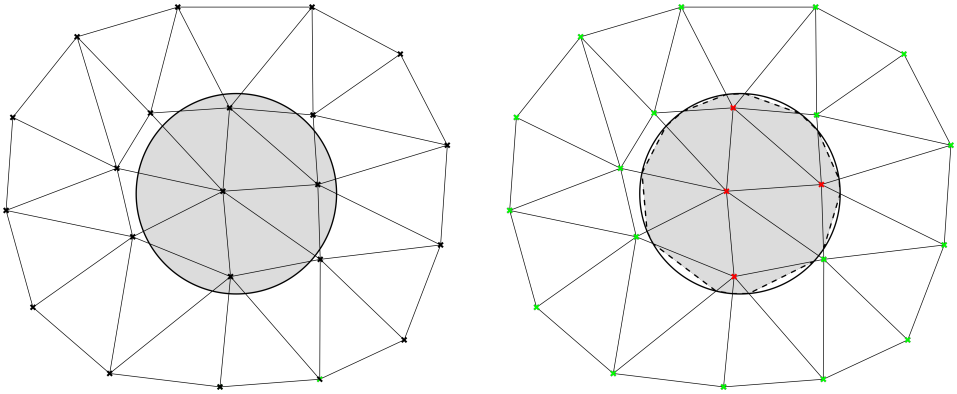


Figure 2.1: Continuous distance function. Body with a well-defined internal volume (left) and continuous distance representation (right). Dashed lines represent the continuous distance representation.

Unfortunately, these methods are not valid for membrane-like bodies as the inside/outside concept becomes meaningless in this case. In consequence, volumeless bodies cannot be represented in terms of a nodal-based distance function. Nevertheless, it is possible to describe these in terms of an element-based level set. As it can be observed in Fig. 2.2, such capability comes however at the price of having a potentially discontinuous distance function, meaning that the same node can have different distance sign and value depending on the element considered. This is exploited in the subsequent chapters, as

well as in [119–122], to model a wide variety of thin-walled structures such as boat sails, lightweight membranes or thin biological tissues.

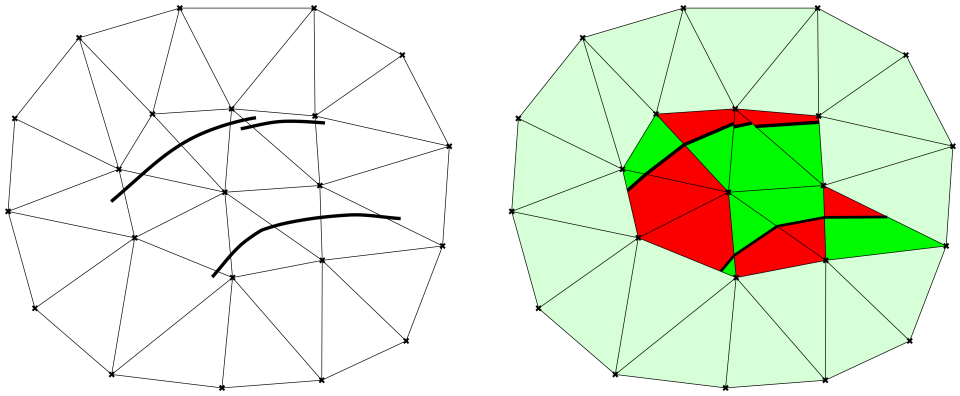


Figure 2.2: Discontinuous distance function. Body without internal volume (left) and discontinuous distance representation (right). Red and green portions of the cut elements indicate the positive and negative discontinuous distance regions. Light green denotes the non-intersected elements.

2.3 Level set methods for CFD and FSI problems

Prior to any further discussion, it is interesting to mention that the first FSI numerical approach was based on a non-conforming mesh method. It was presented in the early 70s by Prof. Peskin and relies on the addition of an artificial body force to consider the immersed structure motion into the fluid problem [83].

Introducing the use of a level set-based representation of the immersed geometries opens the possibility to use a wide variety of methods such as the eXtended Finite Element Method (xFEM) [1, 42, 97, 98], the Immersed Boundary Method (IBM) [59, 66, 67, 85, 114, 115], the Embedded Boundary Method (EBM) (also known as Cut-FEM) [14] or the Shifted Boundary Method (SBM) [70, 71]. Regardless this thesis only considers traditional FE discretizations, the novel Immersogeometric approaches [58, 112] that combine exact geometries and embedded formulations also deserve to be mentioned.

Table 2.1 compares the aforementioned methods according to different criteria. The first one is the computational effort in terms of locality and matrix graph preservation, according to which xFEM is the most expensive technique as it requires the introduction of extra Degrees Of Freedom (DOFs) associated with the FE enrichment unknowns. Aside of introducing the need for blending elements, which turns into a more expensive

distributed memory assembly because of the communication overhead, this also requires the system matrix graph to be reconstructed each time the level set is updated.

The second criterion compares the BC imposition. In this regard, the IBM is possibly the most straightforward approach, as it allows direct imposition of the BC. However, this is not done over the interface but on its closest dry nodes, something that makes the method prone to suffer from mass conservation issues. The SBM method aims at overcoming this limitation by imposing a modified boundary value that takes into account the distance between the interface and the closest dry nodes where the BC is enforced. Contrariwise, the xFEM and the EBM rely on a weak BC imposition. This is normally done using variational techniques such as the penalty method [111, 119], the Nitsche method [21, 72, 73, 78, 106, 110] or the Lagrange multipliers method [8]. Although the weak imposition entails an extra complexity, that might turn into accuracy and stability issues, it allows to consistently enforce the BC over the level set intersections, which are the best non-conforming representation of the immersed geometries that can be implicitly achieved.

Last but not least, the table compares the capability of the methods to be applied to membrane-like bodies. As it can be observed, any method that requires an interior (i.e. dry node) for the BC imposition becomes no longer valid for the simulation of thin-walled geometries, meaning that only the xFEM and the EBM can be applied for such purpose.

Taking into account that industrial CFD and FSI problems likely involve a huge number of DOFs, the chosen alternative must be as computationally efficient as possible. Considering that only the xFEM and the EBM fulfill the thin-walled bodies requirement, the more suitable method for the target applications is the EBM owing to the absence of blending elements and the ability to keep the initial system matrix graph during the entire simulation.

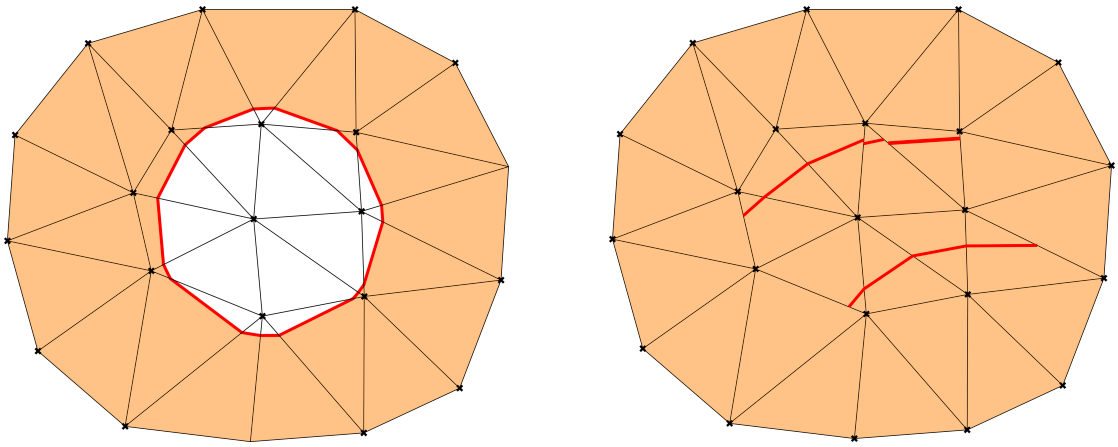
Table 2.1: Non-conforming mesh methods comparison.

		xFEM	IBM	SBM	EBM
Computational effort	Local	No	Yes	Yes	Yes
	Keeps matrix graph	No	Yes	Yes	Yes
BC imposition	Direct	No	Yes	Yes*	No
	over Γ	Yes	No	No	Yes
Thin-walled bodies		Yes	No	No	Yes

2.4 The Embedded Boundary Method. An overview

After having selected the EBM as reference fixed mesh approach, this section briefly reviews its basis as well as the different alternatives that can be employed for the weak imposition of the level set BC. To ease the discussion, from now on the fluid computational domain is denoted as Ω_f while the immersed objects skin representation is denoted as Γ_f.

The first distinguishable feature of the EBM is that the computational domain is not conformed by the element faces but by the intersections of the level set function with the background mesh elements. Aside of being a more accurate representation of the analysed bodies, this enables the application of the method to thin-walled bodies. This is schematically depicted in Fig. 2.3, which recovers the previously presented volumetric (Fig. 2.1) and volumeless (Fig. 2.2) toy examples to show their corresponding embedded computational domain.

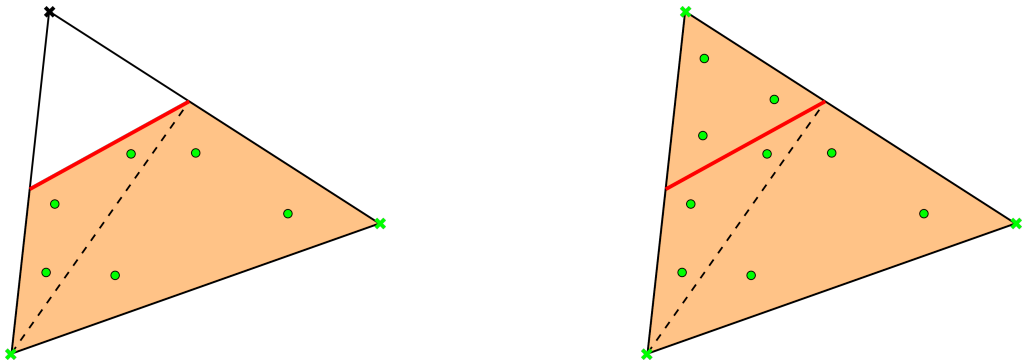


(a) Continuous level set case. (b) Discontinuous level set case.

Figure 2.3: EBM sketch for the examples in Figs. 2.1 and 2.2. Light orange denotes the active computational domain (Ω_f). Red lines denote the level set intersections (Γ_f) that represent the embedded skin.

From the previous images it can be deduced that an auxiliary splitting operation is required in each intersected element to perform the integration properly. As it is depicted in Fig. 2.4, the division of the parent geometry in a set of smaller ones allows relocating the integration points such that only the "wet" part of the cut elements is integrated. The same procedure is equivalently applied to thin-walled bodies (Fig. 2.4b) to consistently integrate both ("wet") sides of the membrane bodies.

Once the volume integration has been performed, the fluid BC needs to be imposed over the either continuous or discontinuous level set intersections that conform Γ_f (highlighted in red in Figs. 2.3 and 2.4). Although it is possible to create new nodes from the splitting pattern to strongly impose the BC [66], this operation is commonly done in a weak (i.e. variational) sense. There is a wide variety of methods that can be used for such purpose, leading to different flavors of the EBM. In the following a brief review of all the approaches that have been successfully applied to CFD problems is presented.



(a) Continuous level set case. (b) Discontinuous level set case.

Figure 2.4: EBM single element sketch. Light orange denotes the active part of the element belonging to Ω_f. Red lines denote the level set intersections Γ_f that represent the embedded skin. Black dashed lines represent the auxiliary divisions for the integration. The black cross markers represent the outside nodes. Light green cross markers represent the fluid nodes. Light green round markers represent the integration points.

The most extended approach to weakly enforce the BCs in the context of EBM methods is possibly the Nitsche method [57, 78], which has been applied for the imposition of the stick (no-slip) condition in both Stokes [72] and Navier-Stokes [58, 73] problems. The modified Nitsche method for no-slip conditions in [21] should be also acknowledged because of its simplicity and remarkable good performance. Regarding the slip wall behavior, the Nitsche method is used in [106] to enforce a pure slip condition. In [110] this is extended to the Navier-slip condition.

The Lagrange multipliers method can be alternatively used to impose Dirichlet BCs [46, 47]. In [106] a Lagrange multiplier approach for the imposition of the pure slip condition in Stokes flows is presented. Furthermore, the Lagrange multipliers technique in [8], which is based on a static condensation, deserves to be highlighted after its ability to keep constant the number of DOFs.

Unfortunately, all the aforementioned EBM approaches require a well-defined internal volume. In consequence, none of the methods above can be directly applied for the analysis of thin-walled bodies, thus precluding the simulation of membrane and shell structures. This could be achieved in an IBM fashion by adding an artificial volume force body [83, 84, 86, 117]. However this kind of methods might lack accuracy and suffer from mass conservation issues in some applications. In [76] an EMB ghost cell method able to work with any type of bodies (volumetric and volumeless) is presented. Nonetheless, such capability comes at the price of having a no longer purely local method as the BC imposition requires the element neighbors. Alternatively, the approach in [111] addresses the problem in a EBM-like manner by introducing a modified FE space in combination with a penalty imposition of the no penetration BC. An important advantage of this method is that the graph of the discrete stiffness matrix is preserved, something that becomes crucial when the level set function needs to constantly updated (i.e. FSI problems).

After having reviewed the literature on Cut-FEM approaches it can be noted that the volumetric problem is pretty well resolved. There is a wide variety of approaches that can efficiently and robustly deal with the problem at hand. Among these, the approaches in [21] and [110] are used in this work for the volumetric no-slip and slip (in a general Navier-slip sense) wall behavior modeling.

However, there is still room for the research on the application of the EBM to the resolution of CFD problems involving thin-walled bodies. More specifically, there is no purely local approach in the literature capable of representing the discontinuities arising from the immersion of membrane-like structures. The chapters 4 and 5 of this thesis aim at accomplishing such objective.

2.5 Fixed mesh methods and moving bodies

The enhanced robustness that fixed mesh approach have to deal with arbitrary large displacements and rotations comes at the price of some particularities that require to be taken care of. These are the small cut instability and the historical data initialization.

2.5.1 Small cut instability

As it can be easily guessed from its name, this problem is an instability that pops up when there is an ill-conditioned intersection between the level set and the background mesh. Even though the concept of ill-conditioned intersection varies depending on the fixed mesh approach used, it can be generalized to any intersection of the embedded skin that occurs extremely close to one node of the background mesh, which is equivalent to say that close to zero values of the level set are potentially ill-conditioned. It is important to note that no distinction is done yet between continuous and discontinuous level set functions, meaning that zero values of the distance function are always ill-defined. For those cases involving volumetric bodies represented by continuous distance functions, the previous definition can be restricted to those level set values close to zero that yield an almost empty element.

The small cut instability commonly appears when the interface moves across the background mesh. Nonetheless, it may also happen in problems involving a steady interface, specially in those ones involving poorly defined input geometries. In consequence, an auxiliary strategy to prevent this undesired situation is always demanded to robustly apply any fixed mesh method.

There are several approaches to prevent the convergence of the problem to be ruined by the ill-conditioned cuts. Some of these are the ghost penalty method [14] or the ghost cell method [5]. Although the stability and robustness of these methods is proved, they require the neighbors of the ill-conditioned elements to be applied. This adds a search operation that turns into an extra computational overhead, which might be particularly relevant when distributed memory platforms are used.

Alternatively, the ill-conditioned cuts can be avoided rather than controlled by using a level set quality check and correction algorithm. This approach starts from the idea that the level set is indeed a representation of the real geometry, whose quality is directly related to the background mesh element size. Taking this into consideration, the distance can be slightly modified by a small quantity depending on the element size without incurring in a noticeable error. For the continuous level set case, this means to avoid the "almost empty" intersections (highlighted in light orange in Fig. 2.5) by slightly moving the interface towards the positive distance side. Likewise, for the discontinuous level set case ill-conditioned cuts can be avoided by slightly modifying the close to zero elemental level set values (Fig. 2.6). This approach arises as a more computationally efficient (almost) purely local alternative that only requires one synchronization operation. The

capability of this apporach to robustly deal with the problems at hand is evinced in Chapters 6, 7 and 8, where it is applied to complex real-life geometries.

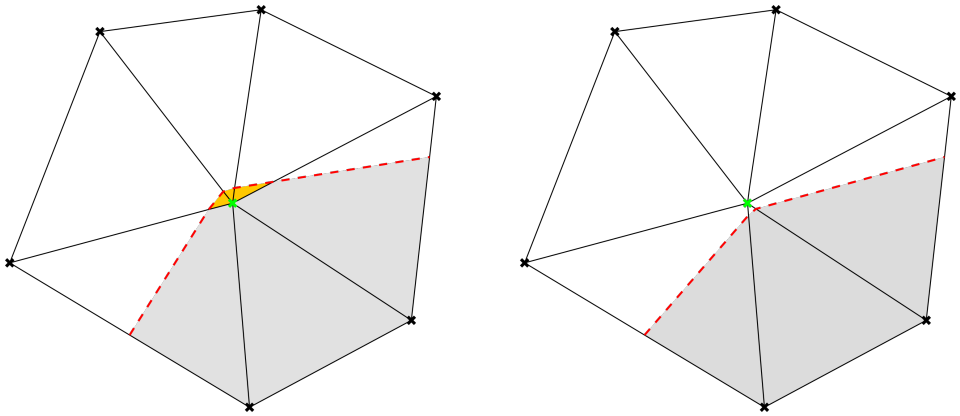


Figure 2.5: Continuous distance check and correction. Grey regions represent the computational domain Ω_f. Red dashed lines represent the level set intersections Γ_f. Before (left) and after (right).

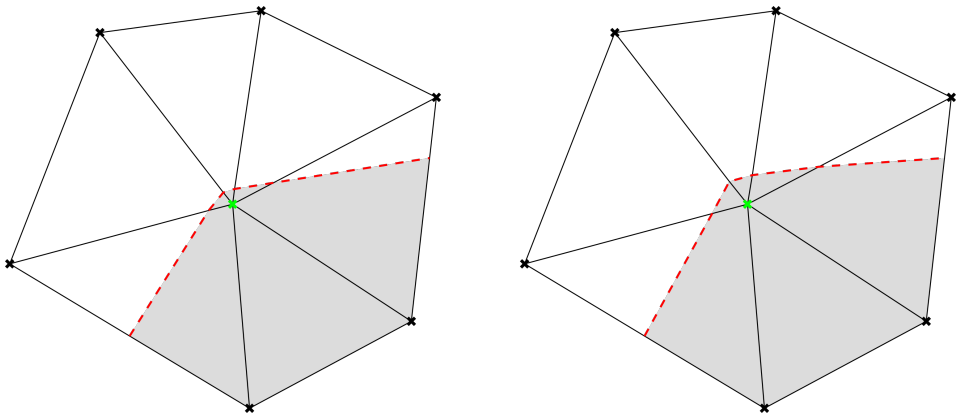


Figure 2.6: Discontinuous distance check and correction. Grey regions represent the computational domain Ω_f. Red dashed lines represent the level set intersections Γ_f. Before (left) and after (right).

2.5.2 Consistent initialization of values. The FM-ALE method

This issue appears when the level set function, regardless its continuous/discontinuous nature, evolves in time. As it is graphically described in Fig. 2.7, when the distance function modifies its spatial configuration the background mesh nodes that used to lie in one side of the level set might end up in the other one. In consequence, the historical

data of these nodes is no longer valid, thus leading to a wrong approximation of all the inertial terms involved in the problems.

It is due mentioning that no distinction is done between continuous and discontinuous distance functions, as the historical values need to be consistently updated in both cases. While in the continuous level set case (volumetric bodies) it is needed to consistently initialize the historical values in those nodes that change their status from "dry" to "wet", in the discontinuous level set case (volumeless bodies) it is required to do so as the historical data becomes meaningless as soon as the nodes move from one side to the other of the immersed thin-walled body. Having said this, it can be easily guessed that the proper treatment of this issue is crucial for the application of fixed mesh methods to any moving boundaries problem (i.e. FSI).

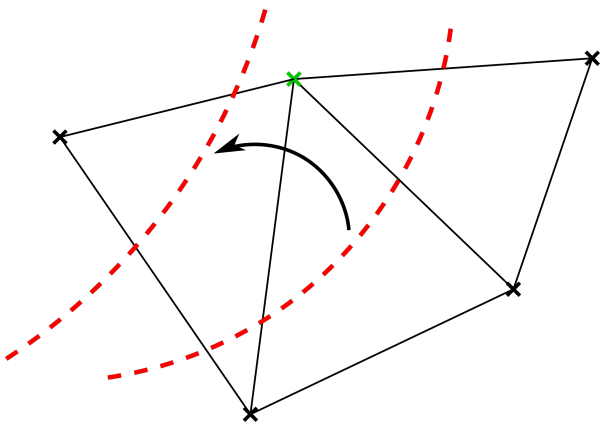


Figure 2.7: Embedded nodal initialization example. The node highlighted in green changes its position from one side to the other of the level set (red dashed line), which moves as indicated by the black arrow.

The consistent historical data initialization can be achieved in an ALE-fashion by using the Fixed Mesh–Arbitrary Lagrangian Eulerian (FM–ALE) method. The FM–ALE was firstly described in [22] to address this issue in moving boundaries embedded CFD problems. In [6] the authors extend the method to solid mechanics and FSI problems. Similarly, they do so for free-surface problems involving floating objects in [7, 9].

The main idea behind the FM-ALE method is to retrieve the consistent historical values from the so called virtual mesh, which is created as a copy of the background mesh used in the resolution of the problem. Each time the level set changes its spatial configuration, the virtual mesh is accordingly deformed by solving a mesh motion problem. Then all the historical values, as well as the mesh velocity, can be projected from the virtual

mesh to the background one. Once the projection is completed, the deformation of the virtual mesh is reverted back to the initial position.

It is due mentioning that this last step is the key of success of the method when it is used in presence of arbitrary large movements of the embedded interfaces. Hence, bringing back the virtual mesh to its original position each time the FM–ALE algorithm is applied implies that the mesh motion problem most likely involves small displacements and rotations, thus avoiding the common element distortion issues associated to body conforming ALE approaches. Besides that, reverting the mesh deformation also makes the problem presumably cheaper in terms of the computational effort.

2.6 Fluid-Structure Interaction

This section reviews the ingredients that are required to extend the embedded CFD solver to FSI problems. Hence, the different alternatives that can be selected to achieve the coupling are reviewed. The last part of the section focuses in detail on the particular choice of this work. To ease the discussion, the fluid and structure domains are denoted as Ω_f and Ω_s from now on. Similarly, their corresponding FSI interfaces are denoted as Γ_f and Γ_s.

2.6.1 Coupling schemes

For a coupled problem of any type it is required to satisfy the interface transmission conditions to guarantee that the coupling is effectively achieved. Such transmission conditions are of course problem dependent and thus need to be particularized depending on the physics to be coupled. For the FSI specific case these are the interface force equilibrium as well as the mass continuity, which demands that Γ_f and Γ_s spatial configurations match.

To ensure that the transmission conditions of the problem are accomplished, a coupling scheme is required. These can be classified according to how the transmission conditions are imposed in Dirichlet-Neumann (DN), Neumann-Neumann (NN) and Robin-type schemes (i.e. Robin-Robin (RR) or Neumann-Robin (NR)).

Owing to its proved well performance in the FSI field [63, 77, 107, 111], the DN coupling scheme is selected as reference coupling scheme in this thesis. In short, the DN scheme consists in enforcing a Dirichlet BC in one subdomain and a Neumann BC in the other

one. The Dirichlet condition is commonly applied in the lower density subdomain, which in the FSI case is likely Ω_f. In consequence, the Neumann BC is applied over Γ_s as a surface load whose value comes from the stress in the fluid boundary. For a thorough review on the stability and performance of the DN scheme, as well as on its interaction with several time discretization techniques the reader is referred to [89].

Furthermore, the implementation of the DN scheme is reasonably simple and does not require to modify the subdomain formulations. This last feature becomes crucial for the efficient application of non-intrusive coupling algorithms like the ones that are to be exploited in chapter 7 of this thesis.

Nevertheless, it has to be mentioned that there exist cases in which the DN may lack convergence or suffer from stability issues [61]. In this context, NN and Robin-type schemes arise as a feasible alternative to overcome these limitations. Although these applications are out of the scope of this thesis, the reader is referred to [4, 45] and [40, 64] for more information regarding RR and NR schemes.

2.6.2 Implementation of coupling schemes

The previously defined transmission conditions require a coupling strategy to ensure that they are satisfied in all the coupling interfaces. Such strategies can be classified according to how the coupled problem is solved in monolithic and partitioned (also known as staggered) strategies.

The idea behind any monolithic strategy is to put together all the subdomain contributions, together with the corresponding transmission conditions, in a unique large system of equations. This approach has been successfully applied in the resolution of body conforming FSI problems in [90, 94]. More related to this thesis is the method in [6], where the authors present a monolithic FSI technique based on a non-conforming discretization in both the fluid and the structure subdomains.

Although monolithic strategies are possibly the most robust approach for the resolution of strongly coupled FSI problems, they tend to yield poorly conditioned systems of equations, which might eventually impede the use of fast iterative solvers. Besides this, the implementation of a monolithic strategy is rather intrusive and likely requires the development of a new solver for the specific purpose of the coupling.

Partitioned strategies avoid these disadvantages by keeping a separated solver for each one of the subdomain problems, meaning that the fulfillment of the transmission condi-

tions is achieved by exchanging information between the coupled subdomains. Although the convergence rate of partitioned strategies might be suboptimal, they allow reusing already existing, possibly robust and thoroughly validated, specific solvers. This idea brings up the concept of black-box coupling [12, 35, 100], in which the coupling strategy is also requested to not interfere with the subdomain solvers more than to get and set information. Hence, the subdomain solvers are considered as a sort of black boxes that return a solution for the input data, that must be provided according to their Application Programming Interface (API).

In the framework of this thesis, this makes possible to directly use the already existent structural mechanics module of Kratos Multiphysics in order to focus the implementation effort on the development of the embedded CFD technologies and its extension to FSI problems.

Partitioned strategies can be further classified according to how many times the information is exchanged per time step. On the one hand, there are the loosely-coupled (or explicit) strategies in which the information is exchanged once per time step [38]. This is obviously the cheapest approach in terms of computational and implementation effort. However, they might lack convergence when there is strong interaction between subdomains or when the added mass effects are important [17, 55, 108].

On the other hand, there are the strongly-coupled (or implicit) strategies in which the information exchange occurs in an iterative manner until a certain convergence criterion is reached. Depending on how the information is used within each iteration, it is possible to distinguish between Jacobi and Gauss–Seidel strongly-coupled strategies. While in Jacobi-type iterations the information exchange occurs at the same time, allowing the parallel resolution of the subdomain problems, in Gauss-Seidel iterations this is done in a sequential manner, enabling the use of the latest information obtained in one subdomain to solve the other one within the same iteration.

The iterative nature of Gauss-Seidel approaches makes them more suitable for the target applications of this thesis. Although their convergence rate might be not as good as that of monolithic approaches, it can be astonishingly improved, specially when facing strongly coupled problems, by using them in combination with convergence acceleration algorithms.

Finally, it is interesting to point out that there exist some semi-implicit strategies that are somehow between the explicit and implicit ones. A very detailed review about several semi-implicit approaches can be found in [64].

2.6.3 Black-box interface residual minimization techniques

The choice of a Gauss–Seidel iterative strategy requires the definition of an interface residual function such that its minimization ensures the fulfilment of the problem transmission conditions. As the interface residual is completely problem dependent, it is assumed to be already defined for the discussion at hand. Hence, the literature review in this subsection focuses on the residual minimization rather than on its definition, which will be addressed in chapter 7.

As it is commented above, the use of a black-box coupling strategy implies restricted access to the subdomain solvers. The most straightforward approach that accomplishes with such requirement is a fixed-point iteration with relaxation. Although convergence can be achieved using a constant relaxation parameter, these methods are commonly used in combination with dynamic relaxation formulas. Among all of them, the second order Aitken scheme is possibly the most extended technique in the FSI community [63, 77] owing to its reasonably good performance and relatively easy implementation.

Alternatively, the interface residual problem can be considered as a vector of unknowns to be minimized. Hence, the problem can be posed as a Newton–Raphson (N–R) iterative procedure, leading to the so called Jacobian-based residual minimization techniques. Ideally, the exact Jacobian that minimizes the vector interface residual could be computed by accessing the subdomain solvers. However, this intentionally precluded by the selection of a black-box coupling strategy. Nevertheless, the exact Jacobian calculation can be avoided by using Jacobian-free Newton-Krylov (JFNK) [60, 62] or Quasi-Newton (QN) [12] methods.

Rather than approximating the complete interface Jacobian, JFNK methods are based on approximating its projection onto the iteration update vector by using a finite differences formula. Owing to the inherent non-linearity of the FSI problem, such projection needs to be linearized by introducing a small perturbation, which needs to be selected by the user. Although this issue can be addressed by using a formula to automatically do that [75], the optimal value for the small perturbation is still completely problem dependent.

In this context, QN methods arise as an alternative to avoid the selection of the perturbation constant. Unlike JFNK methods, QN approaches approximate the complete interface Jacobian by means of a linearized formula that takes the information from the previous FSI iterations. The different QN methods can be characterized according to how such Jacobian approximation formula is conceived [34].

Among all of them, the Broyden's iteration scheme should be acknowledged as the first QN method applied to FSI problems [75]. A more popular approach is the Interface Quasi-Newton with Inverse Jacobian from Least Squares model (IQN-ILS) [33, 34], as well as its block equations version (IBQN-ILS) [34]. In [34] the authors prove that the performance of the IQN-ILS overcomes that of Aitken and of Interface-GMRES schemes for all the reported examples. A very similar approach to the IQN-ILS is the MultiVector Quasi-Newton (MVQN) algorithm presented in [12]. Although the authors report that it converges slightly faster than the IQN-ILS, this comes at the price of requiring a matrix inversion, whose size equals the interface residual one, each time the Jacobian is approximated. This disadvantage has been recently overcome in [103].

Considering that QN approaches involve a rather large number of matrix operations, each QN iteration is possibly more computationally expensive than that one of a JFNK technique. Notwithstanding this, they are reported to be the most efficient Jacobian-free approach in terms of total number of iterations. Taking into account that the computational bottleneck in partitioned FSI problems is likely the resolution of the CFD problem, the reduction of the total coupling iterations is what effectively turns into a performance improvement. Taking this into account, the embedded FSI coupling Gauss-Seidel technique presented in chapter 7 will be based on the use of QN algorithms. More specifically, the MVQN is chosen as reference black-box convergence accelerator owing to its reported good performance.

Chapter 3

Quasi-incompressible Navier–Stokes stabilized formulation

This section presents the quasi–incompressible Navier-Stokes formulation that serves as basis for all the embedded CFD developments of this thesis. Firstly, the viscous incompressible Navier-Stokes governing equations as well as the inclusion an extra pseudo–compressibility term are discussed. Secondly, the FE discretization and the subscales stabilization technique are introduced. In the third section the variational form of the problem is derived together with some notes on its symbolic implementation. Finally, the correctness of the implementation is proven by the method of manufactured solutions.

3.1 The Quasi-incompressible Navier–Stokes equations

This thesis focuses on the resolution of CFD and FSI problems involving viscous fluids. Considering that the Mach number (Ma) of the target applications is below 0.3, the flow can be considered as incompressible and thus modeled with the well-known viscous incompressible Navier–Stokes (N–S) equations [36, 87].

In this context, the Cauchy stress tensor $\boldsymbol{\sigma}$ can be defined as

$$\boldsymbol{\sigma} = \mathbb{C} : \boldsymbol{\nabla}^s \mathbf{u} - p\mathbf{I} \tag{3.1}$$

where $\mathbf{u}$ denotes the velocity, p the pressure, $\boldsymbol{\nabla}^s$ the symmetric gradient operator, $\mathbf{I}$ the 2nd order identity tensor and $\mathbb{C}$ the 4th order viscous constitutive tensor.

Although any viscous constitutive relation could be used, only isotropic Newtonian fluids are considered in this thesis. Hence, the viscous constitutive tensor reads

$$\mathbb{C} = \lambda \mathbf{I} \otimes \mathbf{I} + 2\mu \mathbb{I} \tag{3.2}$$

where $\otimes$ denotes the tensor product and $\mathbb{I}$ the 4th order symmetric identity tensor. μ and λ are the viscosity constants. The first one represents the dynamic viscosity while the second one can be computed as $\lambda = -2\mu/3$ after taking the Stokes' assumption, which is valid in those cases in which the compressibility of the medium is negligible.

Inserting the previous definition of $\boldsymbol{\sigma}$ into the balance of linear momentum and mass conservation equations yields the viscous incompressible N–S equations

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{a} \cdot \boldsymbol{\nabla} \mathbf{u} - \boldsymbol{\nabla} \cdot (\mathbb{C} : \boldsymbol{\nabla}^s \mathbf{u}) + \boldsymbol{\nabla} p = \rho \mathbf{b} \tag{3.3a}$$

$$\frac{D\rho}{Dt} + \rho \boldsymbol{\nabla} \cdot \mathbf{u} = 0 \tag{3.3b}$$

being ρ the density and $\mathbf{b}$ the vector of volumetric (body) forces. $\partial(\bullet)/\partial t$ and $\boldsymbol{\nabla}$ denote the partial time derivative and the gradient operator. $\mathbf{a}$ is the convective velocity, which is defined from the mesh velocity $\mathbf{u}_m$ as $\mathbf{a} = \mathbf{u} - \mathbf{u}_m$.

At this point, it is important to provide some remarks about the convective term in Eq. 3.3a since it strongly affects the applicability and convergence of the method. The first one concerns the inclusion of $\mathbf{u}_m$, which equals 0 in a fixed mesh Eulerian framework. Although this is a valid simplification, it would impede the application of the FM–ALE algorithm in chapter 7. In consequence, it is convenient to keep such ALE definition of the convective velocity $\mathbf{a}$.

The second one concerns the non–linearity. As it is evinced in [32], it is convenient to do a Picard linearization of the convective term for the sake of the non-linear iteration robustness. Hence, the convective velocity is linearized as $\mathbf{a} = \mathbf{u}^{k-1} - \mathbf{u}_m$, where k superscript denote the current iteration. In consequence, the convective term turns into $\left(\mathbf{u}^{k-1} - \mathbf{u}_m\right) \cdot \boldsymbol{\nabla} \mathbf{u}^k$, which is the correct Picard linearization that ensures stability and proper convection of information [32].

With regard to the mass conservation equation, when dealing with purely incompressible fluids it is customary assumed that $\frac{D\rho}{Dt} = 0$, meaning that that the pressure field is defined up to a constant. In the standard body fitted case, this issue is immediately fixed by imposing a Neumann BC in any part of the domain. Unfortunately, this solution

might be no longer valid when working with a level set-based fixed mesh approach. This is due to the fact that, despite the level set is a robust and always defined operation, there are situations in which isolated closed domains of fluid (i.e. "bubbles") can appear. Taking into account that such undesired fluid cavities are likely related to poorly defined input files (e.g. stl) or to moving boundary problems involving complex geometries, it is impossible to a priori known their location. Trying to locate them in order to apply a Neumann BC is not a feasible solution since this is an extremely expensive, and possibly non-deterministic operation.

Alternatively, the unbounded pressure issue can be readily fixed if a slight compressibility is included in the formulation. This can be achieved by rewriting the density time derivative in Eq. 3.3b such that a state equation linking the density to the pressure field can be introduced. For almost incompressible fluids, it is possible to take the simplified speed of sound equation of state

$$c^2 = \frac{\partial p}{\partial \rho} \tag{3.4}$$

Hence, developing the material time derivative of the density in Eq. 3.3b as

$$\frac{\partial \rho}{\partial t} + \mathbf{a} \cdot \boldsymbol{\nabla} \rho + \rho \boldsymbol{\nabla} \cdot \mathbf{u} = 0$$

which is equivalent to

$$\frac{\partial \rho}{\partial p} \frac{\partial p}{\partial t} + \mathbf{a} \cdot \boldsymbol{\nabla} \rho + \rho \boldsymbol{\nabla} \cdot \mathbf{u} = 0$$

allows to get rid of the density partial time derivative by inserting Eq. 3.4 as

$$\frac{1}{c^2} \frac{\partial p}{\partial t} + \mathbf{a} \cdot \boldsymbol{\nabla} \rho + \rho \boldsymbol{\nabla} \cdot \mathbf{u} = 0$$

By further assuming that $\boldsymbol{\nabla} \rho \approx 0$, which physically means that the density space fluctuations are negligible, the term $\mathbf{a} \cdot \boldsymbol{\nabla} \rho$ can be neglected to give

$$\frac{1}{\rho c^2} \frac{\partial p}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{u} = 0 \tag{3.5}$$

Combining this last equation with the momentum conservation law in Eq. 3.3a yields

the final form of the quasi-incompressible N–S equations

$$\rho\frac{\partial \mathbf{u}}{\partial t} + \rho\mathbf{a}\cdot\boldsymbol{\nabla}\mathbf{u} - \boldsymbol{\nabla}\cdot(\mathbb{C}:\boldsymbol{\nabla}^s\mathbf{u}) + \boldsymbol{\nabla}p = \rho\mathbf{b} \tag{3.6a}$$

$$\frac{1}{\rho c^2}\frac{\partial p}{\partial t} + \boldsymbol{\nabla}\cdot\mathbf{u} = 0 \tag{3.6b}$$

It is important to mention that the fully-incompressible form $\nabla\cdot\mathbf{u} = 0$ is recovered as the sound speed $c \to \infty$. This makes possible to effectively deactivate the pseudo-compressible term in those cases in which isolated fluid cavities are not expected to appear. In chapter 5 a test example that proves the capability of this approach to deal with such undesired scenarios is presented.

3.2 Discrete form and stabilization

The FE discretization of mixed formulations, such as the Stokes (or N–S) equations, is commonly associated to saddle point problems [11]. These give rise to numerical artifacts and spurious oscillations, which most probably ruin the solution, when the discretizations of the mixed variables are not compatible between them. In this context, the Ladyzhenskaya–Babuška–Brezzi (LBB) condition, also known as inf–sup condition, provides a criterion to distinguish between stable and unstable discretizations. For more details on inf–sup stable and unstable FE discretizations, the reader is referred to [44, 87].

Aiming to keep the embedded elemental splitting as simple as possible, this thesis only considers simplex elements with same velocity and pressure interpolation order (i.e. linear triangles and tetrahedra [39]). Furthermore, simplex elements are the most straightforward approach to generate unstructured grids around complex geometries. However, they are inf–sup unstable and therefore require the use of a stabilization technique to avoid spurious oscillations. Note that using an inf–sup stable discretization would require the use of higher order approximations, which in the context of EBM would turn into an extremely cumbersome implementation of the splitting algorithms.

There is a wide variety of stabilization techniques that have been proved to be effective for similar N–S problems. Some of these are the Streamline Upwind/Petrov–Galerkin (SUPG) [13, 52], the Galerkin Least-Squares (GLS) [53], the Finite Increment Calculus (FIC) [25] or the Variational MultiScales (VMS) [49, 51], which is the adopted one.

The VMS method relies on the separation of the solution fields $\mathbf{u}$ and p in two scales as

$$\mathbf{u} = \mathbf{u}_h + \mathbf{u}_s \tag{3.7a}$$

$$p = p_h + p_s \tag{3.7b}$$

$\mathbf{u}_h$ and p_h being the velocity and pressure FE resolvable scales. Contrariwise, $\mathbf{u}_s$ and p_s are the subscales representing the velocity and pressure fluctuations that cannot be captured by the FE solution and thus need to be modeled.

The previous separation of scales can be inserted into Eq. 3.6 to yield

$$\begin{aligned} \rho\frac{\partial\left(\mathbf{u}_h+\mathbf{u}_s\right)}{\partial t} + \rho\mathbf{a}\cdot\boldsymbol{\nabla}\left(\mathbf{u}_s+\mathbf{u}_s\right) \\ -\boldsymbol{\nabla}\cdot\left(\mathbb{C}:\boldsymbol{\nabla}^s\left(\mathbf{u}_h+\mathbf{u}_s\right)\right)+\boldsymbol{\nabla}\left(p_h+p_s\right)=\rho\mathbf{b} \end{aligned} \tag{3.8a}$$

$$\frac{1}{\rho c^2}\frac{\partial\left(p_h+p_s\right)}{\partial t}+\boldsymbol{\nabla}\cdot\left(\mathbf{u}_h+\mathbf{u}_s\right)=0 \tag{3.8b}$$

At this point it is required to define a model for the subscales $\mathbf{u}_s$ and p_s, which are assumed to have a bubble shape that implies zero value of all their boundary integrals [49]. In general terms, it can be said that the subscales are built as a projection of the FE residuals onto the space of the small scales. In consequence, the nature of such projection defines the subscale model to end up with. Hence, assuming an identity projection operator leads to the Algebraic Sub-Grid Scales (ASGS) [19] while considering an orthogonal one, which implies the assumption that the subscales space is orthogonal to the FE one, leads to the Orthogonal SubScales(OSS) [20]. Owing to its remarkably good performance and simplicity, which turns into a more efficient implementation, the ASGS is the option selected in this work.

Hence, the ASGS velocity and pressure subscales can be calculated from the momentum and mass conservation discrete residuals $\mathbf{R^M}$ and R^C as

$$\mathbf{u}_s = \tau_1\mathbf{R^M}(\mathbf{u}_h, p_h) \tag{3.9a}$$

$$p_s = \tau_2 R^C(\mathbf{u}_h, p_h) \tag{3.9b}$$

being τ_1 and τ_2 the stabilization constants, which are taken from [19] and computed as

$$\tau_1 = \left(\frac{\rho \tau_{dyn}}{\Delta t} + \frac{c_2 \rho \|\mathbf{a}\|}{h} + \frac{c_1 \mu}{h^2} \right)^{-1} \tag{3.10a}$$

$$\tau_2 = \frac{h^2}{c_2 \tau_1} \tag{3.10b}$$

where $c_1 = 4.0$ and $c_2 = 2.0$, $\|\mathbf{a}\|$ is the convective velocity norm and h is the element size, which is computed as the average of the heights associated to each node of the element. τ_{dyn} is a parameter bounded between 0 and 1 that allows considering the time step value in the calculation of τ_1.

Note that the previous subscales definition (Eq. 3.9) introduces a dependency on the FE solution through the algebraic momentum and mass conservation residuals, which are defined as

$$\mathbf{R}^{\mathbf{M}}(\mathbf{u}_h, p_h) = \rho \mathbf{b} - \rho \frac{\partial \mathbf{u}_h}{\partial t} - \rho \mathbf{a} \cdot \boldsymbol{\nabla} \mathbf{u}_h + \boldsymbol{\nabla} \cdot (\mathbb{C} : \boldsymbol{\nabla}^s \mathbf{u}_h) - \boldsymbol{\nabla} p_h \tag{3.11a}$$

$$R^C(\mathbf{u}_h, p_h) = -\frac{1}{\rho c^2} \frac{\partial p_h}{\partial t} - \boldsymbol{\nabla} \cdot \mathbf{u}_h \tag{3.11b}$$

Besides this, Eq. 3.8 also shows the time dependent nature of the subscales. Considering such time dependency leads to a so called dynamic subscales formulation. Conversely, neglecting it by assuming that $\partial \mathbf{u}_s / \partial t \approx 0$ as well as $\partial p_s / \partial t \approx 0$ leads to a quasi-static subscales formulation. Even though the dynamic approach has superior characteristics [23], these come at the cost of increasing the complexity of the formulation. Moreover, they also involve a huge increase in the memory consumption as they introduce the need of storing the subscales historical data for each integration point. Taking these considerations into account, the quasi-static approach is the preferred option in this thesis.

3.3 Variational form and automatic differentiation

After having defined the stabilized governing equations, the variational (weak) form of the problem needs to be derived. To do that, the velocity and pressure test functions $\mathbf{w}$ and q are defined. For the sake of simplicity, it is also convenient to use the standard notation $(\bullet, \bullet)_\Omega$ for the inner product volume integral. Likewise, the inner product boundary integrals can be denoted as $\langle \bullet, \bullet \rangle_\Gamma$.

Hence, the Galerkin FE functional to be minimized is built as the inner product volume integral of the test functions times the FE residuals as

$$\Psi\left(\mathbf{w}, q, \mathbf{u}, p\right) := \left(\mathbf{w}, \mathbf{R}^{\mathbf{M}}(\mathbf{u}, p)\right)_{\Omega} + \left(q, R^{C}(\mathbf{u}, p)\right)_{\Omega} \tag{3.12}$$

Note that Eq. 3.12 involves the complete FE residuals, which include not only the FE solution but also the subscales. After introducing the scales separation and the quasi-static subscales assumption, the complete FE residuals $\mathbf{R}^{\mathbf{M}}(\mathbf{u}, \mathbf{p})$ and $R^{C}(\mathbf{u}, p)$ read as

$$\begin{aligned} \mathbf{R}^{\mathbf{M}}(\mathbf{u}, p) = \rho\mathbf{b} - \rho\frac{\partial \mathbf{u}_h}{\partial t} - \rho\mathbf{a}\cdot\boldsymbol{\nabla}\left(\mathbf{u}_h + \mathbf{u}_s\right) \\ + \boldsymbol{\nabla}\cdot\left(\mathbb{C} : \boldsymbol{\nabla}^{s}\left(\mathbf{u}_h + \mathbf{u}_s\right)\right) - \boldsymbol{\nabla}\left(p_h + p_s\right) \end{aligned} \tag{3.13a}$$

$$R^{C}(\mathbf{u}, p) = -\frac{1}{\rho c^2}\frac{\partial p_h}{\partial t} - \boldsymbol{\nabla}\cdot\left(\mathbf{u}_h + \mathbf{u}_s\right) \tag{3.13b}$$

Inserting these into the Galerkin functional above yields the stabilized quasi–incompressible Navier–Stokes Galerkin functional

$$\begin{aligned} (\mathbf{w}, \rho\mathbf{b})_{\Omega} - \left(\mathbf{w}, \rho\frac{\partial \mathbf{u}_h}{\partial t}\right)_{\Omega} - (\mathbf{w}, \rho\mathbf{a}\cdot\boldsymbol{\nabla}(\mathbf{u}_h + \mathbf{u}_s))_{\Omega} \\ + (\mathbf{w}, \boldsymbol{\nabla}\cdot(\mathbb{C} : \boldsymbol{\nabla}^{s}(\mathbf{u}_h + \mathbf{u}_s)))_{\Omega} - (\mathbf{w}, \boldsymbol{\nabla}(p_h + p_s))_{\Omega} \\ - \left(q, \frac{1}{\rho c^2}\frac{\partial p_h}{\partial t}\right)_{\Omega} - (q, \boldsymbol{\nabla}\cdot(\mathbf{u}_h + \mathbf{u}_s))_{\Omega} = 0 \end{aligned} \tag{3.14}$$

Taking into account that this formulation is to be discretized using simplicial elements, integration by parts is demanded for getting rid of as higher order derivatives as possible. After doing so and dropping all the remaining higher order terms, the final Galerkin functional to be implemented reads as

$$\begin{aligned} (\mathbf{w}, \rho\mathbf{b})_{\Omega} - \left(\mathbf{w}, \rho\frac{\partial \mathbf{u}_h}{\partial t}\right)_{\Omega} - (\mathbf{w}, \rho\mathbf{a}\cdot\boldsymbol{\nabla}\mathbf{u}_h)_{\Omega} - (\boldsymbol{\nabla}^{s}\mathbf{w}, \mathbb{C} : \boldsymbol{\nabla}^{s}\mathbf{u}_h)_{\Omega} \\ + (\boldsymbol{\nabla}\cdot\mathbf{w}, p_h)_{\Omega} - \left(q, \frac{1}{\rho c^2}\frac{\partial p_h}{\partial t}\right)_{\Omega} - (q, \boldsymbol{\nabla}\cdot\mathbf{u}_h)_{\Omega} + \langle\mathbf{w}, \mathbf{t}\rangle_{\Gamma} \\ + (\rho\mathbf{a}\boldsymbol{\nabla}\mathbf{w}, \mathbf{u}_s)_{\Omega} + (\rho(\mathbf{w}\boldsymbol{\nabla}\cdot\mathbf{a}), \mathbf{u}_s)_{\Omega} + (\boldsymbol{\nabla}\cdot\mathbf{w}, p_s)_{\Omega} + (\boldsymbol{\nabla}q, \mathbf{u}_s)_{\Omega} = 0 \end{aligned} \tag{3.15}$$

being $\mathbf{t}$ the Cauchy traction vector, computed as $\mathbf{t} = (\mathbb{C} : \boldsymbol{\nabla}^{s}\mathbf{u}_h - p_h\mathbf{I})\cdot\mathbf{n}$, and $\mathbf{n}$ the outwards unit normal vector. Note that this equation already takes into account the

subscales null boundary value assumption.

The algebraic form of the discrete functional in Eq. 3.15 can be automatically obtained by using a Computer Algebra System (CAS). Among all the available alternatives, in this work the Python library Sympy [74] is used. Aside of being a much more efficient alternative to the traditional implementation in terms of the human effort, following a symbolic implementation ensures that the differentiation of the functional is always correct since this is automatically performed by the CAS library. This limits the possible error sources to the definition of the Galerkin functional, and thus minimizes the potential implementation mistakes.

Therefore, the elemental Left Hand Side matrix (**LHS**) and Right Hand Side vector (**RHS**) can be automatically obtained by expressing the symbolic functional in terms of the nodal test functions ($\mathbf{w}_I$ and q_I) and the nodal unknowns ($\mathbf{u}_I$ and p_I). Then, assuming a symbolic description of the shape functions and of their derivatives, allows obtaining the elemental **RHS** by automatic differentiation as

$$\mathbf{RHS}_I = \frac{\partial \Psi \left(\mathbf{w}, q, \mathbf{u}, p \right)}{\partial \left(\mathbf{w}_I, q_I \right)} \tag{3.16}$$

while the elemental **LHS** can be similarly derived as

$$\mathbf{LHS}_{IJ} = -\frac{\partial \mathbf{RHS}_I}{\partial \left(\mathbf{u}_{h_J}, p_{h_J} \right)} \tag{3.17}$$

The symbolic implementation of the quasi–incompressible Navier–Stokes stabilized formulation presented in this chapter is available in the Kratos Multiphysics repository.

Last but not least, it is due mentioning that this section assumes previous knowledge on the FE method. In case more details about the basis of the method are required, the reader is referred to [39, 118]. Concerning the specific application of the FE method to CFD problems, the reader is referred to [18, 36, 87].

3.4 Validation

This section validates the FE implementation of the quasi–incompressible Navier–Stokes formulation previously described. To that purpose the Method of Manufactured Solutions (MMS) is used [88, 96]. The problem geometry consists in a 1x1m square while the material properties ρ and μ are equal to 10^3 and 10^{-4}. The speed of sound velocity

c is set to 10^4 in order to introduce the effect of the pseudo–compressible term.

The problem is solved for two different source terms that correspond to two different transient solutions. First the source term

$$\begin{aligned} b_x = &- \rho\pi \sin(\pi x)\cos(\pi y)\sin(\pi t) \\ &+ 2\mu\pi^2 \sin(\pi x)\cos(\pi y)\cos(\pi t) + \rho\pi \cos^2(\pi t)\sin(\pi x)\cos(\pi x) \\ b_y = &\rho\pi \cos(\pi x)\sin(\pi y)\sin(\pi t) \\ &- 2\mu\pi^2 \cos(\pi x)\sin(piy)\cos(\pi t) + \rho\pi \cos^2(\pi t)\sin(\pi y)\cos(\pi y) \end{aligned}$$

that corresponds to the sinusoidal transient field

$$\begin{aligned} u_x &= \sin(\pi x)\cos(\pi y)\cos(\pi t) \\ u_y &= -\cos(\pi x)\sin(\pi y)\cos(\pi t) \\ p &= 0 \end{aligned}$$

is set.

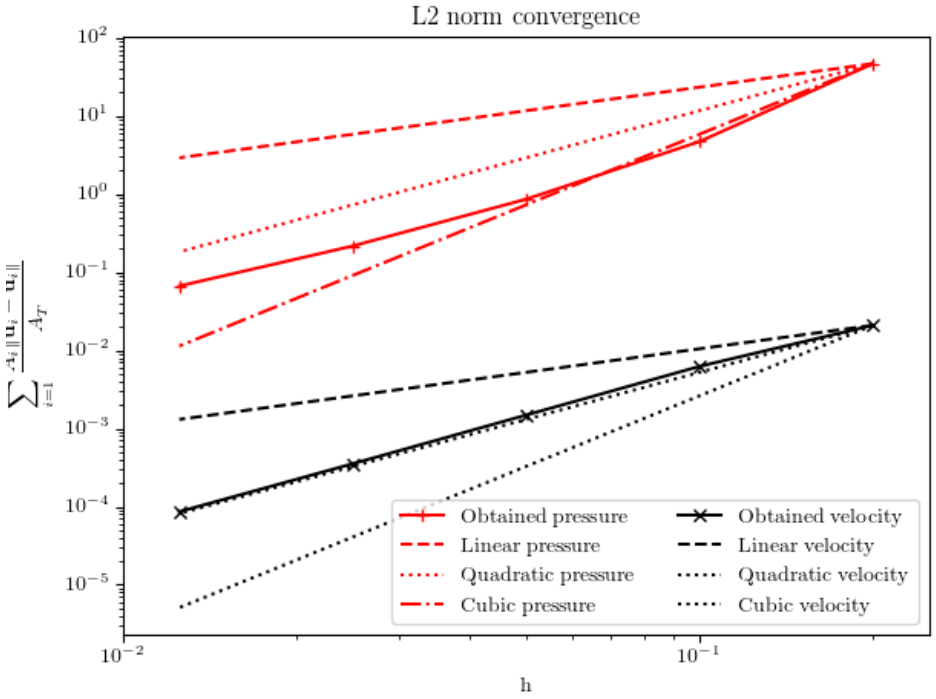
Secondly, the experiment is repeated for the source term

$$\begin{aligned} b_x &= \rho\pi x^2 y\cos\pi t - 2\mu y\sin(\pi t) + \rho x^3 y^2 \sin^2(\pi t) \\ b_y &= -\rho\pi x y^2 \cos(\pi t) + 2\mu x \sin(\pi t) + \rho x^2 y^3 \sin^2(\pi t) \\ p &= 0 \end{aligned}$$

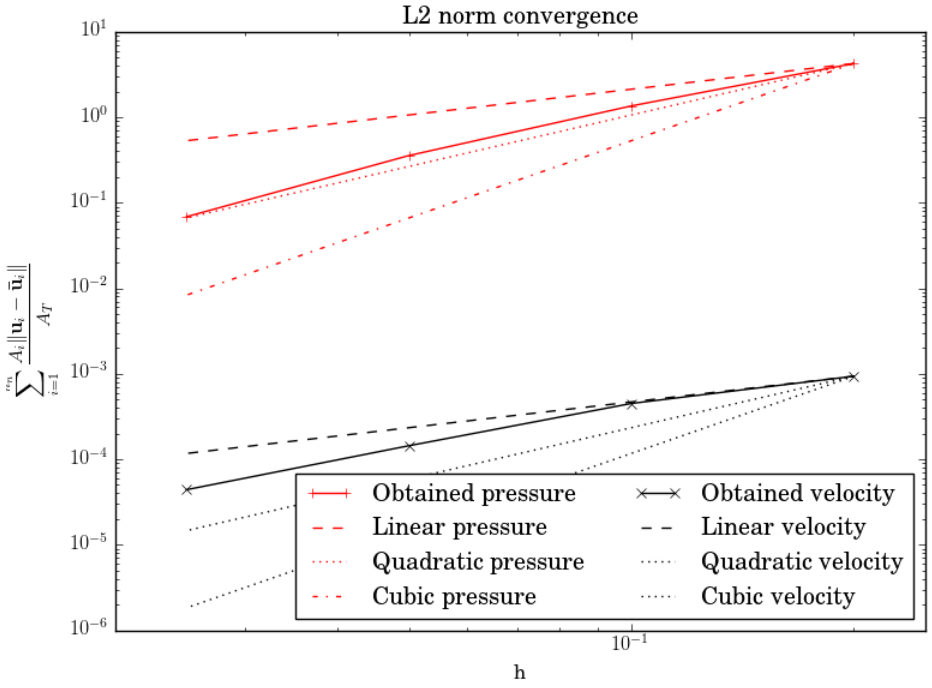
which corresponds to the non-linear transient field

$$\begin{aligned} u_x &= x^2 y \sin(\pi t) \\ u_y &= -xy^2 \sin(\pi t) \\ p &= 0 \end{aligned}$$

Fig. 3.1 collects the weighted average error for both the velocity and pressure fields after 1s of simulation. In both cases it can be observed that the velocity and pressure convergence rates match the expected h^2 and h ones.



(a) Sinusoidal transient field.



(b) Non-linear transient field.

Figure 3.1: Manufactured solution experiments results.

Chapter 4

A modified Finite Element formulation for the imposition of the slip boundary condition over embedded volumeless geometries

4.1 Article data

Title: A modified Finite Element formulation for the imposition of the slip boundary condition over embedded volumeless geometries
Authors: R. Zorrilla, A. Larese and R. Rossi
Journal: Computer Methods in Applied Mechanics and Engineering 353 (2019) 123–157
Received: 2 May 2018 / Accepted: 10 May 2019 / Available online: 15 May 2019
DOI: 10.1016/j.cma.2019.05.007

4.2 Scientific contribution

This chapter collects the first paper of the compendium, which presents a novel embedded mesh method for the resolution of Navier–Stokes problems. Even though the proposed formulation is capable of dealing with any type of body, it is intended to be used in combination with discontinuous level set functions to allow the resolution of

CFD problems involving embedded thin-walled bodies.

The proposal is based on the substitution of the standard FE space by an alternative one in those elements that are intersected by the level set. Such alternative space, which was firstly proposed in [2], makes possible the representation of both the velocity and pressure discontinuities arising from the immersion of any body. In addition to this, a no penetration condition is weakly imposed by doing a convenient integration by parts of the mass conservation equation and the addition of an extra term that penalizes the normal projection of the embedded wall velocity. Taking into account that no constraint is enforced in the tangential direction, the imposition is equivalent to a slip BC.

The paper also discusses alternative approaches to impose the slip condition, namely a MultiFreedom Constraint (MFC) approach for body fitted discretizations and a Nitsche-based imposition of the Navier-slip BC for volumetric embedded bodies. These alternative approaches are also applied to validate the discontinuous embedded proposal. The accuracy of the method is proved as well as its ability to keep the $h^{3/2}$ convergence order of the modified FE space. Furthermore, the chapter also showcases a potential real application that the proposed technique could have in the context of membrane structures CFD by solving the flow around two boat sails.

The results obtained in this chapter are a very first step towards the final VWT objective. More specifically, this chapter settles the basis of the discontinuous embedded Navier–Stokes formulations that are to be exploited in the subsequent chapters for the resolution of more complex CFD and FSI problems.

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 353 (2019) 123–157
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

A modified Finite Element formulation for the imposition of the slip boundary condition over embedded volumeless geometries

R. Zorrilla[a,b,*], A. Larese[b,c], R. Rossi[a,b]

[a] *Universitat Politècnica de Catalunya (UPC), Departament d'Enginyeria Civil i Ambiental, Spain*
[b] *International Center for Numerical Methods in Engineering (CIMNE), Spain*
[c] *Università degli Studi di Padova, Department of Mathematics, Italy*

Received 2 May 2018; received in revised form 6 February 2019; accepted 10 May 2019
Available online 15 May 2019

Abstract

This work describes a novel formulation for the simulation of Navier–Stokes problems including embedded objects. The new proposal is based on the use of a modified finite element space, which replaces the standard one within the elements intersected by the immersed geometry. The modified space is able to exactly reproduce the jumps happening at the embedded boundary while preserving the conformity across the faces intersected by the embedded object. The paper focuses particularly on the imposition of a slip boundary condition on the surface of the embedded geometry, proposing a new technique for the application of such constraint. The new proposal is carefully benchmarked using the results of a body fitted technique and of an alternative embedded approach. Potential applications of interest are also presented.

MSC: 00-01; 99-00

Keywords: Embedded boundary methods; Immersed boundary methods; Navier–Stokes; Volumeless bodies; Slip boundary condition; Discontinuous shape functions

1. Introduction

The Computational Fluid Dynamics (CFD) simulation of the fluid flow around objects is typically achieved by the construction of a volume discretization that matches the geometry of the body of interest as closely as possible. This gives rise to the so called body-fitted discretization. While seeming appealing, such approaches have obvious limitations, that become apparent for the simulation of moving bodies. The need of overcoming such limitations has led to the development of a variety of alternative methods, such as the Immersed Boundary Method (IBM) [1,2] or the Embedded Boundary Method (EBM). Such techniques make it possible to automatically include arbitrary, possibly moving, bodies into the fluid domain, and to account for their interaction with the fluid flow. The crucial difference with respect to the body-fitted alternatives is that the analysed body and the fluid are discretized separately. While performing the simulation, the body is overlapped onto the fluid domain at the position of interest. An

* Corresponding author at: Universitat Politècnica de Catalunya (UPC), Departament d'Enginyeria Civil i Ambiental, Spain.
E-mail addresses: rzorrilla@cimne.upc.edu (R. Zorrilla), antonia.larese@unipd.it (A. Larese), rrossi@cimne.upc.edu (R. Rossi).

https://doi.org/10.1016/j.cma.2019.05.007
0045-7825/

automatic intersection is performed, typically by employing a level set technique [3], to implicitly represent the object of interest in the fluid domain.

While the use of an unfitted approach inevitably introduces complexities into the formulation, the resulting methods can overcome some of the difficulties associated with the conforming counterparts. For example, while it is possible to employ Arbitrary Lagrangian–Eulerian (ALE) techniques [4,5], large displacements and rotations of the body typically yield to extremely distorted (or even inverted) elements, practically limiting the use of ALE solutions to relatively small boundary movements. Such limitation does not exist for fixed mesh methods, which can handle arbitrarily large displacements and rotations, or even changes in the topology (simply impossible for ALE techniques), by updating the level set representation in accordance to the movement of the body.

Furthermore, since the geometric distance computation is a robust operation, the level set based approaches open new possibilities to directly include the exact CAD geometry (involving trimmed NURBS) into the simulation pipeline or to simulate "dirty" geometries. This is because the distance computation from a NURBS surface is an operation as robust as computing the distance from a lower order discretization. Furthermore, the use of a level set approach intrinsically filters out the geometric details that cannot be represented by the volume mesh, improving the robustness and speed of the model preparation phase. Taking into account that the model preparation of realistic engineering problems is known to amount to more than the 50% of the total analysis time [6], this is clearly an important advantage, particularly considering that the details can always be recovered upon refinement. Moreover, the mesh generation step is typically more robust since the mesh is not required to comply with the details of the body geometry. Lastly, the model preparation can also benefit from the use of fast octree mesh generators [7] to provide initial meshes which can be then improved by the use of mesh adaptation techniques [8,9].

A rather large literature study the use of fixed mesh approaches to solve complex incompressible Navier–Stokes (N–S) problems. For instance, the IBM is successfully applied to model a helicon ribbon mixer problem in [10]. This method is also employed in [11] for solving complex real engineering geometries without need to modify the input files before the mesh generation.

In the literature, the distinction between "Immersed" and "Embedded" techniques tends to be blurry and different authors may provide a different definition. For the sake of clarity, we will use the term "Immersed" when the Solid–Fluid coupling is performed by imposing a constraint over the entire overlapping zone between the domains and the term "Embedded" when the coupling is enforced at the frontier between solid and fluid.

According to such definition, even though the IBM out stands for its robustness and implementation simplicity, it may lack precision in some applications, since the boundary condition is not directly enforced over the interface cut. Instead of that, the immersed object velocity is directly imposed on those fluid nodes lying on the interior of the object, making it impossible to recover the original interface.

On the contrary, EBM relies on applying the boundary conditions over the interface cut. This is commonly done in a weak sense by using techniques such as the penalty method or the Nitsche method [12]. In [13], the authors apply a modified Nitsche method to impose the no-slip boundary condition to the N–S equations. In [14], a stabilized Nitsche method is used for the imposition of the no-slip boundary condition to the Stokes equations. A similar technique is applied in [15] for the Oseen equations. Regarding the slip boundary condition, Lagrange multipliers technique and the Nitsche method are used to apply the slip boundary condition to the Stokes equations in [16]. This is extended to apply a general Navier-slip boundary condition to the N–S equations in [17]. In [18], the authors present an innovative approach where a spline-based surface is directly immersed in the fluid domain to impose a Nitsche no-slip boundary condition.

A common feature of the above highlighted formulations is the need for a well-defined internal volume. This turns into a limitation when shell or membrane bodies, such as lightweight structures, biological tissues or boat sails, need to be analysed. Multiple works address this limitation by adding a volume force that modifies the flow pattern in accordance to the embedded volumeless geometry [1,19,20]. Even though this is probably the most straightforward approach, it is difficult to represent the flow discontinuities as well as to precisely compute the required volume force value. An alternative based on adding the immersed bodies dynamics to the flow dynamics to model embedded moving fibres is presented in [21]. More complex approaches also exist, as the one presented in [22], which uses a fractional step ghost cell method for representing complex moving geometries. While the method presented in [22] works for any type of geometry (with/without internal volume), it requires the neighbour elements for the imposition of the boundary conditions. As it is widely known, the implementation of this kind of operations becomes even more cumbersome in a distributed memory environment (MPI).

The aim of the current work is to present a generalized unfitted technique which is able to overcome all the previously commented limitations. The proposal takes the idea from the use of the discontinuous space investigated in [23]. Our proposal makes possible to represent discontinuities in the fluid flow, so it is suitable not only for the simulation of bodies with a well defined volume but also for volumeless ones. Besides, the formulation is purely elemental and therefore easily extensible to the use of a distributed memory environment.

A classical difficulty of embedded approaches is the lack of resolution in the vicinity of the body, which impedes the correct simulation of viscous effects close to the body surface. Current work proposes a technique for the imposition of the slip boundary condition on embedded boundaries. While, from a theoretical point of view the use of a slip BC is only well defined for inviscid fluids, the slip approximation is good for high Reynolds (Re) flows [24]. In that work, the results of the viscous-slip approach are compared with experimental data, confirming that this combination is a valid alternative when a high boundary layer resolution is not required. Apart from the high-Re scenarios, the slip approximation is also relevant for some highly viscous flows applications. For example it could be an alternative to consider the effect of lubrication on the domain boundaries (lubrication is indeed applied with the exact purpose of avoiding a stick condition between the viscous fluid and the walls).

All the formulations discussed in this work have been implemented within the Kratos Multiphysics open source framework [25,26].

2. Methodology

Firstly, the governing equations are presented and the finite element formulation is derived using an automatic differentiation technique. After that, the approaches to impose the slip boundary condition are described. Special emphasis is put on the one proposed by the authors, which uses an alternative set of shape functions (henceforth named "Ausas" shape functions after the name of the original author).

2.1. The Navier–Stokes equations

In this work we focus on the incompressible Navier–Stokes equations for Newtonian fluids. The Cauchy stress tensor $\boldsymbol{\sigma}$ is defined as $\boldsymbol{\sigma} = -p\mathbf{I}+\mathbb{C} : \boldsymbol{\nabla}^s\mathbf{u}$, where $\mathbf{u}$ is the velocity, p the pressure, $\boldsymbol{\nabla}^s$ the symmetric gradient operator and $\mathbb{C}$ the constitutive tensor describing the viscous behaviour. By substituting $\boldsymbol{\sigma}$ into the balance of linear momentum and mass conservation equations yields the well-known viscous incompressible Navier–Stokes equations

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \rho\mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u} - \boldsymbol{\nabla}\cdot\left(\mathbb{C} : \boldsymbol{\nabla}^s\mathbf{u}\right) + \boldsymbol{\nabla}p = \rho\mathbf{b} \tag{1a}$$

$$\frac{D\rho}{Dt} + \rho\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \tag{1b}$$

where ρ the density and $\mathbf{b}$ the body force. The operator $\partial(\bullet)/\partial t$ is the partial time derivative and $\boldsymbol{\nabla}$ is the gradient operator.

For purely incompressible fluids, it is customary to assume $\frac{D\rho}{Dt} = 0$. This implies that the pressure is defined up to a constant, which is typically fixed once a Neumann boundary condition is imposed. Unfortunately, this feature becomes problematic when dealing with poorly defined input geometries or moving boundaries since isolated closed domains of fluid (with no Neumann Boundary) may appear.

This issue is readily fixed if a slight compressibility is included in the formulation. In particular if we assume the simplified equation of state $p = \rho c^2 \implies c^2 = \partial p/\partial\rho$, which is valid for almost incompressible fluids, we can rewrite the density time derivative in terms of the pressure as

$$\frac{D\rho}{Dt} + \rho\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \longrightarrow \frac{\partial\rho}{\partial p}\frac{Dp}{Dt} + \rho\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \longrightarrow \frac{1}{c^2}\frac{Dp}{Dt} + \rho\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \tag{2}$$

If we further assume that $\boldsymbol{\nabla}\rho \approx \mathbf{0}$, physically expressing that density fluctuations are negligible, the term $\mathbf{u}\cdot\boldsymbol{\nabla}\rho$ can be neglected to give

$$\frac{1}{\rho c^2}\frac{\partial p}{\partial t} + \boldsymbol{\nabla}\cdot\mathbf{u} = 0 \tag{3}$$

The final form of the governing equations is thus

$$\rho\frac{\partial\mathbf{u}}{\partial t} + \rho\mathbf{u}\cdot\boldsymbol{\nabla}\mathbf{u} - \boldsymbol{\nabla}\cdot\left(\mathbb{C} : \boldsymbol{\nabla}^s\mathbf{u}\right) + \boldsymbol{\nabla}p = \rho\mathbf{b} \tag{4a}$$

$$\frac{1}{\rho c^2}\frac{\partial p}{\partial t} + \nabla \cdot \mathbf{u} = 0 \tag{4b}$$

Note that the single-fluid fully-incompressible form $\nabla \cdot \mathbf{u} = 0$ is recovered if the sound speed $c \to \infty$. Specifically, the compressibility is not needed for the examples presented in the current work. Hence through the paper the speed of sound is taken as 10^{12} m/s, so that the compressibility is effectively negligible in all of the discussed cases. However, the option to employ a higher value is left to the user.

2.2. Discrete form and stabilization

In the current work, only simplicial elements are considered. It is widely known that these elements do not satisfy the *inf–sup condition* and the use of a stabilization method is therefore required. There exist several alternatives that are proved to be effective for similar problems. Among those techniques, we can mention the Finite Increment Calculus (FIC) [27], the Streamline upwind/Petrov–Galerkin (SUPG) [28] and the Variational Multiscales Method (VMS) [29,30]. We adopt the VMS method, which relies on the separation of the solution fields $(\mathbf{u}, p)$ in two scales as

$$\mathbf{u} = \mathbf{u_h} + \mathbf{u_s} \tag{5a}$$

$$p = p_h + p_s \tag{5b}$$

The first (Eq. (5)) is the FE resolvable scale $(\mathbf{u_h}, p_h)$. The second one, referred to as the subscale $(\mathbf{u_s}, p_s)$, represents the fluctuations that cannot be captured by the FE solution. After inserting Eq. (5) into the N–S equations (Eq. (4)), the governing equations of the problem read

$$\begin{aligned} &\rho \frac{\partial (\mathbf{u_h} + \mathbf{u_s})}{\partial t} + \rho (\mathbf{u_h} + \mathbf{u_s}) \cdot \nabla (\mathbf{u_s} + \mathbf{u_s}) \\ &-\nabla \cdot (\mathbb{C} : \nabla^s (\mathbf{u_h} + \mathbf{u_s})) + \nabla (p_h + p_s) = \rho \mathbf{b} \end{aligned} \tag{6a}$$

$$\frac{1}{\rho c^2}\frac{\partial (p_h + p_s)}{\partial t} + \nabla \cdot (\mathbf{u_h} + \mathbf{u_s}) = 0 \tag{6b}$$

Different models for the subscale are presented in the literature. In general terms, we can say that the subscales are normally expressed as a projection of the FE residuals onto the space of the small scales. Depending on the nature of this projection, the Algebraic Sub-Grid Scales (ASGS) [31] or the Orthogonal Sub-Grid Scales (OSS) [32] are obtained. In this work, the ASGS technique is selected. This method recovers the velocity $(\mathbf{u_s})$ and pressure (p_s) subscales by using an algebraic approach based on the FE velocity $(\mathbf{u_h})$ and pressure (p_h) solutions as

$$\mathbf{u_s} = \tau_1 \mathbf{R^M}(\mathbf{u_h}, p_h) \tag{7a}$$

$$p_s = \tau_2 R^C(\mathbf{u_h}, p_h) \tag{7b}$$

where $\mathbf{R^M}$ and R^C are the residuals of the momentum and mass conservation equations. Note that the subscale dependency on the FE solution is introduced by such residuals of the governing equations, which are defined as

$$\mathbf{R^M}(\mathbf{u_h}, p_h) = \rho \mathbf{b} - \rho \frac{\partial \mathbf{u_h}}{\partial t} - \rho \mathbf{u_h} \cdot \nabla \mathbf{u_h} + \nabla \cdot \left(\mathbb{C} : \nabla^s \mathbf{u_h}\right) - \nabla p_h \tag{8a}$$

$$R^C(\mathbf{u_h}, p_h) = -\frac{1}{\rho c^2}\frac{\partial p_h}{\partial t} - \nabla \cdot \mathbf{u_h} \tag{8b}$$

Eq. (6) shows that the subscales are time dependent, leading to a so called *dynamic subscales formulation*. If this time dependency is neglected by assuming that $\partial \mathbf{u_s}/\partial t \approx \mathbf{0}$ and $\partial p_s/\partial t \approx 0$, a *quasi-static subscales formulation* is obtained. Even though the dynamic approach has somewhat superior characteristics [33], these come at the cost of additional complexity in the formulation as well as of increased memory consumption and computational effort, which lead us to prefer the use of the quasi-static approach.

Both $\mathbf{u_s}$ and p_s are assumed to be such that their boundary integrals are 0. The stabilization constants τ_1 (Eq. (9a)) and τ_2 (Eq. (9b)) are taken from [31]. These are defined as

$$\tau_1 = \left(\frac{\rho \tau_{dyn}}{\Delta t} + \frac{c_2 \rho \|\mathbf{u_h}\|}{h} + \frac{c_1 \mu}{h^2}\right)^{-1} \tag{9a}$$

$$\tau_2 = \frac{h^2}{c_1 \tau_1} \tag{9b}$$

where τ_{dyn} is a parameter bounded between 0 and 1. $c_1 = 4.0$ and $c_2 = 2.0$ are the stabilization constants, $\|\mathbf{u}\|$ is the convective velocity norm, μ is the dynamic viscosity and h is the element size. In this work h is always computed as the average of the heights associated to each node of the element.

2.3. Automatic differentiation

Once the governing Partial Differential Equations (PDE) are defined, we proceed to discretize the problem, which we cast in residual form.

By using the standard notation $(\bullet, \bullet)_\Omega$ to denote the inner product volume integral, the discrete FE functional can be defined as

$$\Psi(\mathbf{w}, q, \mathbf{u}, p) := \left(\mathbf{w}, \mathbf{R}^{\mathbf{M}}(\mathbf{u}, p)\right)_\Omega + \left(q, R^C(\mathbf{u}, p)\right)_\Omega \tag{10}$$

where $\mathbf{w}$ and q are the velocity and pressure test functions.

Substituting the momentum and mass conservation residuals, together with the solution decomposition in Eq. (5), into the Galerkin functional (Eq. (10)) yields the Navier–Stokes functional to be solved

$$\begin{aligned}(\mathbf{w}, \rho\mathbf{b})_\Omega - \left(\mathbf{w}, \rho\frac{\partial \mathbf{u_h}}{\partial t}\right)_\Omega - (\mathbf{w}, \rho\mathbf{u_h}\cdot\nabla(\mathbf{u_h}+\mathbf{u_s}))_\Omega \\ + \left(\mathbf{w}, \nabla\cdot(\mathbb{C}:\nabla^s(\mathbf{u_h}+\mathbf{u_s}))\right)_\Omega - (\mathbf{w}, \nabla(p_h+p_s))_\Omega \\ - \left(q, \frac{1}{\rho c^2}\frac{\partial p_h}{\partial t}\right)_\Omega - (q, \nabla\cdot(\mathbf{u_h}+\mathbf{u_s}))_\Omega = 0\end{aligned} \tag{11}$$

where the quasi-static subscales model is already taken into consideration. After integrating by parts, the previous Navier–Stokes functional reads

$$\begin{aligned}&(\mathbf{w}, \rho\mathbf{b})_\Omega - \left(\mathbf{w}, \rho\frac{\partial \mathbf{u_h}}{\partial t}\right)_\Omega - (\mathbf{w}, \rho\mathbf{u_h}\cdot\nabla\mathbf{u_h})_\Omega - \left(\nabla^s\mathbf{w}, \mathbb{C}:\nabla^s\mathbf{u_h}\right)_\Omega \\ &+ (\nabla\cdot\mathbf{w}, p_h)_\Omega - \left(q, \frac{1}{\rho c^2}\frac{\partial p_h}{\partial t}\right)_\Omega - (q, \nabla\cdot\mathbf{u_h})_\Omega + \langle\mathbf{w}, \mathbf{t}\rangle_\Gamma \\ &+ (\rho\mathbf{u_h}\nabla\mathbf{w}, \mathbf{u_s})_\Omega + (\rho(\nabla\cdot\mathbf{u_h})\mathbf{w}, \mathbf{u_s})_\Omega + (\nabla\cdot\mathbf{w}, p_s)_\Omega + (\nabla q, \mathbf{u_s})_\Omega = 0\end{aligned} \tag{12}$$

where the dot product boundary integral is denoted as $\langle\bullet, \bullet\rangle_\Gamma$, with $\mathbf{t}$ being the Cauchy traction vector, computed as $\mathbf{t} = (\mathbb{C}:\nabla^s\mathbf{u_h} - p_h\mathbf{I})\cdot\mathbf{n}$, and $\mathbf{n}$ the boundary outwards unit vector. Note that the null boundary value of the subscales assumption is considered here.

The discrete functional (Eq. (12)) is then symbolically implemented in Python using the Computer Algebra System (CAS) library *Sympy* [34]. To automatically obtain the elemental Left Hand Side (**LHS**) and Right Hand Side (**RHS**), the symbolic functional needs to be expressed in terms of the nodal test functions ($\mathbf{w}_I$ and q_I) and of the nodal unknowns ($\mathbf{u}_I$ and p_I). Then, by assuming a symbolic description of the shape functions and of their derivatives, the elemental **RHS** is obtained by symbolic differentiation as

$$\mathbf{RHS}_I = \frac{\partial\Psi(\mathbf{w}, q, \mathbf{u}, p)}{\partial(\mathbf{w}_I, q_I)} \tag{13}$$

while the elemental **LHS** is similarly obtained as

$$\mathbf{LHS}_{IJ} = -\frac{\partial\mathbf{RHS}_I}{\partial\left(\mathbf{u}_{\mathbf{h},J}, p_{h,J}\right)} \tag{14}$$

2.4. Embedded formulations and immersed bodies representation

The main distinguishing feature of embedded (or immersed) approaches with respect to body fitted alternatives is that the domain is meshed without taking into account the geometry of the analysed bodies. Instead, the object representation is achieved by the use of a level set function [3] defined as the signed distance to the object skin, which guarantees that the body shape can be recovered as the zero iso-surface of the level set field.

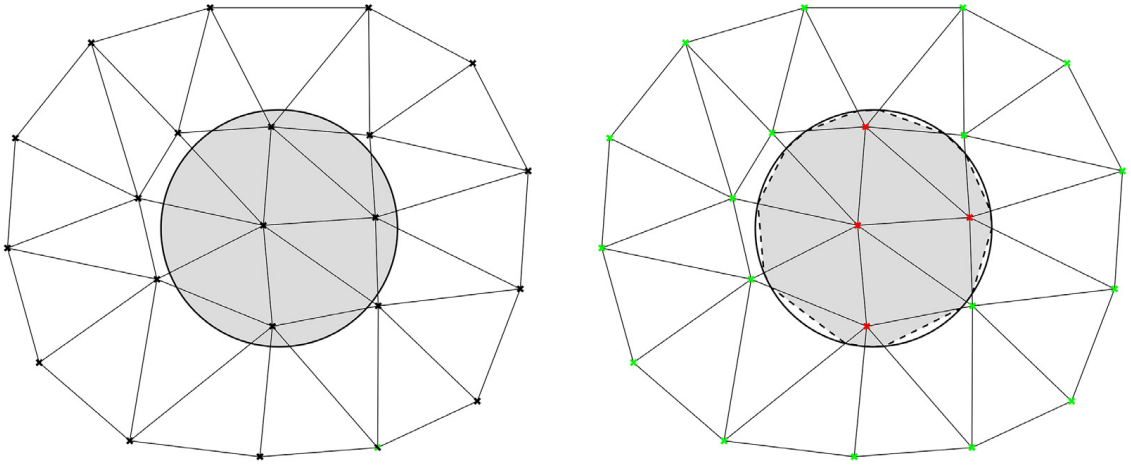


Fig. 1. Continuous distance function. Body with a well-defined internal volume (left) and continuous distance representation (right). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Considering the nature of the analysed bodies, the distance functions can be roughly divided in two types. The first type is used to describe those objects that have a well-defined internal volume (e.g. aerofoils). These bodies can be represented using a continuous signed distance function. As depicted in Fig. 1, such function takes a positive value in the fluid domain nodes (light green) or a negative one in the structure domain nodes (red). The zero isosurface (dashed line) is therefore the immersed object skin representation.

The second type is used for representing bodies without internal volume (e.g. boat sails). This kind of geometries cannot be represented by a continuous distance function since no intersections could be found. This limitation can be overcome using a discontinuous distance field, which is computed (and stored) element by element, meaning that the same node can have different distance values depending on the element considered. Such feature is what makes the distance function discontinuous and allows tracking such volume-less geometries. Fig. 2 shows a qualitative example of this elemental distance function. Uncut elements, which have a positive constant distance value, are coloured in light green. On the other hand, green and red are used to colour the positive and negative distance regions of cut elements.

Moreover, it is worth mentioning the treatment of complex intersection patterns and their implications in the calculation of such discontinuous distance function. Fig. 2 qualitatively describes an example of these cases. By inspecting one of the elements intersected by two bodies at the same time, it is observable that the multiple intersection pattern is approximated as plane which is not coincident with the neighbours' intersections, generating a discontinuity in the distance field. This kind of complexities together with the discussion and implementation of the techniques to deal with them are extensively described in [35].

Finally, it is interesting to comment that all the geometries that could be represented by a continuous distance field could also be represented by a discontinuous one as well. From a mathematical point of view, this is explained by the fact that the space of continuous distance functions is contained in the discontinuous one. Indeed, any continuous distance algorithm firstly computes the elemental body–skin intersections, which altogether conform the discontinuous distance field, for then obtaining the continuous one by avoiding the jumps between neighbour elements.

2.5. Body fitted slip boundary condition

Even though this work mainly focuses on embedded fluid formulations, body fitted discretizations are used as reference solution during the implementation and validation of the presented formulation. This subsection describes the technique used for the imposition of the slip boundary condition when dealing with body fitted discretizations.

In this case, the body fitted slip condition is imposed in a "MultiFreedom Constraint" (MFC) fashion [36]. This approach is based on the rotation of the assembled stiffness matrix of the problem. This means that for each slip node, a local coordinate system is defined such that one component (two in 3D) is tangent to the slip boundary

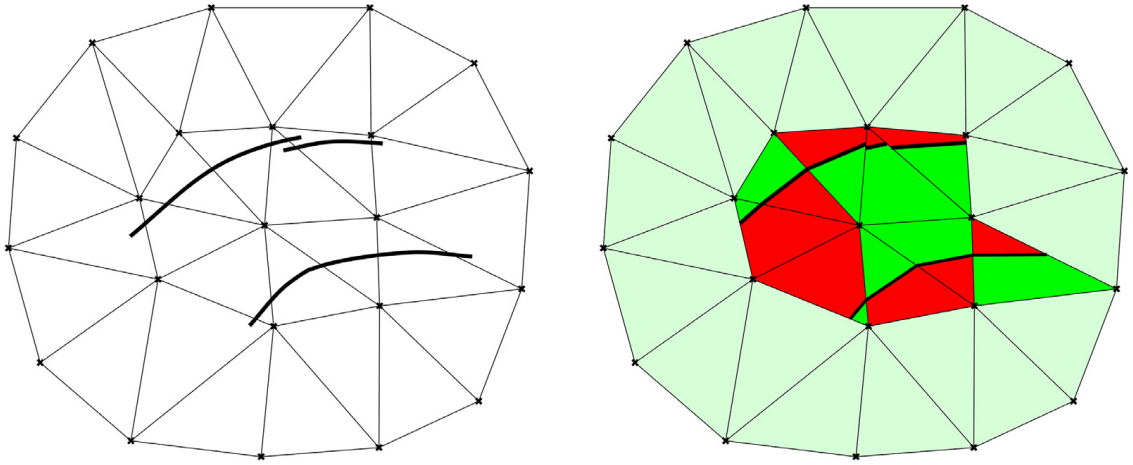


Fig. 2. Discontinuous distance function. Body without internal volume (left) and discontinuous distance representation (right). Red and green portions of the cut elements indicate the positive and negative discontinuous distance regions. Light green denotes the non-intersected elements. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

while the other one is orthogonal to it. Thus, the Cartesian velocity components u_x and u_y (as well as u_z in 3D) turn into a normal velocity component u_n together with a tangential one $u_{t,1}$ (besides another one $u_{t,2}$ in 3D).

In what follows, the methodology is described for a sample problem of the form $\mathbf{Ku} = \mathbf{f}$. Note that the same procedure holds for the N–S assembled system of equations if the rotation operations are applied to the velocity DOFs disregarding the pressure ones, which is to apply the rotation to the velocity DOFs submatrices.

Therefore, starting from the assembled system of equations $\mathbf{Ku} = \mathbf{f}$, the unknowns vector $\mathbf{u}$ can be split into two sets. As shown below, one set contains the slip boundary nodes DOFs $\mathbf{u}_\Gamma$, while the other contains the rest of the nodal unknowns $\mathbf{u}_{\text{int}}$

$$\begin{pmatrix} \mathbf{K}_{\text{int,int}} & \mathbf{K}_{\text{int},\Gamma} \\ \mathbf{K}_{\Gamma,\text{int}} & \mathbf{K}_{\Gamma,\Gamma} \end{pmatrix} \begin{pmatrix} \mathbf{u}_{\text{int}} \\ \mathbf{u}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{f}_{\text{int}} \\ \mathbf{f}_\Gamma \end{pmatrix} \tag{15}$$

Then, the rotation operator $\mathbf{P}$ is defined. Such operator keeps the orientation of the interior set of nodes $\mathbf{u}_{\text{int}}$ but reorients the slip boundary ones $\mathbf{u}_\Gamma$. Thus, by defining the nodal rotation operation $\mathbf{u} = \mathbf{P}\hat{\mathbf{u}}$ the rotated slip DOFs set $\hat{\mathbf{u}}$ can be obtained. The rotation operator $\mathbf{P}$ as well as the rotation operation can be expressed as

$$\begin{pmatrix} \mathbf{u}_{\text{int}} \\ \mathbf{u}_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{pmatrix} \begin{pmatrix} \hat{\mathbf{u}}_{\text{int}} \\ \hat{\mathbf{u}}_\Gamma \end{pmatrix} \tag{16}$$

being $\mathbf{R}$ the nodal rotation matrix and $\mathbf{I}$ the identity tensor.

The nodal rotation matrix $\mathbf{R}$ in the rotation operator $\mathbf{P}$ is computed node-by-node as

$$\mathbf{R}^n = \begin{pmatrix} v_{1,x} & v_{2,x} & n_x \\ v_{1,y} & v_{2,y} & n_y \\ v_{1,z} & v_{2,z} & n_z \end{pmatrix} \tag{17}$$

where $\mathbf{n}$ is the outwards unit normal vector and $\mathbf{v}_1$ and $\mathbf{v}_2$ are a pair of in-plane vectors orthogonal to $\mathbf{n}$. Note that this requires the computation of the unit normal vector not in the faces conforming to the slip boundary but on its nodes.

Then, the previous rotation operation is applied to the original system of equations as

$$\mathbf{P}^T\mathbf{KP}\hat{\mathbf{u}} = \mathbf{P}^T\mathbf{f} \tag{18}$$

to solve for the rotated set of DOFs $\hat{\mathbf{u}}$.

Once the global system of equations has been rotated, the slip boundary condition is nothing but a strong imposition of a stick condition in the orthogonal direction to the slip boundary. In other words, the slip boundary normal velocity u_n is set to 0 (or to the mesh velocity if an ALE framework is used) by a direct substitution of the DOFs value in the rotated unknowns vector. In the same way, the rotated global stiffness matrix $\mathbf{P}^T\mathbf{KP}$ rows

corresponding to u_n DOFs are all set to zero but the main diagonal component, which is set to 1 in order to enforce the imposed u_n value. The last step is the post-processing of the obtained solution $\hat{\mathbf{u}}$ to express it in terms of the original coordinate system. This is a node-by-node operation that can be easily done by computing the matrix vector product $\mathbf{u} = \mathbf{P}\hat{\mathbf{u}}$.

Finally, it is important to point out that it is not explicitly required, nor advisable for the sake of computational efficiency, to assemble the entire rotation operator. Hence, the proper implementation consists in looping the slip boundary nodes to locally perform the rotation operations by taking the assembled global system submatrices instead.

2.6. Embedded Nitsche slip boundary condition

The first formulation to impose the slip boundary condition in an embedded framework discussed in this work has been recently published by Winter et al. in [17]. This technique has been selected to serve as reference embedded slip solution because of its accuracy and stability properties. It consists in a stabilized Nitsche imposition of the general Navier condition described as

$$(\mathbf{u} - \mathbf{g})\,\mathbf{P}^n = \mathbf{0} \tag{19a}$$

$$\left(\varepsilon\left(\left[\mathbb{C} : \left(\boldsymbol{\nabla}^s \mathbf{u}\right)\right] \cdot \mathbf{n} - \mathbf{h}\right) + \mu\,(\mathbf{u} - \mathbf{g})\right)\mathbf{P}^t = \mathbf{0} \tag{19b}$$

where $\mathbf{g}$ and $\mathbf{h}$ are the velocity and the tangential traction to be imposed over the boundary. As it is clearly seen, the general Navier condition is composed by a normal contribution (Eq. (19a)) together with a tangential one (Eq. (19b)). The normal and tangential projection matrices are denoted as $\mathbf{P}^n$ and $\mathbf{P}^t$ and can be computed as $\mathbf{P}^n = \mathbf{n} \otimes \mathbf{n}$ and $\mathbf{P}^t = \mathbf{I} - \mathbf{n} \otimes \mathbf{n}$, being $\mathbf{I}$ the identity tensor.

Note that this boundary condition behaves as a wall-law in accordance to the slip length parameter ε. Therefore, it becomes a no-slip boundary condition when $\varepsilon = 0$ and a full-slip boundary condition when $\varepsilon \to \infty$ and $\mathbf{h} = 0$.

As it is mentioned above, the imposition of the boundary condition in Eq. (19) is done by using a stabilized Nitsche method in both the normal and tangential directions. The Nitsche imposition normal component reads as

$$\begin{aligned}\langle \frac{\mu}{\gamma h}\,(\mathbf{u_h} - \mathbf{g})\,\mathbf{P}^n, \mathbf{w}\rangle_\Gamma - \langle (\mathbf{u_h} - \mathbf{g})\,\mathbf{P}^n, \left(q\mathbf{I} + \zeta\mathbb{C} : \boldsymbol{\nabla}^s \mathbf{w}\right)\mathbf{n}\rangle_\Gamma + \\ \langle \frac{\phi_u}{\gamma h}\,(\mathbf{u_h} - \mathbf{g})\,\mathbf{P}^n, \mathbf{w}\rangle_\Gamma\end{aligned} \tag{20}$$

while the tangential one is

$$\begin{aligned}&\langle \frac{1}{\varepsilon + \gamma h}\left(\varepsilon\left(\mathbb{C} : \boldsymbol{\nabla}^s \mathbf{u_h}\mathbf{n} - \mathbf{h}\right) + \mu\,(\mathbf{u_h} - \mathbf{g})\right)\mathbf{P}^t, \mathbf{w}\rangle_\Gamma - \\ &\zeta\langle \frac{\gamma h}{\varepsilon + \gamma h}\left(\varepsilon\left(\left[\mathbb{C} : \left(\boldsymbol{\nabla}^s \mathbf{u_h}\right)\right] \cdot \mathbf{n} - \mathbf{h}\right) + \mu\,(\mathbf{u_h} - \mathbf{g})\right)\mathbf{P}^t, \left(\boldsymbol{\nabla}^s \mathbf{w}\right) \cdot \mathbf{n}\rangle_\Gamma\end{aligned} \tag{21}$$

being γ a penalty constant and $\zeta \in \{-1, 1\}$. If $\zeta = -1$ the Nitsche formulation is adjoint inconsistent. According to the original authors, the adjoint inconsistent formulation enjoys improved *inf–sup* stability for any value of the penalty constant γ [17]. Even though optimal convergence is not guaranteed for the velocity L^2-error in this case, the adjoint inconsistent formulation has been used in this work owing to its better stability properties. ϕ_u is a stabilization constant defined as

$$\phi_u = \mu + \rho\|\mathbf{u}\|h + \frac{\rho}{\Delta t}h^2 \tag{22}$$

More information regarding the implementation, stability analysis and validation can be found in [17].

2.7. Embedded discontinuous slip boundary condition

This subsection proposes an alternative approach to impose the slip boundary condition on embedded boundaries. The new technique is based on the use of the discontinuous Ausas FE space [37] in the intersected elements. The imposition of the slip BC is achieved by integrating the mass conservation equation by parts so that the condition is weakly applied on the embedded boundary. This is completed with the use of a penalty approach for the imposition of the same constraint in the momentum equation.

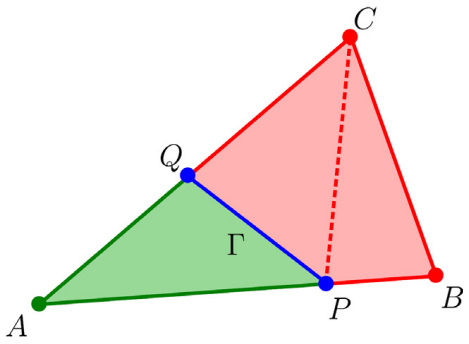
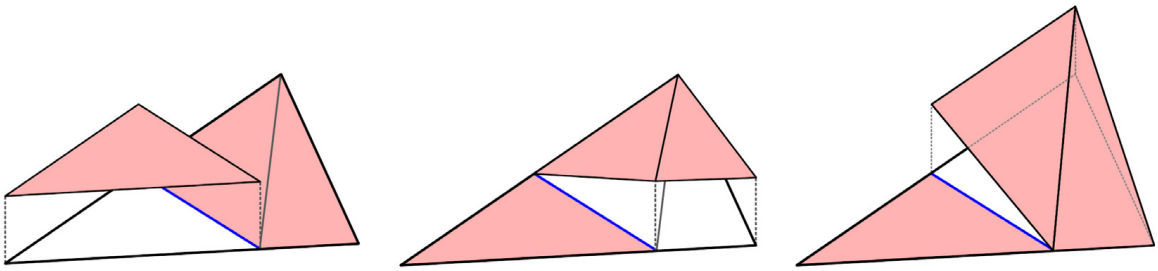


Fig. 3. Partition of a triangular finite element ABC into subelements following the interface PQ. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
Source: [37].



(a) N_A (b) N_B (c) N_C

Fig. 4. Triangle shape functions for the Ausas finite element space and the splitting pattern in 3.
Source: [37].

The Ausas FE space has been successfully used for representing the discontinuities arising from the resolution of two-phase flow problems in [37] and [38]. A similar idea is used in the present case, but aiming to capture the jump in both velocity and pressure fields coming from the immersion of an object in the fluid domain.

Furthermore, it is important to remark that the capability of representing discontinuities makes it possible to consider those cases in which the embedded body has no internal volume (e.g. boat sails). This ability overcomes the limitation of the slip formulation presented in the previous subsection [17] or other no-slip formulations in the literature [13], which require at least one internal node to perform the imposition.

On the other hand, it is also worth mentioning that the Ausas shape functions are conforming with the standard FE space ones used in the neighbouring elements. This property is extremely advantageous from the implementation point of view, since no modification is required in the blending elements (the ones that are attached to any split element), allowing them to use the standard FE space with no modification.

To describe the main geometrical features of the Ausas FE space, the same sample splitting pattern used by the original authors in [37] is taken and presented again in Fig. 3. By inspecting the shape functions representation derived from the previous splitting pattern (Fig. 4) the next two main attributes can be noted:

- The shape function values along an intersected edge are constant. For instance, in the example depicted in Fig. 3, the shape function at the intersection points Q and P takes the same value than in A on the green side. On the other hand, on the red side the value at the intersection point Q takes the same value than in C. The same can be said for P and B.
- The shape function gradients are approximatively null in the intersection normal direction. This property makes them suitable for the application of the slip BC, since the absence of tangential stresses corresponds to a zero gradient in the normal direction.

Concerning the null tangential stress requirement, it is approximatively satisfied inside the elements because of the inability of the Ausas FE space to capture the gradient in the intersection normal direction. Moreover, the zero

shear stress Neumann boundary condition is imposed over the interface cuts as

$$\langle \mathbf{w},\, p_h \mathbf{n} - \mathbf{t_h} \rangle_\Gamma = \langle \mathbf{w},\, 2p_h \mathbf{n} - \left(\mathbb{C} : \boldsymbol{\nabla}^s \mathbf{u_h}\right) \cdot \mathbf{n} \rangle_\Gamma = 0 \tag{23}$$

If the Neumann boundary condition in Eq. (23) is added to the original boundary term coming from the integration by parts of the momentum equation, both contributions can be condensed in a unique boundary term

$$\langle \mathbf{w},\, 2p_h \mathbf{n} - \mathbb{C} : \boldsymbol{\nabla}^s \mathbf{u_h} \cdot \mathbf{n} \rangle_\Gamma + \langle \left(\mathbb{C} : \boldsymbol{\nabla}^s \mathbf{u_h}\right) \cdot \mathbf{n} - p_h \mathbf{n} \rangle_\Gamma = \langle \mathbf{w},\, p_h \mathbf{n} \rangle_\Gamma \tag{24}$$

which substitutes the boundary term in the functional depicted in Eq. (12).

Two additional strategies are adopted for the imposition of the non-penetrability constraint. The first one is to integrate by parts one of the mass conservation equation terms as

$$-\left(q, \boldsymbol{\nabla} \cdot \mathbf{u_h}\right)_\Omega = \left(\boldsymbol{\nabla} q, \mathbf{u_h}\right)_\Omega - \langle q, \mathbf{u_h} \cdot \mathbf{n} \rangle_\Gamma \tag{25}$$

As a consequence, a boundary term which includes the normal projection of the velocity appears. Hence, a weak imposition of the non-penetrability condition can be done by dropping this term only in the embedded interface boundary integrals. It is important to recall that this new boundary term must be considered in all the other domain boundaries.

The second strategy to impose the non-penetrability requirement is to add a penalty constraint to the normal projection of the velocity in the momentum conservation equation. This new term reads as

$$\langle \kappa \mathbf{w},\, \mathbf{u_h} \mathbf{n} \otimes \mathbf{n} \rangle_\Gamma = 0 \tag{26}$$

where κ stands for the consistent penalty coefficient computed as

$$\kappa = \frac{C_{Pen}\left(\frac{\rho h^d}{\Delta t} + \mu h^{d-2} + \rho \|\mathbf{u}\| h^{d-1}\right)}{A_{int}} \tag{27}$$

where C_{Pen} is a constant, which is set to 10.0 in all the cases discussed in this work, d is the working dimension (2 in 2D and 3 in 3D) and A_{int} is the intersection area.

To sum up, the proposed formulation implements the two requirements of a slip boundary condition, which are null normal projection of the velocity and null tangential stress, in a weak sense by:

- integrating the mass conservation equation by parts and dropping the boundary term on the elemental cuts.
- using a penalty constraint to the normal component of the velocity.
- using the discontinuous FE space presented in [37] in the intersected elements together with a null tangential stress Neumann boundary condition imposition in the elemental cuts.

3. Validation examples

This section is intended to describe the benchmarking task, which has been conducted to test and compare all the implemented functionalities with low and high Re in both 2D and 3D. Hence, the geometry, boundary conditions and simulation settings that are required to reproduce the presented experimental cases are detailed together with the discussion of the obtained results. A relation of the presented benchmark cases and a brief motivation is listed below

- 2D flow inside a ring: to obtain the convergence rates of the presented formulation and assess its behaviour with faceted approximations of curved geometries.
- 2D squeezing flow: to obtain the convergence rates of the presented formulation in the high Re and inviscid scenarios.
- 2D elbow with internal wall: to test the presented formulation when modelling embedded flow discontinuities at low Re number.
- 2D flow around cylinder: to compare the three slip formulations when modelling bodies with internal volume at high Re number.
- 2D flow around vertical plate: to prove the capabilities of the presented discontinuous embedded formulation when modelling volumeless immersed bodies.
- 2D divergent channel with moving cylinder: to preliminary assess the capabilities of the discussed embedded formulations when facing moving boundaries problems.

Table 1
2D flow inside a ring mesh refinement settings.

	Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5	Mesh 6
Element size [m]	0.1428	0.06667	0.03448	0.01754	0.00884	0.00444	0.00223
Radial divisions	7	15	29	57	113	225	449
Perimeter divisions	21	43	85	169	337	673	1345

- 3D elbow with internal wall: to test the presented discontinuous embedded formulation in a 3D case.
- 3D flow around two boat sails: to show a potential industrial application of the presented discontinuous embedded formulation.

3.1. 2D flow inside a ring

This first example has two main objectives. The first one is to obtain the velocity and pressure convergence rates of the presented formulation. The second one is to assess the behaviour of the method when dealing with polygonal approximations of curved boundaries.

As pointed out in [16], the imposition of slip boundary conditions on curved boundaries might become an issue when such faceted approximations are used. In [16], the method of manufactured solutions is applied to check the convergence of a Stokes formulation together with a Nitsche-based slip boundary condition imposition. According to the authors, it can be observed that velocity approximations almost vanish at the boundary vertices of the curve approximation when convergence does not hold. They associate this behaviour to the fact that the slip boundary condition tends to impose the velocity field to be parallel to each face of the polygonal approximation, which is to say to vanish at the common vertex between faces.

The geometry of the problem, which is equivalent to the one discussed in [16], consists of two concentric cylinders generating an interior fluid cavity. Thus, the fluid domain Ω can be described as $(x, y) \in \mathbb{R}^2 : \ 0.25 \leq x^2 + y^2 \leq 1$ (a ring with inner radius equal to 0.5 m and outer radius equal to 1.0 m). Initially, the fluid is at rest, but a movement is induced by imposing a constant angular velocity in the outer cylinder. The analytical solution which corresponds to the application of a slip BC over the inner boundary is

$$v_r = 0 \tag{28a}$$

$$v_\Theta = r \tag{28b}$$

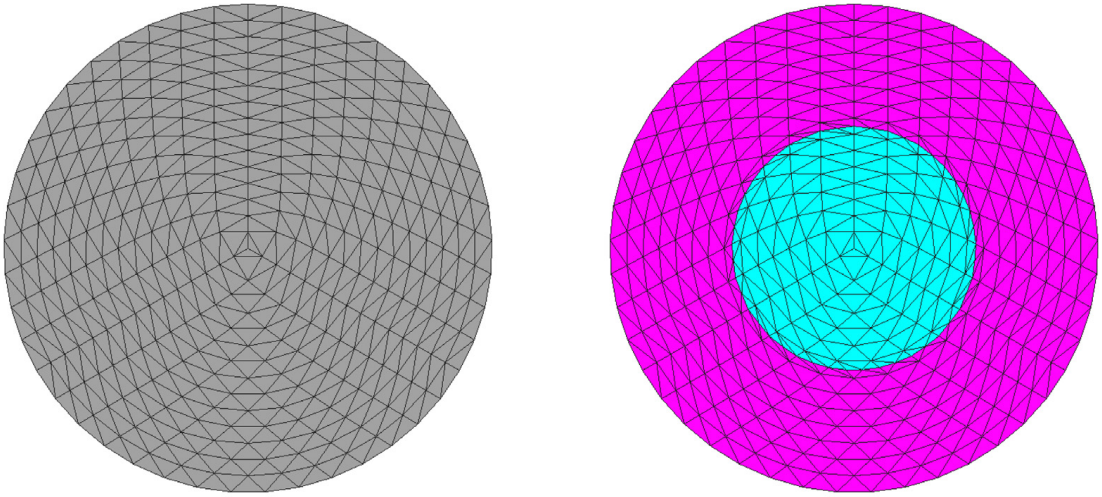
$$p = \frac{\rho}{2}\left(r^2 - 1\right) \tag{28c}$$

The dynamic viscosity μ and density ρ are 1.0 kg/m^3 and 1e−3 kg/ms. Regarding the boundary conditions, the velocity and pressure fields are fixed to the analytical solution along the outer contour. The BDF2 time scheme is used for the time discretization. 10 time steps of 200 s (Δt) are computed, resulting in a total simulation time of 2000 s, which guarantees that a steady-state solution is reached.

A centred structured mesh is used in all the cases. The domain is meshed without taking the inner cylinder into account, which is represented by a radial discontinuous distance function. The elemental sizes for each refinement level as well as the radial and perimeter subdivisions are collected in the Table 1. Fig. 5a depicts the mesh after the first refinement (Mesh 1 in Table 1) while Fig. 5b details the intersection pattern which results from the distance function. As it can be observed, the pink region in Fig. 5b represents the fluid domain whereas the light blue one is the interior part of the inner cylinder.

Table 2 collects the $L^2(\Omega)$-norm of both the velocity and pressure errors. Figs. 6 and 7 display a convergence rate of the order of $h^{3/2}$ for both the velocity and the pressure unknowns. These results are in line with the ones obtained by Ausas et al. in [37] and are perfectly expected due to the worse interpolation properties of the modified discontinuous FE space used in the intersected elements.

Finally, the velocity and the pressure fields are shown in Fig. 8 for the coarsest, intermediate and finest meshes. The coarsest mesh solution is far from the expected one, while the intermediate and finest meshes give a good approximation. As observed by Urquiza et al. in [16], the coarsest mesh is very inaccurate, however upon mesh refinement the method converges to the analytical solution.



(a) Mesh 1 capture. (b) Distance intersection pattern.

Fig. 5. 2D flow inside a ring. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

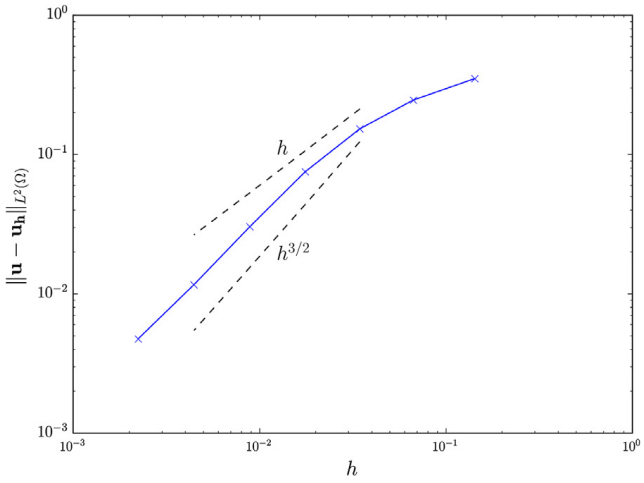


Fig. 6. 2D flow inside a ring. Velocity convergence rates. Solid lines represent the obtained results. Dashed lines represent h and $h^{3/2}$ convergence rates.

Table 2
2D flow inside a ring velocity and pressure error norms.

	Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5	Mesh 6
$\|\mathbf{u} - \mathbf{u_h}\|_{L^2(\Omega)}$	0.351411	0.245968	0.15305	0.075216	0.030324	0.011653	0.004751
$\|p - p_h\|_{L^2(\Omega)}$	0.149445	0.080384	0.050110	0.025281	0.010379	0.004016	0.001641

3.2. 2D squeezing flow

The aim of this test is to evaluate the convergence rates of the presented formulation for the transient high Re and inviscid scenarios. To that purpose, the squeezing flow between two moving plates with time dependent velocity is solved. However, only the top half of the domain is considered. The symmetry condition is used to verify the proposed embedded slip formulation.

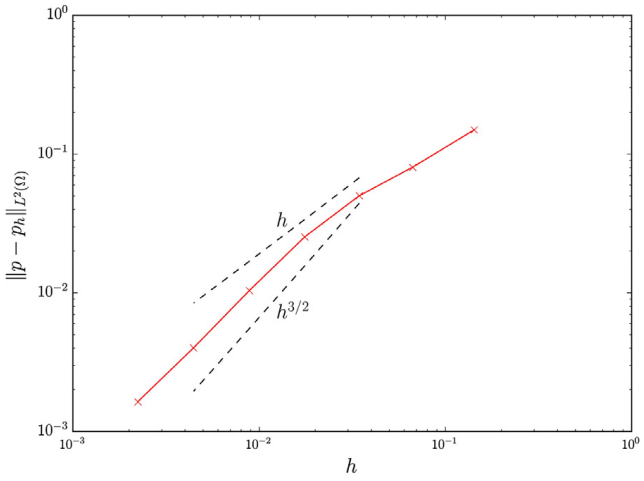


Fig. 7. 2D flow inside a ring. Pressure convergence rates. Solid lines represent the obtained results. Dashed lines represent h and $h^{3/2}$ convergence rates.

Table 3
2D squeezing flow mesh refinement settings.

	Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4
Element size [m]	0.008333	0.004166	0.002083	0.001042	0.000521
Vertical divisions	31	61	121	241	481
Horizontal divisions	120	240	480	960	1920

After such simplification, the problem geometry consists of a rectangular shaped domain of 1 × 0.25 m (width × height). A space and time dependent velocity equal to

$$v_y = \begin{cases} -2tx & x < 0.5 \\ -2(1-x)t & x \geq 0.5 \end{cases} \tag{29}$$

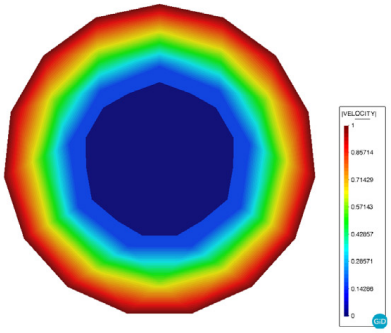
is imposed to the top plate. Since a symmetric solution is expected, the horizontal velocity component v_x is also fixed to 0 on the top edge midpoint. A free outlet condition is assumed in both the left and right edges. Zero dynamic viscosity and unit density are considered in the inviscid limit case. For the high Re scenario, the dynamic viscosity is set to 5e−9 kg/ms to have a Re number equal to 10^5Re.

The reference solutions ($\bar{\mathbf{u}}$, $\bar{p}$) for the convergence study are computed using an extremely fine body-fitted mesh. On the other hand, Table 3 collects the mesh settings (horizontal and vertical edges subdivisions) for the embedded cases. Note that the embedded meshes height is 0.3 m and a distance field $d(x, y) = y$ is used to get an equivalent computational domain.

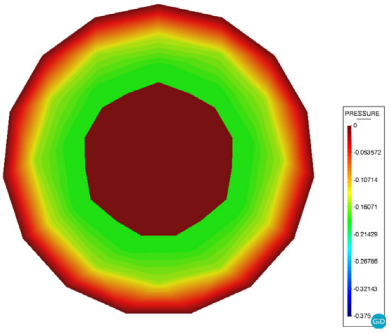
The BDF2 scheme is used for the time discretization. A total of 3 time steps of 1e−5 s (Δt) are run in both test cases. A previous convergence study using the body fitted solution proved that the selected time step value is small enough to ensure that the error in space dominates the time dependent one for the reference mesh.

Tables 4 and 5 collect the $L^2(\Omega)$-norm of both the velocity and pressure errors for the 10^5Re and inviscid scenarios. As it can be observed from Figs. 9 and 10, the velocity and pressure convergence rate is around h^2 for the 10^5Re case. On the other hand, Figs. 11 and 12 show that in the inviscid limit the convergence rates for both velocity and pressure fields deteriorate to h. This behaviour might be explained by the fact that the penalty constant κ (Eq. (27)) converges to a finite value in the inviscid limit. This can be seen as, in the assumption that $A_{int} \approx h^{d-1}$, $h \to 0$ and $\mu \to 0$, the penalty coefficient κ converges to $C_{Pen}\rho\|\mathbf{u}\|$. We observe however that to the best of our knowledge classical convergence estimates are not valid for the inviscid scenario.

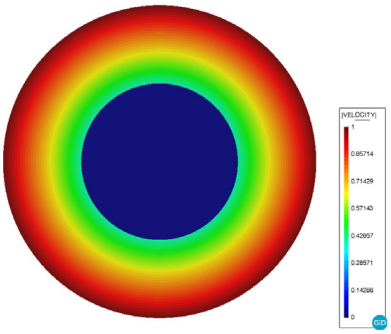
Finally, the obtained velocity and pressure fields are shown in Figs. 13 and 14 for the reference body fitted and embedded solutions. As it is clearly seen, no differences can be detected between the two solutions, showing that the formulation also works for the transient high Re and inviscid limit cases.



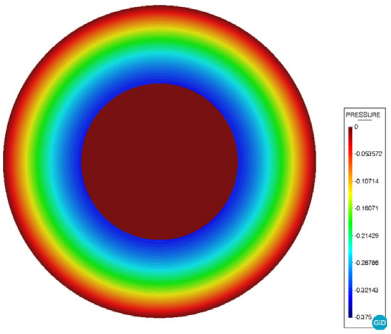
(a) Mesh 0 velocity.



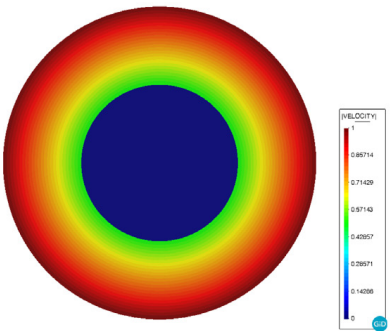
(b) Mesh 0 pressure.



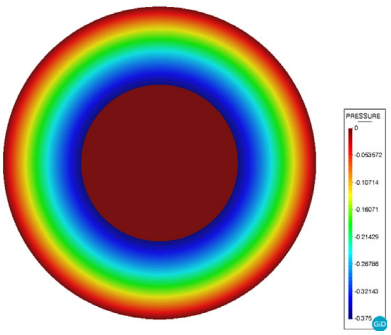
(c) Mesh 3 velocity.



(d) Mesh 3 pressure.



(e) Mesh 6 velocity.



(f) Mesh 6 pressure.

Fig. 8. 2D flow inside a ring. Coarsest, intermediate and finest meshes solutions.

Table 4
2D squeezing flow velocity and pressure error norms (10^5Re).

	Mesh 0	Mesh 1	Mesh 2	Mesh 3
$\|\bar{\mathbf{u}} - \mathbf{u_h}\|_{L^2(\Omega)}$	8.22085e−7	5.78898–7	1.22989e−7	3.33973e−8
$\|\bar{p} - p_h\|_{L^2(\Omega)}$	3.39966e−2	2.38613e−2	4.60455e−3	8.71143e−4

Table 5
2D squeezing flow velocity and pressure error norms (inviscid).

	Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4
$\|\bar{\mathbf{u}} - \mathbf{u_h}\|_{L^2(\Omega)}$	8.83323e−7	7.52714e−7	2.6534e−7	1.41099e−7	7.18316e−8
$\|\bar{p} - p_h\|_{L^2(\Omega)}$	3.65597e−2	3.10994e−2	1.06083e−2	5.45133e−3	2.76408e−3

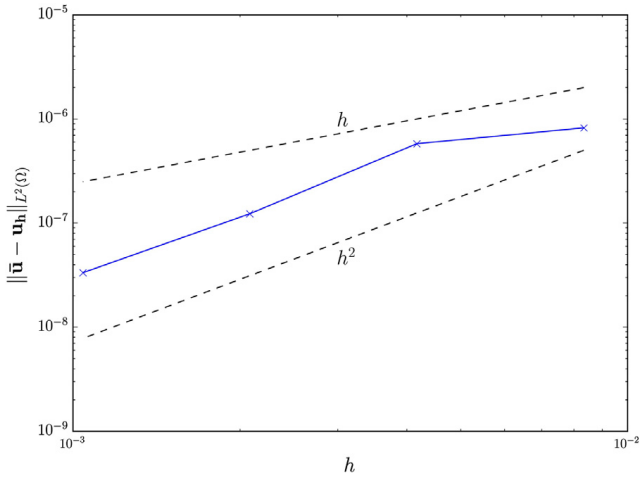


Fig. 9. 2D squeezing flow. Velocity convergence rates (10^5Re). Solid lines represent the obtained results. Dashed lines represent h and h^2 convergence rates.

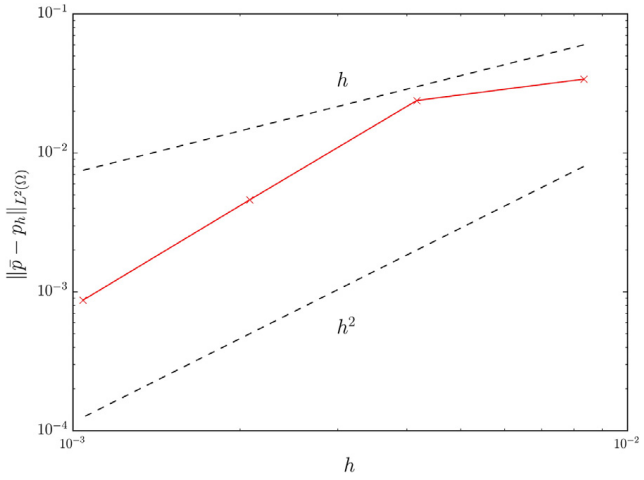


Fig. 10. 2D squeezing flow. Pressure convergence rates (10^5Re). Solid lines represent the obtained results. Dashed lines represent h and h^2 convergence rates.

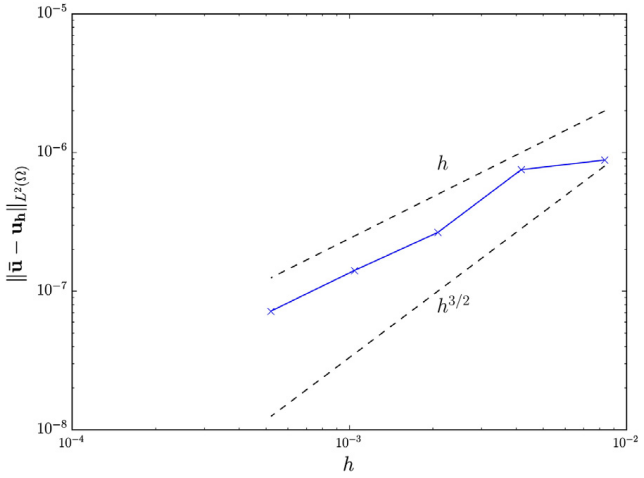


Fig. 11. 2D squeezing flow. Velocity convergence rates (inviscid). Solid lines represent the obtained results. Dashed lines represent h and $h^{3/2}$ convergence rates.

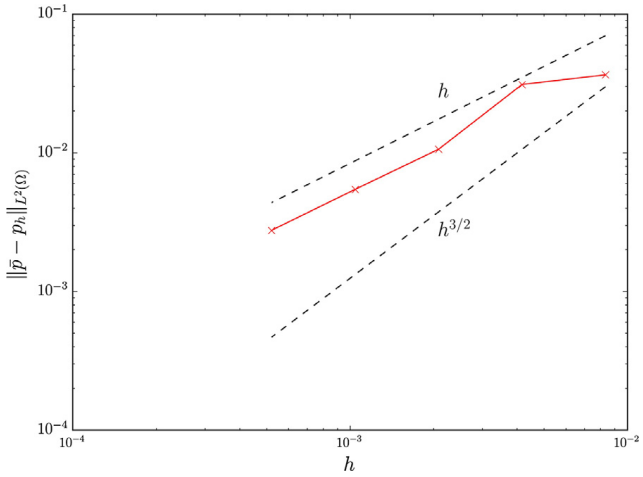


Fig. 12. 2D squeezing flow. Pressure convergence rates (inviscid). Solid lines represent the obtained results. Dashed lines represent h and $h^{3/2}$ convergence rates.

Table 6
2D elbow with internal wall number of elements for different refinement levels.

Mesh	Reference	Present work
Coarse	2 400	2 300
Medium	9 600	8 900
Fine	38 400	36 200
Very fine	153 600	147 000

3.3. 2D elbow with internal wall

The purpose of this test is to assess the correctness and performance of the presented discontinuous formulation, as well as its capability of modelling embedded flow discontinuities. For that aim, the presented theoretical example, which has analytical solution, has been solved with several mesh refinement levels. The obtained solution is compared with the body fitted and literature ones.

This example was firstly proposed in [39], and consists of a 90° curved 2D pipe conforming to an elbow shape. A zero-thickness wall is placed inside the curved pipe, generating two separated fluid ducts with varying cross section area. The geometry of the problem is depicted in Fig. 15.

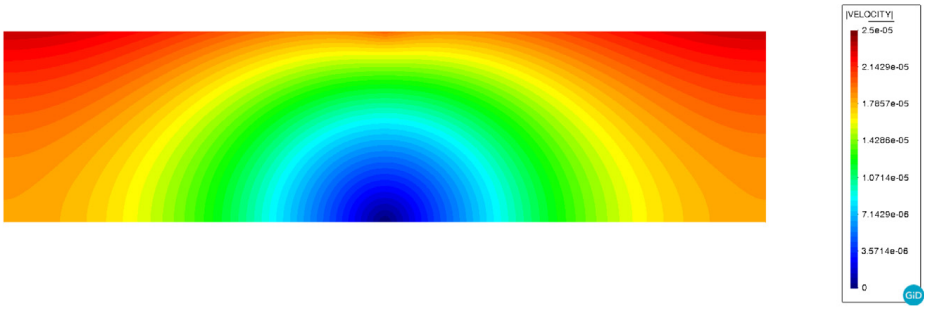
The fluid properties are selected such that the Re number has unit value. Taking the radius of each one of the ducts as reference length, such unit Re number can be achieved by setting density ρ to 1 kg/m^3, dynamic viscosity μ to 1 kg/ms and a constant inlet velocity u_x of 1 m/s. Furthermore, the pipe walls are assumed to be no-slip and the pressure is fixed to zero on the outlet.

Concerning the time discretization, the BDF2 time scheme is used with a time step (Δt) of 10^{-2} s. The total simulation time is 1 s, which is enough to reach a stationary solution.

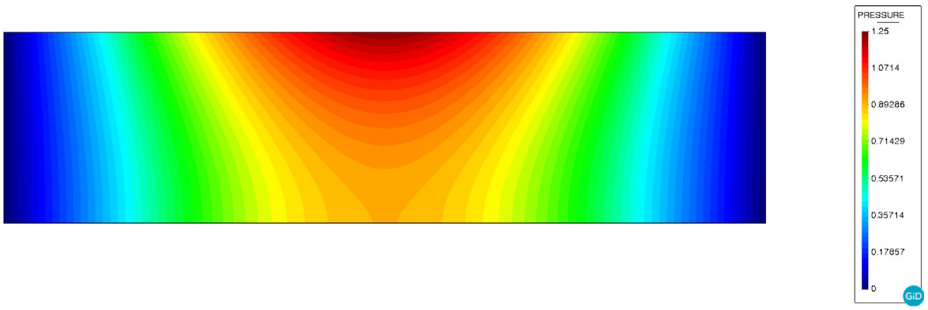
The meshes employed were as similar as possible to the ones used for the reference solution. Thus, four structured triangular grids were used. The number of elements employed on each grid, together with the reference ones in [39], are presented in Table 6.

In the following, the performance of the proposed discontinuous embedded formulation is assessed. First of all, the new formulation results are presented. After that, the obtained solution is compared with the expected and body-fitted ones and a different discontinuous formulation proposed in [39]. Besides that, the results of the mesh refinement study are also discussed.

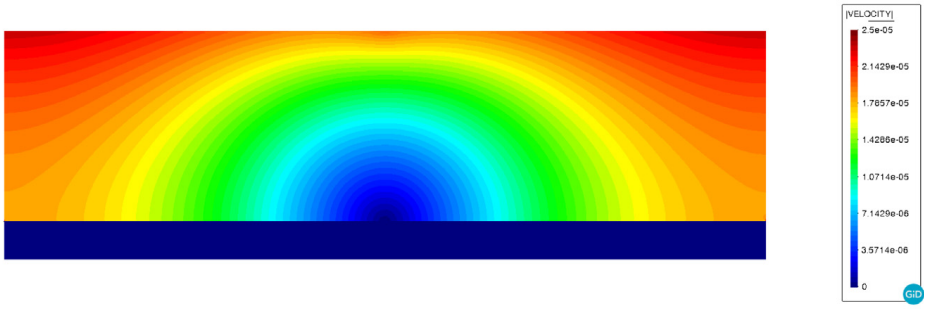
Fig. 16 shows the presented discontinuous formulation velocity and pressure fields for the medium mesh. As expected, the contraction generated by the wall in the right duct induces a pressure gradient, which turns to a flow acceleration that preserves the inlet flow rate. The opposite behaviour can be observed in the left duct, since the cross section remains constant.



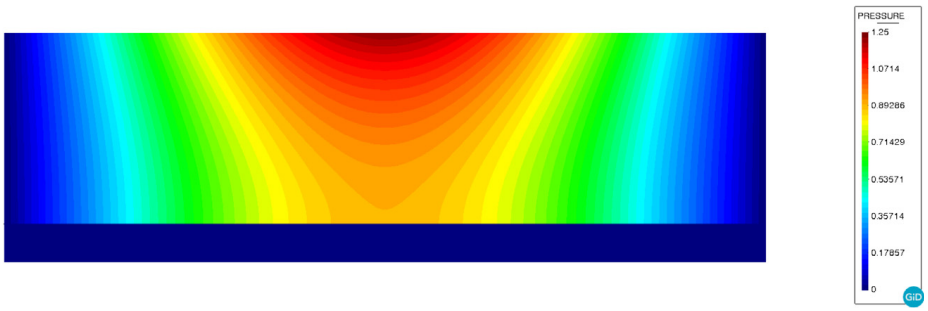
(a) Body fitted velocity.



(b) Body fitted pressure.



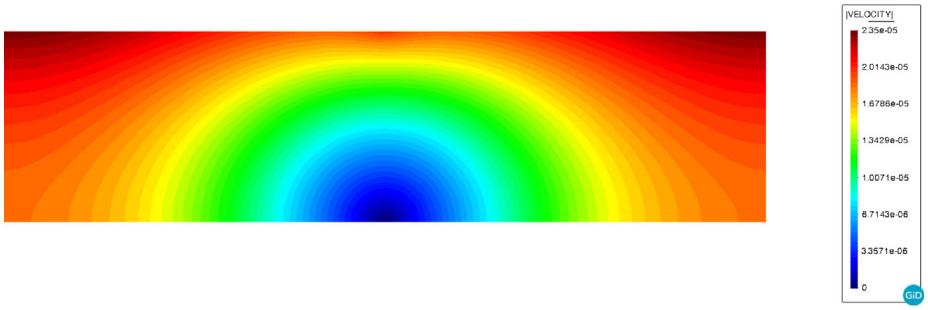
(c) Embedded mesh 3 velocity.



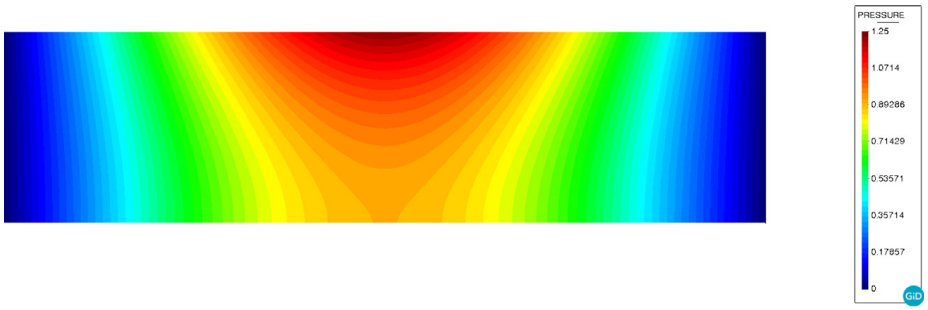
(d) Embedded mesh 3 pressure.

Fig. 13. 2D squeezing flow. Reference body fitted and embedded solutions (10^5Re).

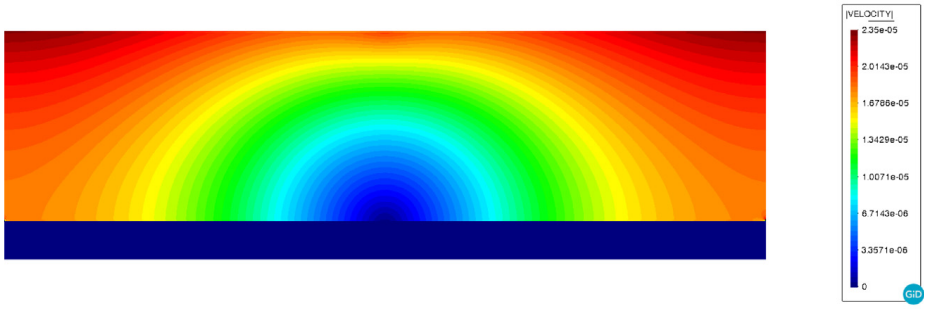
The medium mesh results obtained with the presented discontinuous embedded formulation are compared with the body fitted solution and with the reference ones in Fig. 17a. Such comparison is done by means of the y-component velocity distribution at the outlet. At a first glance, an extremely good correlation can be seen. However, minor differences can be observed in the values close to the wall. Taking into account that the meshes



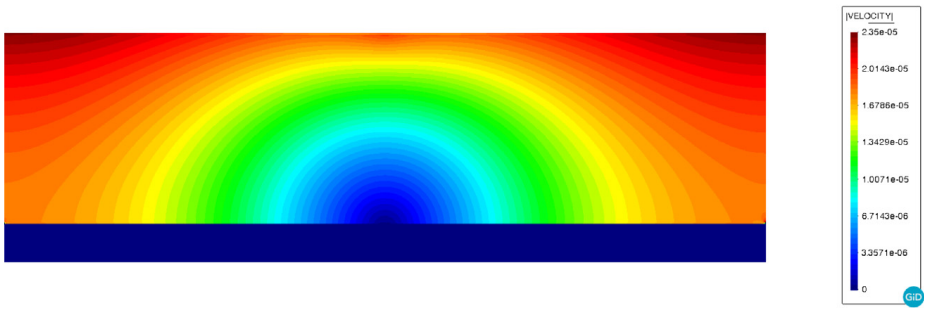
(a) Body fitted velocity.



(b) Body fitted pressure.



(c) Embedded mesh 4 velocity.



(d) Embedded mesh 4 pressure.

Fig. 14. 2D squeezing flow. Reference body fitted and embedded solutions (inviscid).

are quite similar but not perfectly equal between the three cases, sufficiently good agreement is found to asseverate that the proposed discontinuous embedded formulation correctly solves the posed problem.

A mesh refinement study is done by using the proposed embedded formulation. The y-component of the outlet velocity is used again as the comparison magnitude. As it can be observed in Fig. 17b, the results in the region close to the internal wall improve as long as the mesh is refined.

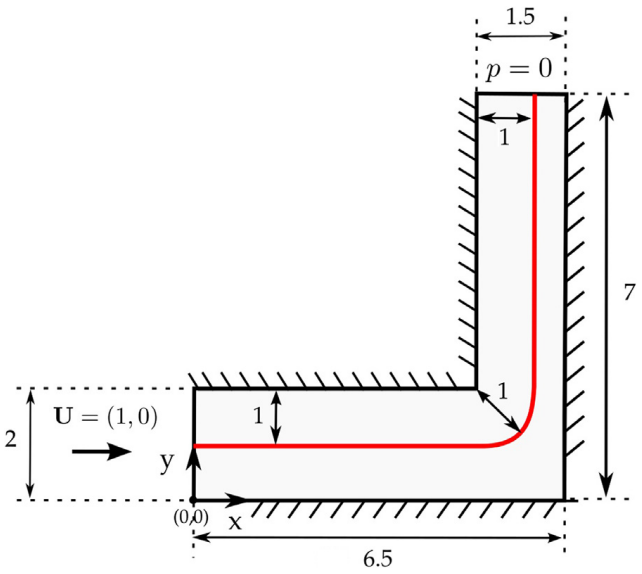
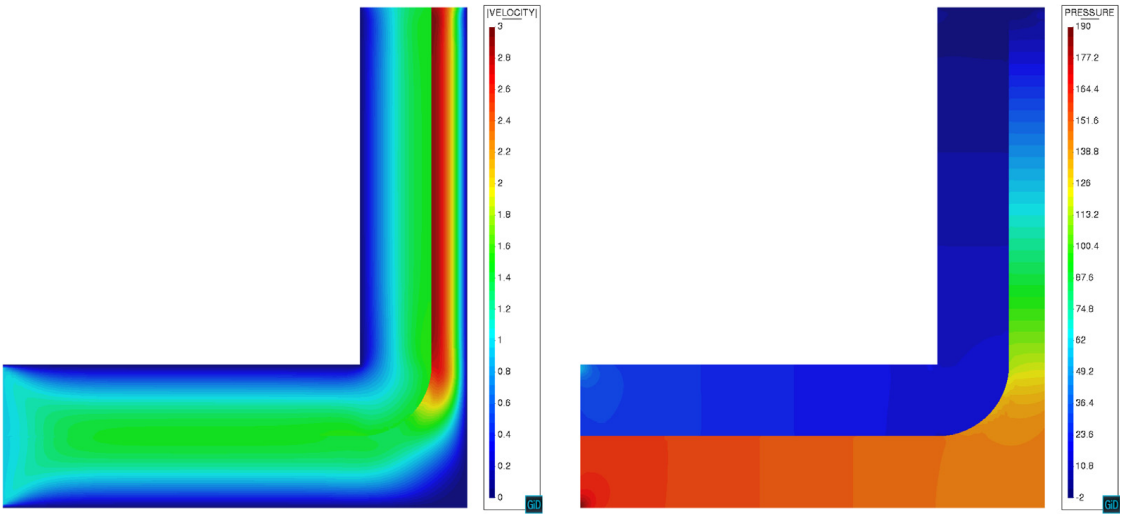


Fig. 15. 2D elbow with internal wall.
Source: [39].



(a) Velocity modulus (m/s). (b) Pressure (Pa).

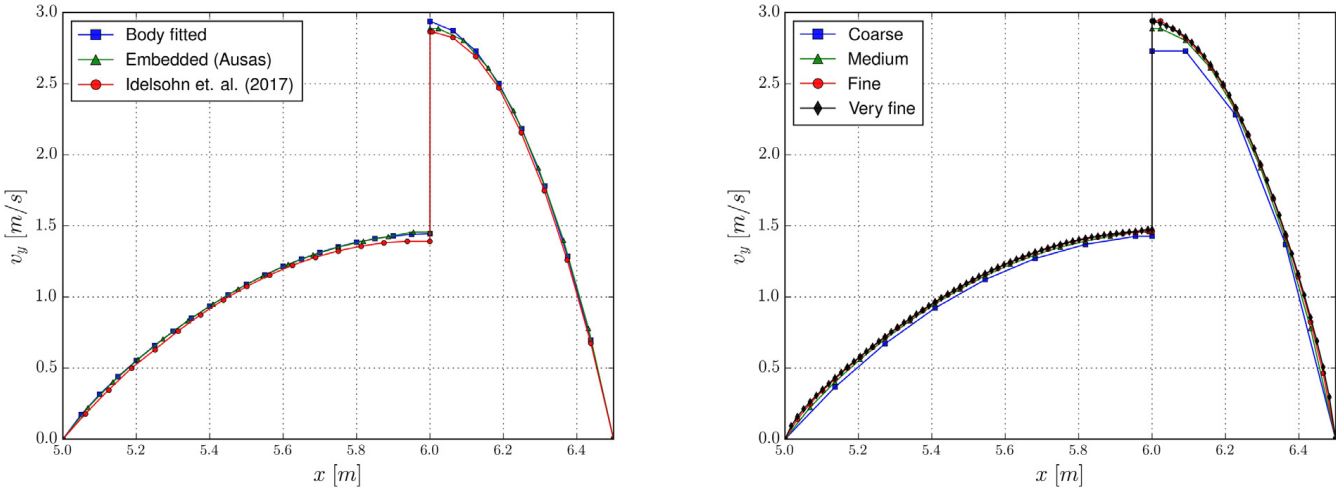
Fig. 16. 2D elbow with internal wall. Discontinuous embedded solution (medium mesh).

Table 7 shows the maximum y-component velocity values in the outlet of both sides of the membrane. As it pointed out in [39], considering that the flow is parabolic in both sides of the internal wall, it can be proven that the analytical maximum velocity must be 1.5m/s and 3m/s in the left and right ducts. Taking these two values as a reference, the absolute errors are also computed and presented in Table 7. The values converge to the expected solution as the mesh is refined.

3.4. 2D flow around cylinder

The fourth benchmark is the inviscid incompressible flow around a cylinder. The purpose of this test is to compare the two described embedded slip formulations with the body fitted one. This problem has analytical solution, so the three presented solutions are also compared with the theory.

The problem geometry consists of a cylinder with radius 0.1 m, which has its centre point in (0.0,0.0) m, placed inside a 10.0 × 4.0 m channel. The bottom left corner of the channel is located in (−2.0,−2.0) m. Note that, the



(a) Medium mesh comparison. (b) Ausas solutions comparison.

Fig. 17. 2D elbow with internal wall. Outlet v_y.

Table 7
2D elbow with internal wall. Maximum vertical velocity in the outlet using different meshes [m/s].

Expected	$\|\mathbf{v}_\mathbf{y}\|^{left}_{max}$ 1.5	abs. err. –	$\|\mathbf{v}_\mathbf{y}\|^{right}_{max}$ 3.0	abs. err. –
Coarse	1.4261	0.0739	2.7301	0.2699
Medium	1.4561	0.0439	2.8874	0.1126
Fine	1.4630	0.0370	2.9413	0.0587
Very fine	1.4759	0.0241	2.9367	0.0633

Table 8
2D flow around cylinder mesh settings.

Region	A	B	C	D
Mesh size [m]	0.05	0.025	0.01	0.001

channel size is set such that the blockage coefficient is equal to the 5%. This ensures that the problem boundaries do not affect the solution around the immersed cylinder. A constant inlet is imposed at the left edge, symmetry boundary conditions are used in both the top and bottom edges and the pressure is fixed to zero at the right end edge.

Owing to the inviscid assumption, the Re number is infinite. The dynamic viscosity μ is set to 0.0 kg/ms, the density ρ to 1 kg/m^3 and the inlet velocity to 1 m/s. Concerning the time discretization, the BDF2 scheme is used with a time step Δt of 10^{-2} s and a total simulation time of 1 s.

A distance modification criterion is used to avoid badly defined intersections that might compromise the convergence of the embedded cases. Thus, when the distance value is less than 0.01% the elemental size (i.e. the zero of the level set function is close to a node) the distance field is corrected by slightly modifying it as it is explained in the next paragraph. This makes it possible to avoid the use of small-cut stabilization techniques [14].

For the continuous distance case (Nitsche formulation), this is only needed in those elements where the fluid portion is almost null. Thus, the threshold tolerance is set with negative sign to deactivate the almost empty elements. On the contrary, this is not a problem when using the presented discontinuous formulation, since both sides of the level set are assembled and solved. Therefore, an absolute threshold criterion is enough to ensure that the distance value is not too close to 0.

Fig. 18 describes the problem geometry as well as the auxiliary meshing subregions, which mesh sizes are collected in Table 8.

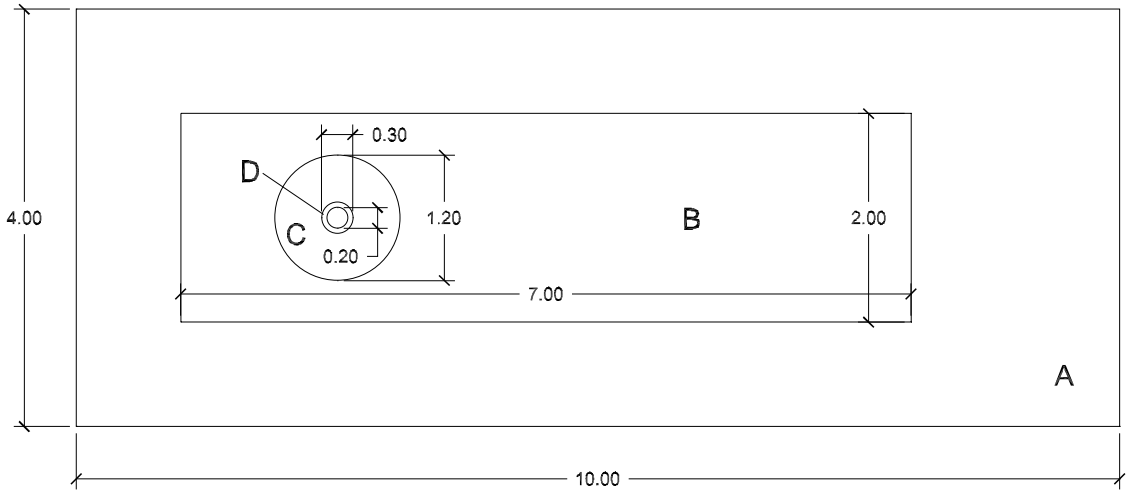


Fig. 18. 2D flow around cylinder problem geometry and mesh regions.

The pressure coefficient (C_p) is selected as reference comparison magnitude. From a known pressure in the far field p_∞, the pressure coefficient can be computed in each node as

$$C_{p,i} = \frac{p_i - p_\infty}{\frac{1}{2}\rho \|\mathbf{v}_i\|^2} \tag{30}$$

The C_p analytical solution for the presented case can be found in [40] and is computed as

$$C_p(\theta) = 1 - 4\sin^2(\theta) \tag{31}$$

Fig. 19 describes the evolution of the pressure coefficient according to the angle θ, being $\theta = 0$ in the cylinder downstream point. At first glance, the evolution of the body fitted C_p has almost perfect agreement with the analytical solution. With regard to both embedded formulations, no differences can be observed in the evolution of C_p if they are compared with the body fitted solution.

However, if it is zoomed in on the downstream region ($\theta \approx 0$) minor disparities with respect to the analytical solution are observed (Fig. 20a). In this region, the C_p values are slightly below the expected ones, confirming that both embedded formulations are a bit more diffusive than the body fitted one. Such numerical diffusivity results in an artificial flow separation located in the leeward region of the cylinder.

On the other hand, it is worth to comment something on the results obtained for either the top ($\theta \approx \pi/2$) or bottom ($\theta \approx 3\pi/2$) parts of the cylinder. By a close-up inspection, it can be noted that minor oscillations appear in both embedded solutions (Fig. 20b). Such oscillations are much less significant in the Nitsche solution and are most likely to be associated to poorly-conditioned intersection patterns. Furthermore, the Ausas FE space solution presents a staggered pattern, reflecting the interpolation properties of the method.

The expected and obtained peak C_p values are presented in Table 9 for each one of the presented solutions. The relative errors of such values with respect to the analytical solution are collected in Table 10. As it has been pointed before, significant differences between the different formulations can only be found in $\theta = 0$ region. In this location, the C_p relative error of the Nitsche formulation is 11.3% while the discontinuous Ausas formulation one is 12.29%.

Finally, the velocity field contour lines surrounding the cylinder are shown in Fig. 21. If both embedded solutions are compared with the body fitted one, the previously commented small flow separation in the downstream region of the cylinder becomes evident. Such artificial flow separation is slightly more significant in the discontinuous formulation case, which is something that can be expected due to the worse interpolation properties of the Ausas FE space.

3.5. 2D flow around vertical plate

The objective of this test is to assess the capability of the proposed discontinuous formulation to model bodies without internal volume at high Re numbers. As commented before, the viscous-slip approach makes sense when

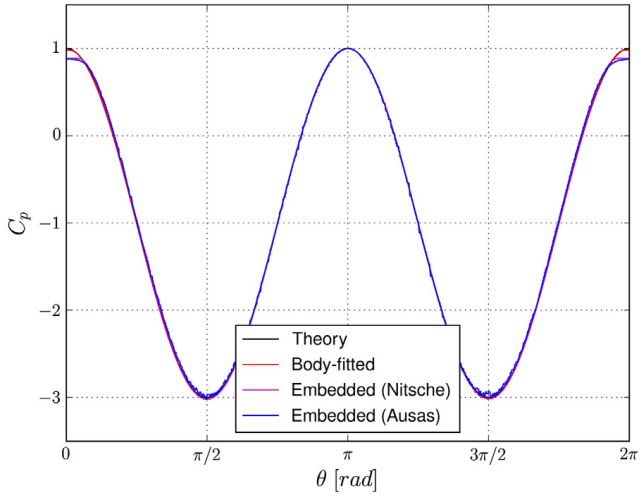
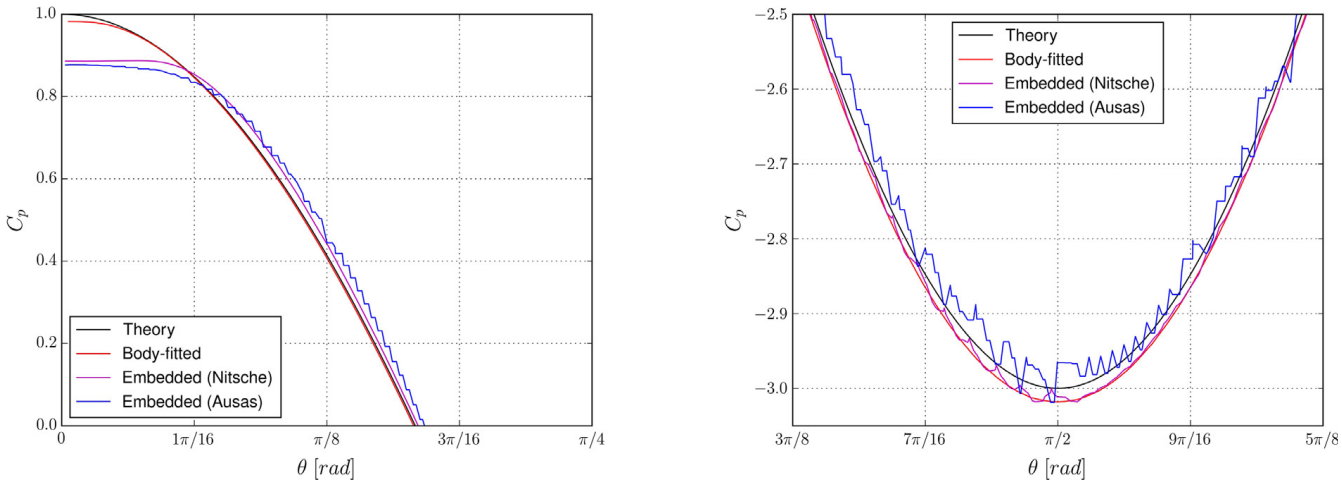


Fig. 19. Inviscid incompressible 2D flow around a cylinder. $C_p(\theta)$ complete evolution.



(a) $\theta \in [0, \pi/4]$ detail. (b) $\theta \in [3\pi/8, 5\pi/8]$ detail.

Fig. 20. Inviscid incompressible 2D flow around a cylinder. $C_p(\theta)$ evolution details.

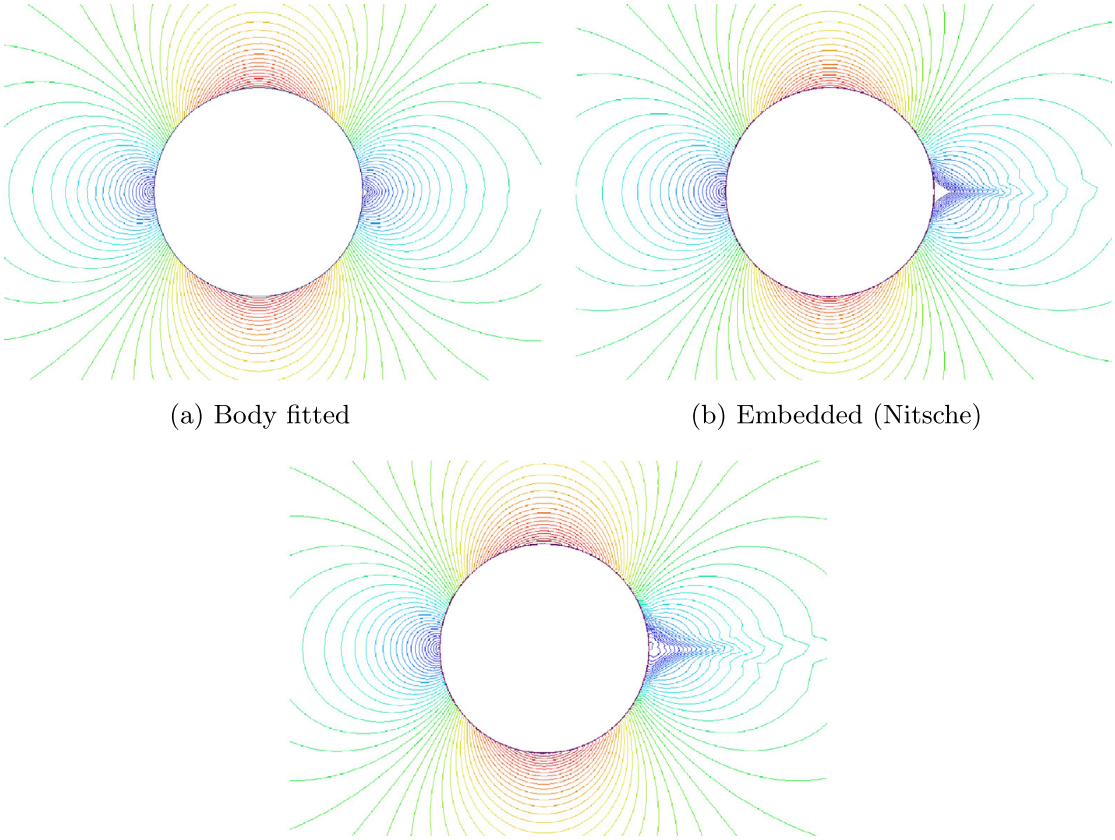
Table 9
Inviscid incompressible 2D flow around a cylinder. C_p peak values.

Formulation	C_p			
	$\theta = 0$	$\theta = \frac{\pi}{2}$	$\theta = \pi$	$\theta = \frac{3\pi}{2}$
Theory	1.0	−3.0	1.0	−3.0
Body fitted	0.9823	−3.0179	1.0017	−3.0146
Embedded (Nitsche)	0.8870	−3.0180	1.0011	−3.0191
Embedded (Ausas)	0.8771	−3.0185	1.0000	−3.0053

the wall viscous effects are negligible. This is an assumable simplification in convection dominated flows which is to say, in high Re number flows. To this end, the flow around a vertical plate problem is solved using the presented discontinuous embedded formulation. The performance of the obtained solution is analysed by comparing with the body-fitted solution.

Table 10
Inviscid incompressible 2D flow around a cylinder C_p. Peak values relative errors.

Formulation	C_p rel. err. [%]			
	$\theta = 0$	$\theta = \frac{\pi}{2}$	$\theta = \pi$	$\theta = \frac{3\pi}{2}$
Body fitted	1.77	0.60	0.17	0.49
Embedded (Nitsche)	11.30	0.60	0.11	0.64
Embedded (Ausas)	12.29	0.62	0.0	0.18



(a) Body fitted (b) Embedded (Nitsche)

(c) Embedded (Ausas)

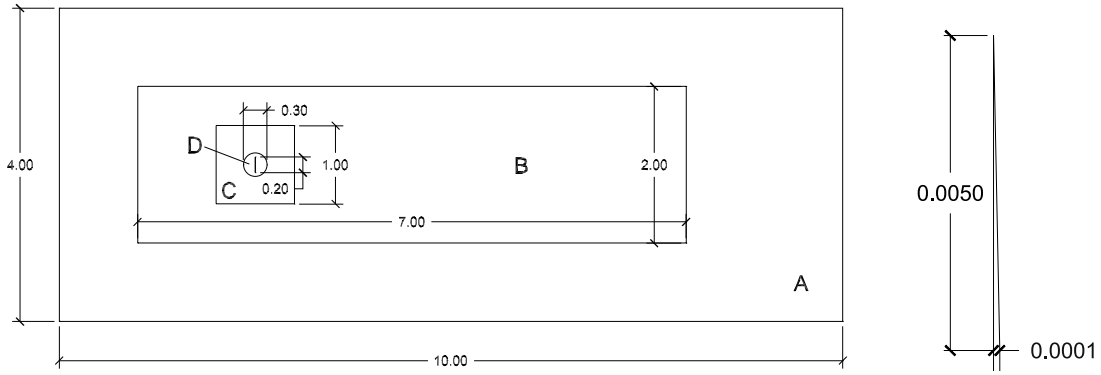
Fig. 21. Inviscid incompressible 2D flow around a cylinder. Velocity field contour lines.

In this case, the channel geometry and boundary conditions are the same as in the previous example. Concerning the immersed object, a vertical plate, which has a height and centre as 0.2 m and (0.0,0.0) m respectively, is placed instead of the cylinder. Fig. 22a depicts the problem geometry together with the auxiliary mesh regions.

It is worth mentioning that the plate has an almost zero thickness of 10^{-4}m in the body fitted case. If a zero-thickness plate is considered, the nodal normals in both plate tips are zero. This results in a zero-column in the rotation matrix (Eq. (17)) that causes the body-fitted slip formulation to fail. Furthermore, the body fitted plate tips are modified to be sharp ended so that the effect onto the fluid flow properly represents a thin plate (Fig. 22b).

On the other hand, no special treatment for the plate tips were done in the embedded case. Even though this induces a non-physical jump, it has been proven in [23] that these discontinuities are negligible, meaning that no special procedure is required in the elements containing the plate endpoints.

Taking the plate height as reference length, the fluid properties and flow conditions are set such that the Re number is 10^6Re. Therefore, the density ρ is 1 kg/m^3, the dynamic viscosity μ is 2e−7 kg/ms and a constant inlet value u_x of 1 m/s is imposed in the left edge of the domain. Concerning the time discretization, the BDF2 scheme is used again with a time step Δt equal to 10^{-2} s and a total simulation time of 20 s. The bad intersections correction that has been explained in the previous subsection is applied as well.



(a) Geometry and mesh regions. (b) Sharp end.

Fig. 22. 2D flow around a vertical plate. Problem set-up.

Table 11
2D flow around vertical plate mesh settings.

Region	A	B	C	D
Mesh size (m)	0.05	0.025	0.01	0.0025

With regard to the space discretization, Table 11 shows the mesh size in each one of the meshing subdivisions depicted in Fig. 22a.

The pressure coefficient (Eq. (30)) is the main comparison magnitude. However, since this problem has a periodic solution, a time averaged C_p is used instead. Thus, the time averaged C_p values considering the results from simulation time 10 s to 20 s are used for the results assessment.

Fig. 23 presents the evolution of the time averaged C_p values along the plate. First of all, it has to be said that the disparity of the results in both plate ends is associated to the plate thickness discrepancies. Note that the plate has the ideal zero thickness in the embedded case but it is modelled considering an infinitesimal thickness in the body fitted one. Preliminary tests showed that such disparities are even larger if a rectangular body-fitted plate (without the modified sharp tips) is analysed.

Disregarding the plate edges disparities, the embedded formulation results have quite good agreement with the reference body fitted ones in the upstream side of the plate. The C_p value in the stagnation point ($y = 0$ m) is 0.996 for the body fitted case and 0.978 for the discontinuous embedded formulation. Considering that the pressure coefficient in the stagnation point is expected to be equal to 1.0, a relative error of 0.4% and 2.2% is obtained.

Concerning the results in the downstream side, a similar trend but with differences in the obtained values is observed. Having a look onto the downstream C_p peak values, the body fitted one is -4.201 meanwhile the discontinuous embedded one is -4.060, representing a relative difference of 3.36%.

Besides that, it is interesting to comment on the staggered distribution of the discontinuous embedded solution. Once again, this behaviour is explained by the worse interpolation properties of the Ausas FE space, which always yields a continuous interpolation in one of the element subdivisions.

As in the previous example, Fig. 24 shows the body fitted and embedded velocity contour lines for a period of oscillation. Comparing both velocity distributions, it can be said that both solutions are in good agreement in the plate surroundings. However, the aforementioned poorer interpolation properties of the Ausas FE space are noted in the downstream side of the plate, where a sharp staggered velocity distribution is observed. These results are in accordance with the results that have been discussed in the C_p distribution.

3.6. 2D divergent channel with moving cylinder

This example is intentionally set to show two of the main potential applications of the presented formulations. On one hand, it is intended to prove the capabilities of embedded formulations in dealing with large moving boundaries without remeshing. On the other hand, it is also good to assess how the pseudo-compressibility term, added to the

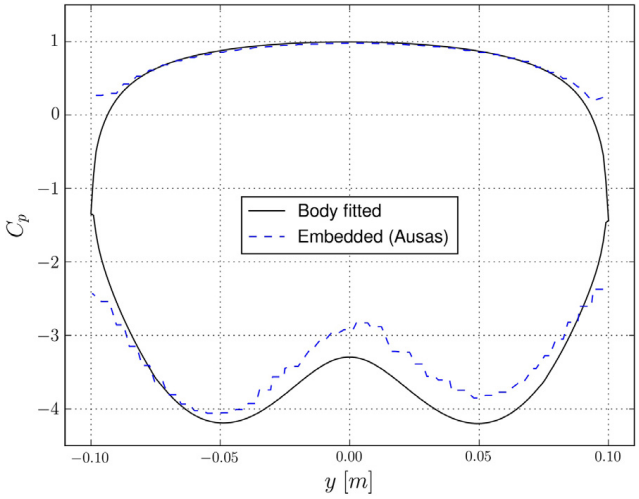


Fig. 23. 2D flow around a vertical plate. C_p time averaged values.

mass conservation equation, can alleviate the convergence problems in those cases where the pressure tends to blow up.

Thus, the problem geometry (Fig. 25) consists of a 2D divergent channel together with a moving cylinder of radius 0.1 m immersed in the divergent region. This problem tries to reproduce, in an extremely simplified manner, a valve which regulates the flow by opening according to the pressure value.

Complementary, Eq. (32) describes the horizontal displacement of the cylinder centre point for a given period of oscillation T. In this case, the period is set to 5 s. Assuming that the cylinder origin is placed in (x_{cyl},0)m coordinates, the x-component of the cylinder centre point can be computed as

$$x_{cyl} = 0.15 + 0.5(1 - abs(sin(\frac{2.0\pi t}{T}))) \tag{32}$$

yielding a maximum displacement of 0.5 m.

For the sake of simplicity, the cylinder movement is considered to be quasi-static. This means that the cylinder skin velocity is assumed to be null and is not included in the fluid problem.

The fluid density ρ is unitary and the dynamic viscosity μ is 10^{-3} kg/ms. Besides, the pseudo-compressibility constant c (speed of sound velocity) is reduced to 10^6 m/s. Such choice is enough to guarantee that the problem remains well defined even though isolated fluid cavities appear. A constant unit inlet is imposed at the left edge, slip boundary conditions are set at the channel top and bottom walls and the pressure is fixed to zero at the right edge.

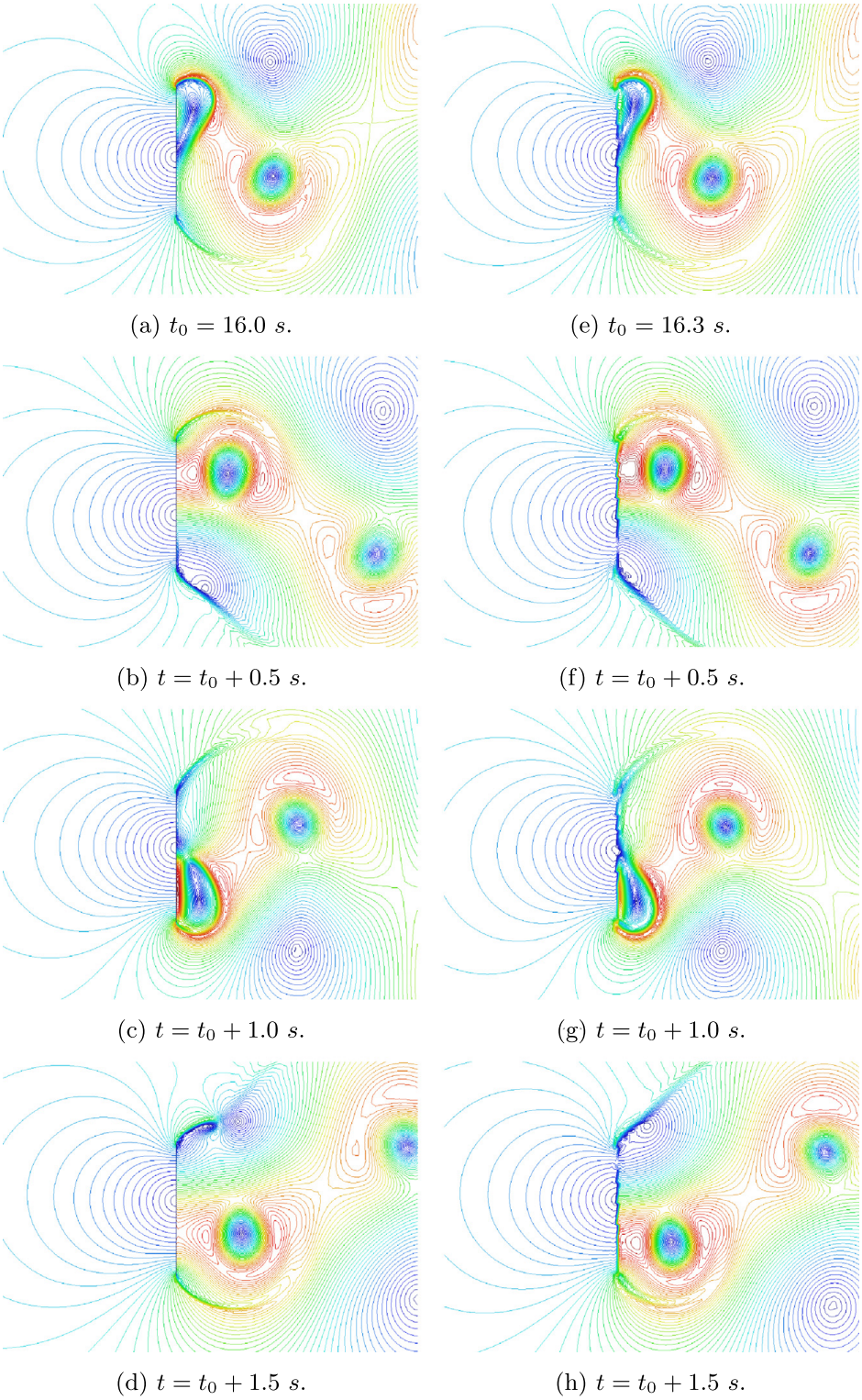
The BDF2 time scheme is used again with a time step of 0.1 s. The mesh employs approximately 60 k elements featuring an average size of 5e−3 m.

Finally, the same correction to bad intersections explained in the previous examples is used. However the distance threshold is increased to 1% the element size due to the larger possibility of having poor intersections owing to the large boundary movement.

In the next lines, the solutions obtained with both embedded formulations discussed in this work are compared.

First of all, it is important to mention that both embedded slip formulations discussed in this paper are capable of solving the posed problem. Taking into account the fact that solving this test case using a body fitted formulation might require remeshing due to the extremely large displacement, the utility of embedded techniques in the resolution of this kind of problems is proven.

If the solution between the proposed discontinuous embedded formulation is compared to the Nitsche one, no differences can be observed. Figs. 26 and 27 collect such comparison for the velocity and pressure fields for a quarter of a period of oscillation. As it can be observed in these sequence of snapshots, the pressure in the inlet channel becomes larger as long as the cylinder approximates to it. Similar behaviour can be noted in the velocity field, which peak values appear in the cylinder surroundings in order to keep the flow rate.



(a) $t_0 = 16.0\ s.$ (e) $t_0 = 16.3\ s.$

(b) $t = t_0 + 0.5\ s.$ (f) $t = t_0 + 0.5\ s.$

(c) $t = t_0 + 1.0\ s.$ (g) $t = t_0 + 1.0\ s.$

(d) $t = t_0 + 1.5\ s.$ (h) $t = t_0 + 1.5\ s.$

Fig. 24. 2D flow around a vertical plate. Body fitted (left) and discontinuous embedded (right) formulations velocity field contour lines (results are shown for a period of oscillation in each case).

3.7. 3D elbow with internal wall

This example is a 3D extension to the previously presented one and is aimed to prove that the presented formulation also works in 3D. For that purpose, the problem is solved using the presented discontinuous formulation

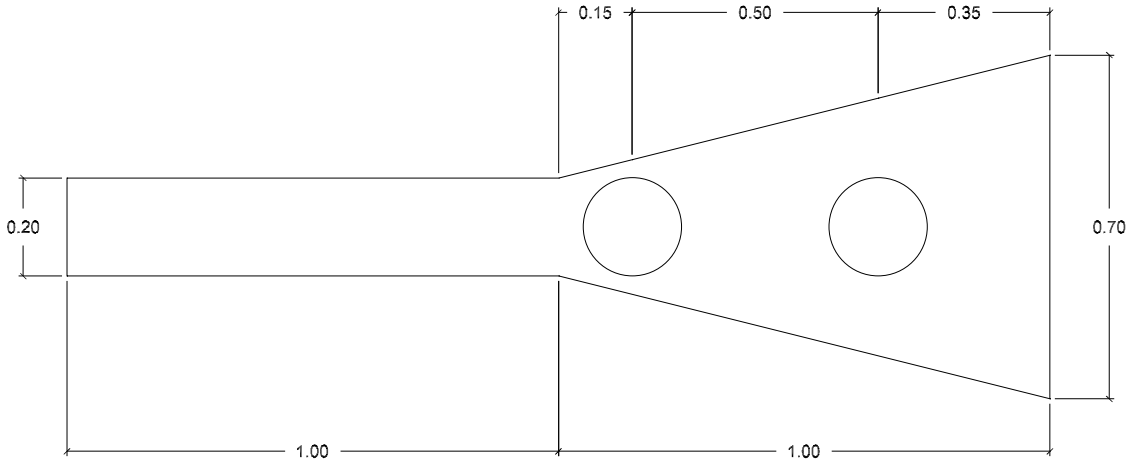
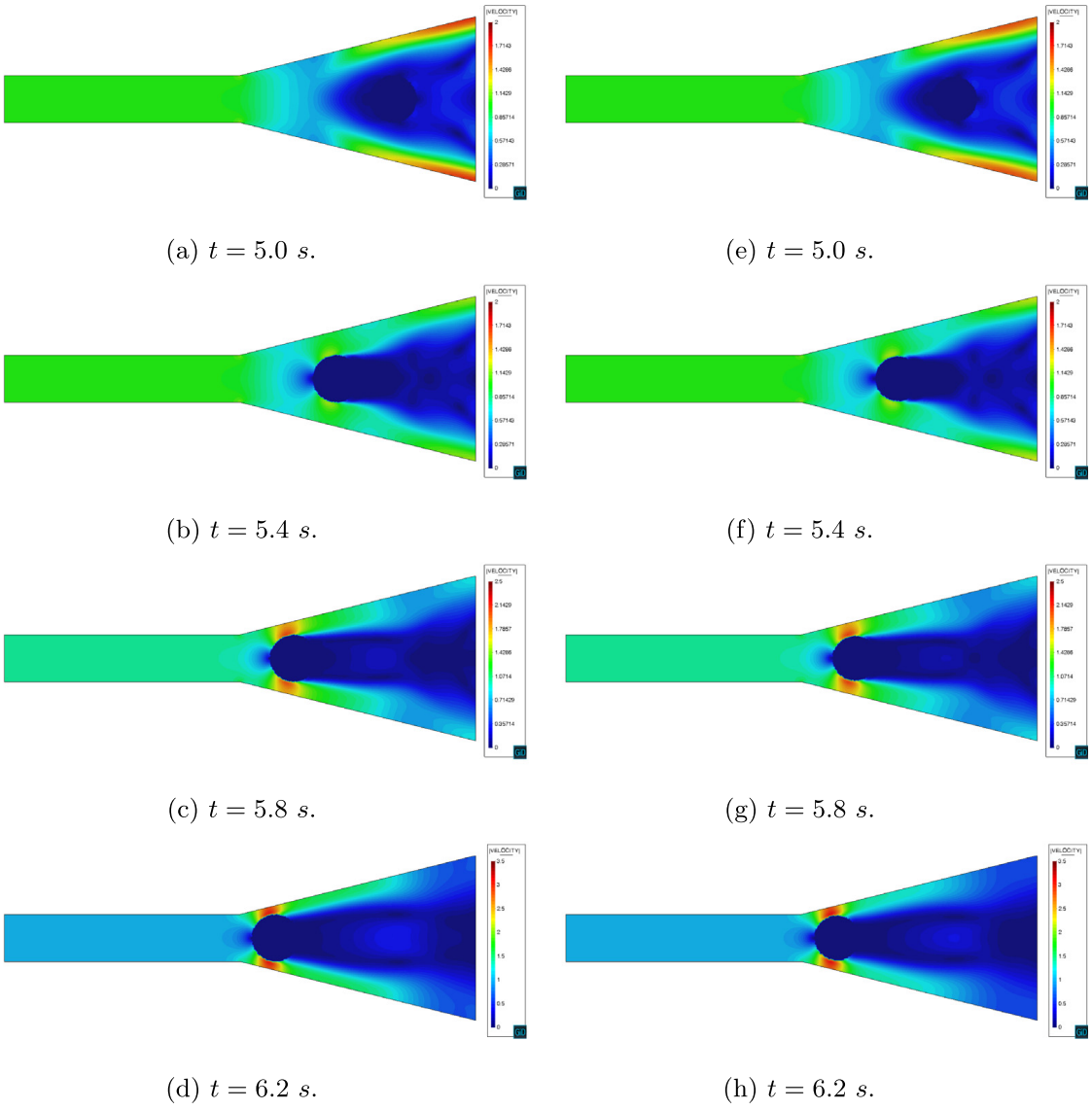


Fig. 25. 2D divergent channel with moving cylinder problem geometry. The maximum and minimum displacement positions of the embedded cylinder are shown as well.



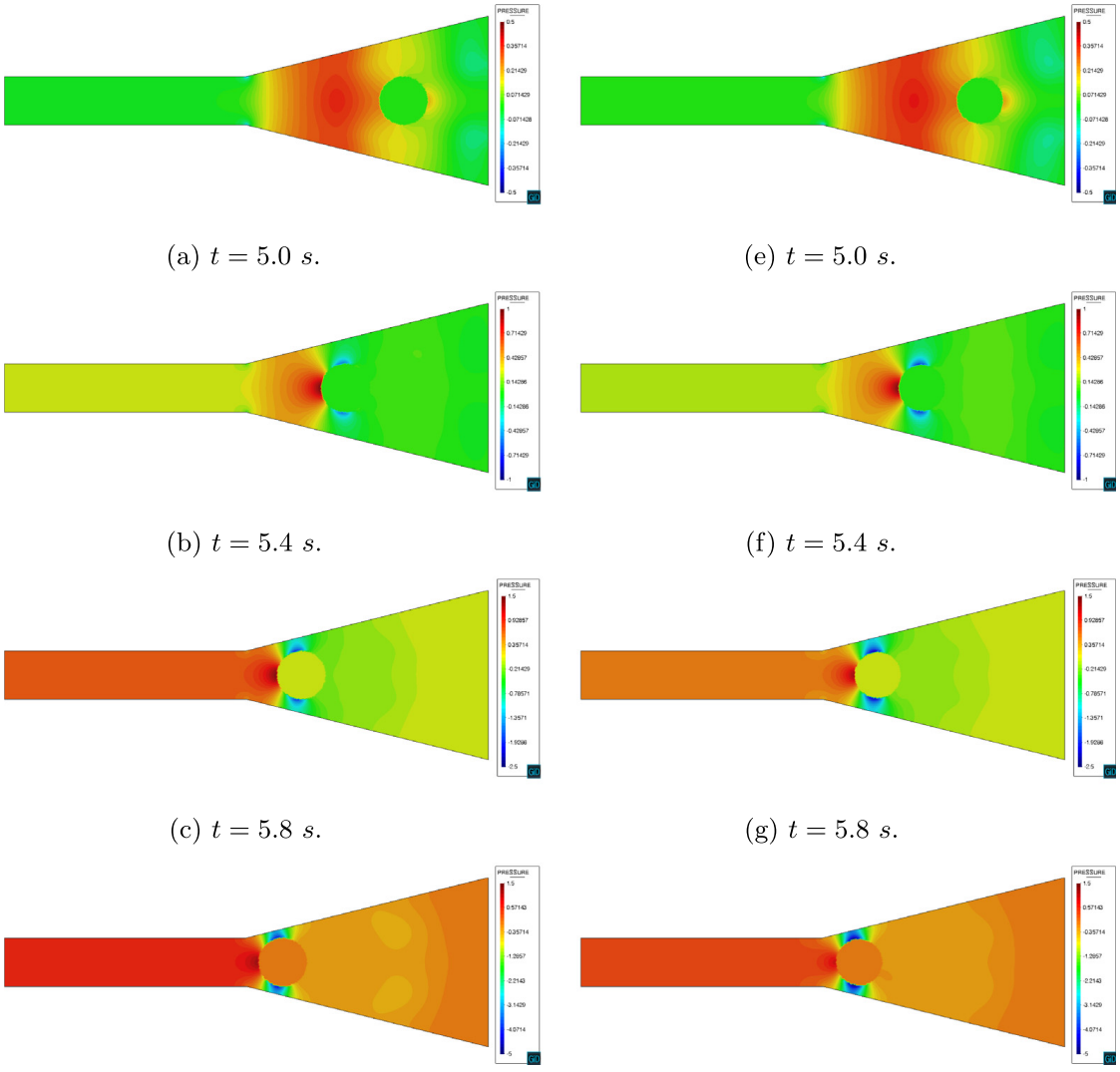
(a) $t = 5.0\ s.$ (e) $t = 5.0\ s.$

(b) $t = 5.4\ s.$ (f) $t = 5.4\ s.$

(c) $t = 5.8\ s.$ (g) $t = 5.8\ s.$

(d) $t = 6.2\ s.$ (h) $t = 6.2\ s.$

Fig. 26. 2D divergent channel with moving cylinder. Embedded discontinuous (left) and continuous (right) formulations velocity field (m/s) for a quarter of a period of oscillation. Note that the contour legends are scaled between plots for a better visualization.



(a) $t = 5.0\ s.$ (e) $t = 5.0\ s.$

(b) $t = 5.4\ s.$ (f) $t = 5.4\ s.$

(c) $t = 5.8\ s.$ (g) $t = 5.8\ s.$

(d) $t = 6.2\ s.$ (h) $t = 6.2\ s.$

Fig. 27. 2D divergent channel with moving cylinder. Embedded discontinuous (left) and continuous (right) formulations pressure field (Pa) for a quarter of a period of oscillation. Note that the contour legends are scaled between plots for a better visualization.

as well as the Nitsche one. Note that the Nitsche formulation is not capable of handling immersed objects without internal volume such as the membrane separating the two ducts. As a consequence, two different Nitsche cases are solved: one for each one of the ducts. This can be easily done by switching the distance function sign. As reference solution for the comparison, the same case is solved using the body fitted formulation.

With regard to the geometry, the one described in Fig. 15 is extruded to have a unit thickness value. To preserve the 2D flow pattern, a symmetry boundary condition is imposed at both sides of the 3D domain. Moreover, the other boundary conditions remain as in the original 2D case. Concerning the rest of the problem settings, the original 2D ones are used again.

For the sake of computational effort, the 3D equivalent mesh to the medium refinement level described in Table 6 is used. Thus, all meshes are conformed by equal order pressure–velocity linear tetrahedra.

The main comparison magnitude is the outlet velocity (z-component in the 3D case). Even though there are no 3D reference results, the analytical values in [39] have been taken as reference solution since the flow is expected to behave as in the 2D case because of the symmetry conditions.

Table 12 shows the midplane vertical velocity component values for the three described formulations. The results show good agreement with the expected results for all the studied cases. Similar relative errors can be observed,

Table 12
3D elbow with internal wall. Midplane maximum vertical velocity in the outlet [m/s] formulations comparison.

Expected	$\lvert v_y \rvert_{max}^{left}$ 1.5	rel. err. –	$\lvert v_y \rvert_{max}^{right}$ 3.0	rel. err. –
Body fitted	1.446	3.60	2.9081	3.06
Embedded (Nitsche — left)	1.4368	4.21	–	–
Embedded (Nitsche — right)	–	–	2.8936	3.55
Embedded (Ausas)	1.4851	0.99	2.9007	3.31

independently of the formulation used, indicating that the new formulation is performing on par with well established approaches.

The obtained velocity and pressure fields are compared in Figs. 28 and 29. Recall that in the Nitsche formulation case, each one of the ducts needs to be computed separately. As it can be observed, the obtained flow distribution is in good agreement with the 2D one (Fig. 16a) and no differences can be observed between the different solutions. Regarding the pressure field, the body fitted (Fig. 29a) and Nitsche (Figs. 29c and 29d) formulations have a remarkable similarity. In general terms, this affirmation can be extended to the discontinuous embedded formulation (Fig. 29b) since only minor perturbations, which are associated to the intersected elements modified FE space, can be observed in the elbow curvature region.

3.8. 3D flow around two boat sails

The last example shows the application of the new method to a complex double sided geometry. The model chosen here describes a two boat sails setting (Fig. 30). No reference solution of any type is available for the problem at hand so that the example simply represents a proof-of-concept application of the proposed technique in a volumeless scenario. Here the sail geometry is imported directly from a .stl file, and the solution is performed without any preprocessing step thus proving our point on the robustness of the proposed approach.

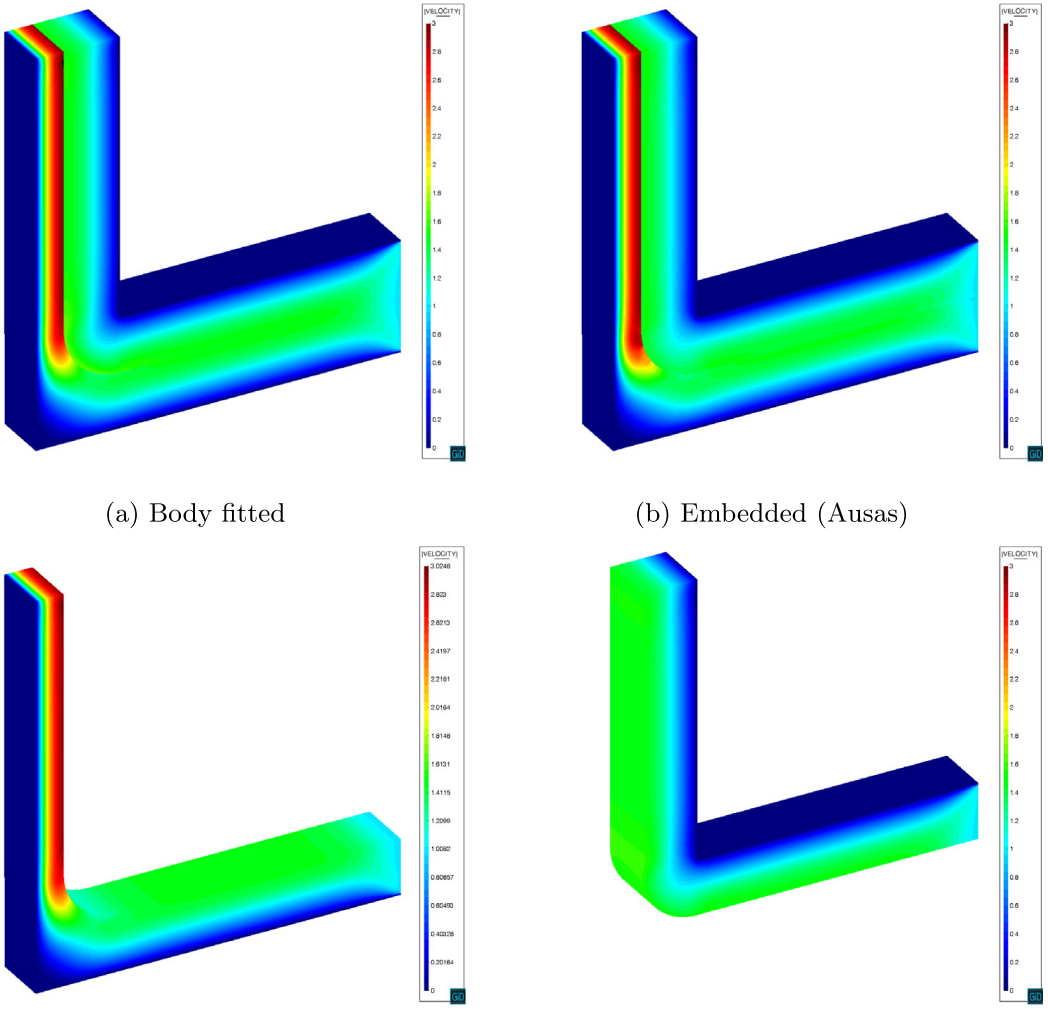
The two sails, which height is approximately 4 m, are placed inside a $7 \times 9 \times 16$ m channel. A constant inlet velocity u_x of 3.6 m/s (7 knots) is set. The pressure is fixed at the outlet and a slip boundary condition is set in the other channel boundaries. The BDF2 time discretization scheme is used with a time step Δt of 0.1 s and a total simulation time of 20 s. Concerning the space discretization, a mesh conformed by approximately 4.8 M linear tetrahedral elements is used.

Unlike in the previous examples, where the distance could be computed analytically, a robust distance calculation algorithm is required [35]. Fig. 31 depicts the reconstructed geometry, which matches the input shape quite well except at the sails boundary, where a saw-toothed shape is obtained as a consequence of the employed distance algorithm, which cannot represent "partial" intersections, in other words, the elements that are not completely intersected by the sails skin.

Despite the lack of reference results makes a comparison of the obtained solution with alternative formulations impossible, the results appear to be convincing. The formulation is capable of representing the discontinuities between the windward and leeward sides within one element. A positive overpressure appears in the windward region of the sails (Fig. 32b) while suction appears in the leeward one (Fig. 32a) without any sign of numerical overshoots or undershoots. As it is expected, the obtained velocity field (Fig. 33) is tangent to the sails surface as corresponds to the imposition of the slip BC (Fig. 34).

4. Conclusion

The aim of this work is to study the imposition of the slip boundary condition in the viscous incompressible Navier–Stokes equations in combination with boundary non-conforming mesh discretizations. This objective is achieved on the one hand, by discussing the imposition of the slip boundary condition in body fitted formulations by means of an MPC fashion technique. On the other hand, two different formulations for the embedded boundary case were studied, one based on a Nitsche imposition and another one based on the use of the modified Ausas FE space.



(a) Body fitted

(b) Embedded (Ausas)

(c) Embedded (Nitsche - left)

(d) Embedded (Nitsche - right)

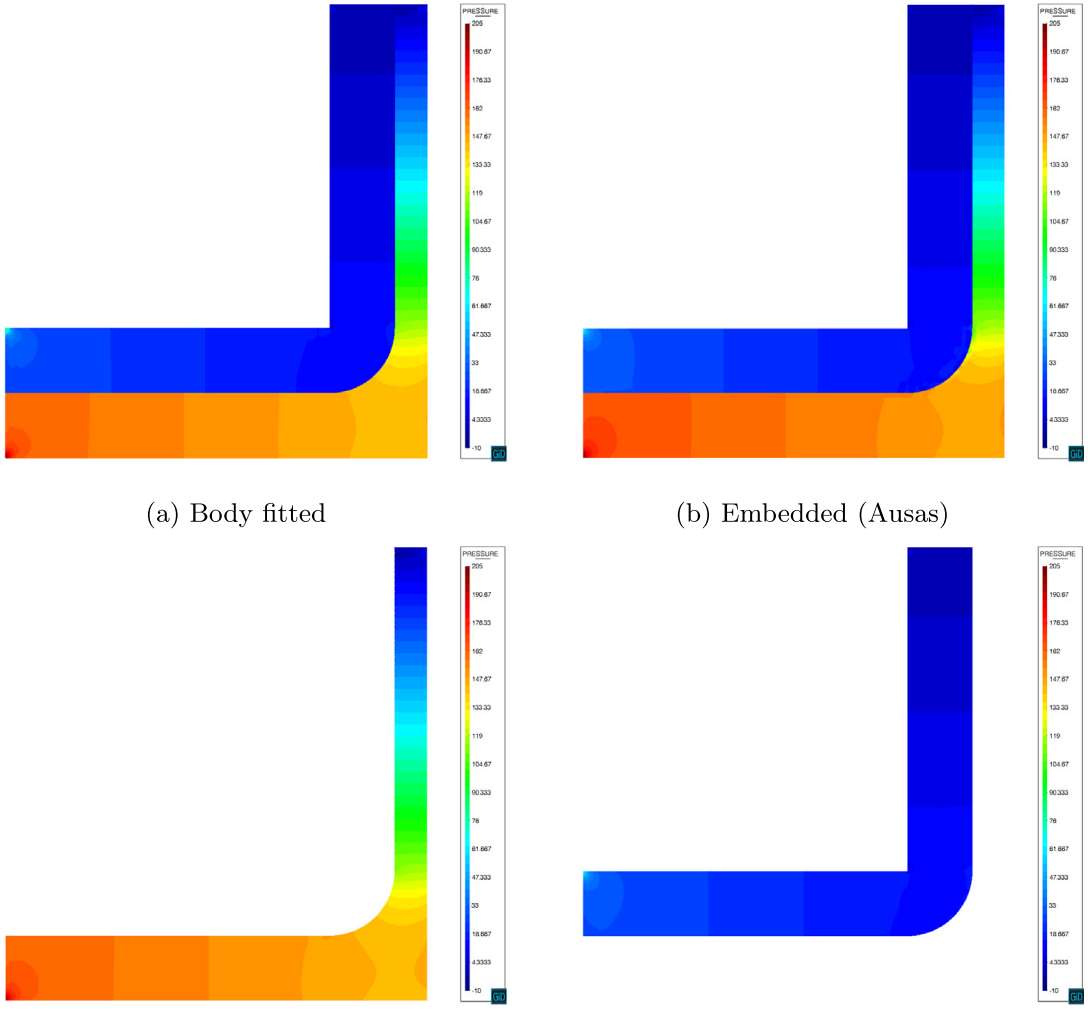
Fig. 28. 3D elbow with internal wall. Velocity modulus (m/s) comparison.

The former one, which relies on the imposition of a Navier-Slip boundary condition using a stabilized Nitsche method, is intended to be used in those cases in which the immersed object has a well defined internal volume, meaning that its skin can be described with a continuous distance function. The latter, which is proposed by the authors, modifies the FE space in the intersected elements to use the aforementioned Ausas discontinuous shape functions set. The convergence of the presented method is confirmed as well as its capability of representing bodies without internal volume. This interesting feature, which makes possible to solve CFD analyses of slender structures, is successfully proved in the simulation of the 2D/3D elbow with internal wall, the 2D flow around a vertical plate and the flow around two boat sails cases.

Apart from that, it is also important to stress the fact that cases with high Re number have been solved (Re $= \infty$ and Re $= 10^6$), obtaining a good agreement with the expected results. These results are crucial for the extension of the proposed technique to real application cases, where the viscous-slip approach can be used instead of the boundary layer mesh plus no-slip condition combination.

Regarding the accuracy of the obtained results, the body fitted solution is in general terms always better than the embedded ones. If the two studied embedded approaches are compared, both methods have a similar level of accuracy, being the Nitsche method slightly more precise in some particular cases. This is perfectly expectable according to the worse interpolation properties of the modified Ausas FE space.

Besides that, the capability of the discussed embedded formulations of solving large boundary movements problems is proven with the resolution of the 2D divergent channel with moving cylinder example. Furthermore, it



(a) Body fitted

(b) Embedded (Ausas)

(c) Embedded (Nitsche - left)

(d) Embedded (Nitsche - right)

Fig. 29. 3D elbow with internal wall. Midplane pressure (Pa) comparison.

has to be stressed that the addition of the pseudo-compressibility term in the mass conservation equation prevents the pressure to blow up in those cases where the pressure is only defined up to a constant. This feature is exploited here to solve a problem where the flow is close to be interrupted. In addition, this capability would make possible to solve isolated fluid cavities, likely associated to bad intersections coming from complex distance fields.

To sum up, it can be said that the correctness of all the implementations is proven. Regarding the discussed embedded slip formulations, for those bodies with well-defined internal volume, the optimal choice is concluded to be the stabilized Nitsche formulation, already published in [17], due to its slightly better accuracy and performance. However, the discontinuous embedded formulation presented by the authors arises as a feasible alternative for the analysis of bodies without internal volume, which could not be represented otherwise.

Finally, it is interesting to pinpoint the further capabilities that are still pending to be explored, such as the extension to more complex large moving boundary problems, which would require a robust initialization algorithm to be applied when topology changes occur, or the Fluid–Structure Interaction analysis of membrane and shell structures. Furthermore, it could also be interesting to investigate alternatives (e.g. Nitsche) to enhance the interface boundary condition imposition in the presented discontinuous embedded formulation.

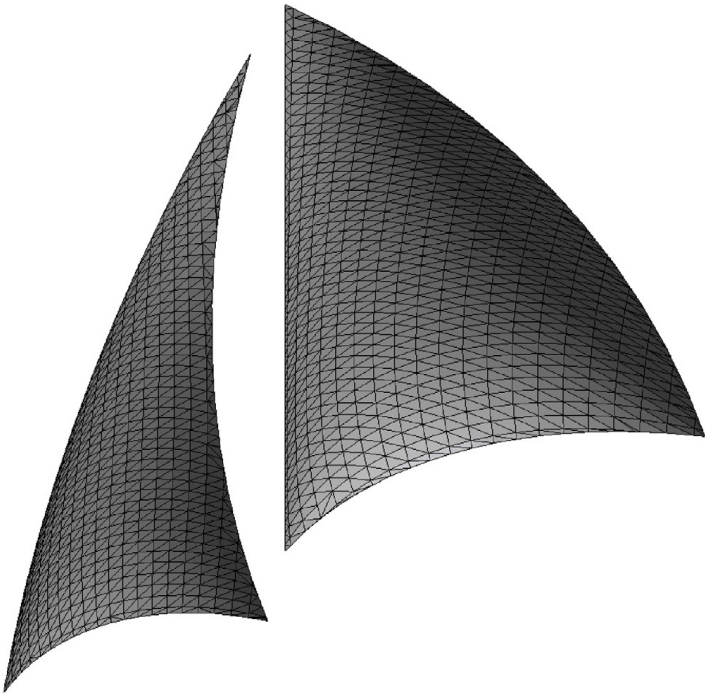


Fig. 30. 3D flow around two boat sails. .stl input geometry.

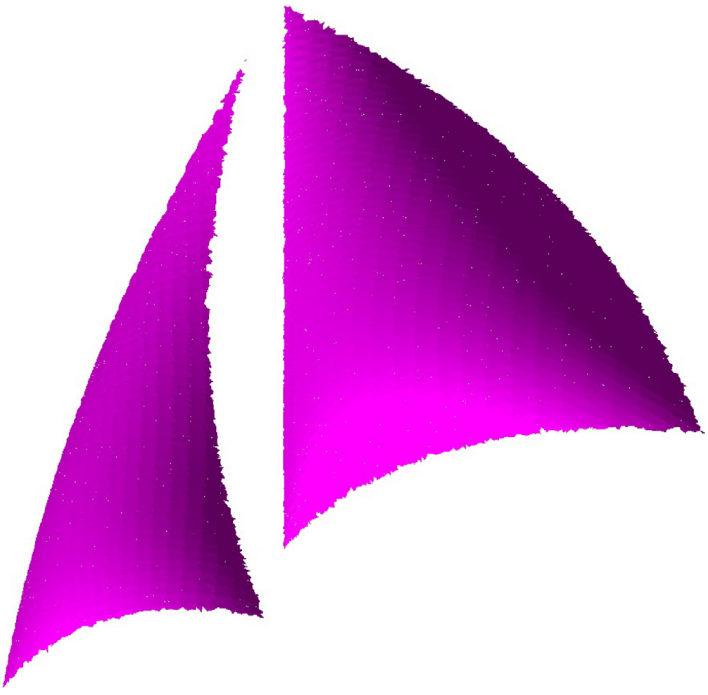
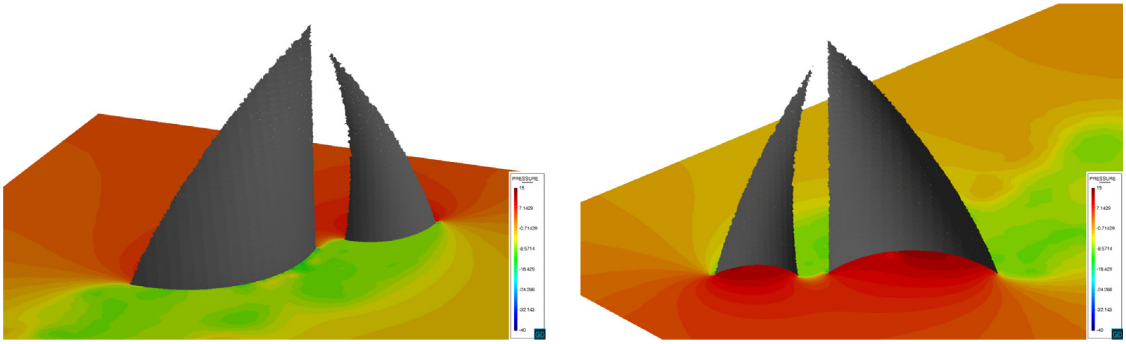


Fig. 31. 3D flow around two boat sails. Element intersections from discontinuous distance field.

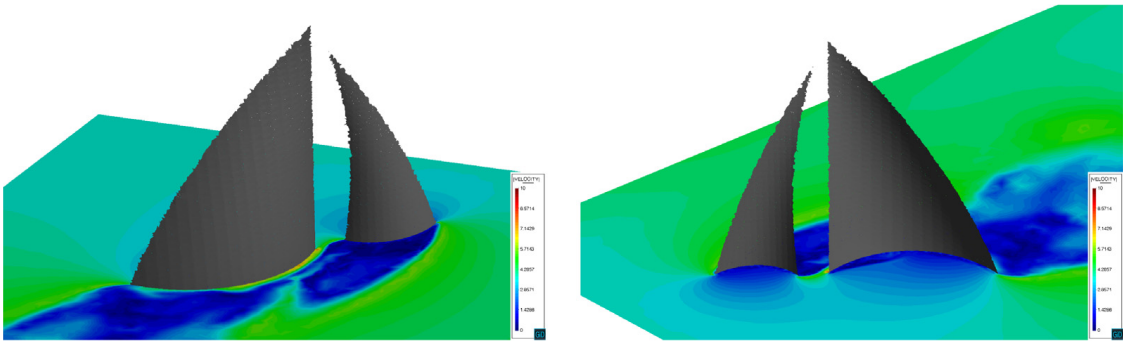
Acknowledgements

The research was partly supported by the International Graduate School of Science and Engineering (IGSSE) through the project ATMOPACE and the European Commission (EC) through the project ExaQUte (H2020-FETHPC-2016-2017-800898). The Spanish Ministry of Economy and Competitiveness (Ministerio de Economia y Competitividad, MINECO) through the projects HIRMA (RTC-2016-4967-5) and PRECISE (BIA2017-83805-R)



(a) Leeward view (b) Windward view

Fig. 32. 3D flow around two sails. Pressure field (Pa).



(a) Leeward view (b) Windward view

Fig. 33. 3D flow around two sails. Velocity field (m/s).

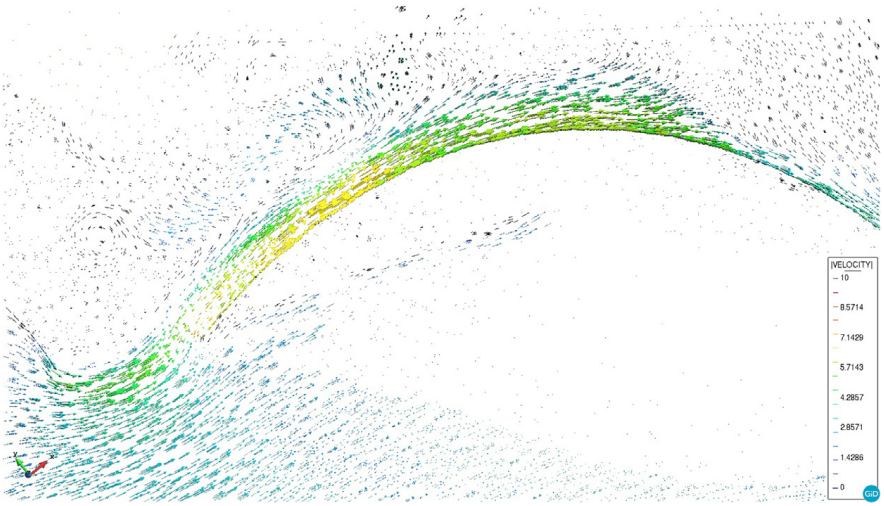


Fig. 34. 3D flow around two boat sails. Velocity vector field on horizontal cut.

is also greatly acknowledged. Rubén Zorrilla and Dr. Larese gratefully acknowledge the support of the Spanish and Italian ministries respectively for their FPU grant (FPU15/03796) and Rita Levi Montalcini fellowship (Programma per Giovani Ricercatori “Rita Levi Montalcini” - bando 2016).

References

[1] C. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252, http://dx.doi.org/10.1016/0021-9991(77)90100-0.

[2] C. Peskin, The immersed boundary method, Acta Numer. 112 (2002) 479–517, http://dx.doi.org/10.1017/S0962492902000077.

[3] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, first ed., Applied Mathematical Siences, vol. 153, Springer-Verlag New York, 2003.

[4] C. Hirt, A. Amsden, J. Cook, An arbitrary lagrangian-eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253, http://dx.doi.org/10.1016/0021-9991(74)90051-5.

[5] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian-Eulerian methods, in: Encyclopedia of Computational Mechanics, American Cancer Society, 2004, http://dx.doi.org/10.1002/0470091355.ecm009, Ch. 14. arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/0470091355.ecm009. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/0470091355.ecm009.

[6] J. Cottrell, T. Hughes, Y. Bazilevs, Isogeometric Analysis: Toward Integration of CAD and FEA, John Wiley & Sons, 2009.

[7] A. Coll, Robust Volume Mesh Generation for Non-watertigh Geometries (Ph.D. thesis), Universitat Politècnica de Catalunya, 2014.

[8] P. Frey, F. Alauzet, Anisotropic mesh adaptation for cfd computations, Comput. Methods Appl. Mech. Engrg. 194 (48) (2005) 5068–5082, http://dx.doi.org/10.1016/j.cma.2004.11.025.

[9] P. Benard, G. Balarac, V. Moureau, C. Dobrzynski, G. Lartigue, Y. D'Angelo, Mesh adaptation for large-eddy simulations in complex geometries, Internat. J. Numer. Methods Fluids 81 (12) (2015) 719–740, http://dx.doi.org/10.1002/fld.4204.

[10] F. Bertrand, P. Tanguy, F. Thibault, A three-dimensional fictitious domain method for incompressible fluid flow problems, Internat. J. Numer. Methods Fluids 25 (6) (1997) 719–736, http://dx.doi.org/10.1002/(SICI)1097-0363(19970930)25:6<719::AID-FLD585>3.0.CO;2-K.

[11] R. Löhner, J. Cebral, F. Camelli, S. Appanaboyina, J. Baum, E. Mestreau, O. Soto, Adaptive embedded and immersed unstructured grid techniques, Comput. Methods Appl. Mech. Engrg. 197 (25) (2008) 2173–2197, http://dx.doi.org/10.1016/j.cma.2007.09.010.

[12] J. Nitsche, Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind, Abh. Math. Semin. Univ. Hambg. 36 (1) (1971) 9–15, http://dx.doi.org/10.1007/BF02995904.

[13] R. Codina, J. Baiges, Approximate imposition of boundary conditions in immersed boundary methods, Internat. J. Numer. Methods Engrg. 80 (2009) 1379–1405, http://dx.doi.org/10.1002/nme.2662.

[14] A. Massing, M. Larson, A. Logg, M. Rognes, A stabilized nitsche fictitious domain method for the stokes problem, J. Sci. Comput. 61 (3) (2014) 604–628, http://dx.doi.org/10.1007/s10915-014-9838-9.

[15] A. Massing, B. Schott, W. Wall, A stabilized nitsche cut finite element method for the oseen problem, Comput. Methods Appl. Mech. Engrg. 328 (2018) 262–300, http://dx.doi.org/10.1016/j.cma.2017.09.003.

[16] J. Urquiza, A. Garon, M.-I. Farinas, Weak imposition of the slip boundary condition on curved boundaries for ?stokes flow, J. Comput. Phys. 256 (2014) 748–767, http://dx.doi.org/10.1016/j.jcp.2013.08.045.

[17] M. Winter, B. Schott, A. Massing, W. Wall, A nitsche cut finite element method for the oseen problem with general navier boundary conditions, Comput. Methods Appl. Mech. Engrg. 330 (2018) 220–252, http://dx.doi.org/10.1016/j.cma.2017.10.023.

[18] D. Kamensky, M.-C. Hsu, D. Schillinger, J. Evans, A. Aggarwal, Y. Bazilevs, M. Sacks, T. Hughes, An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves, Comput. Methods Appl. Mech. Engrg. 284 (2015) 1005–1053, http://dx.doi.org/10.1016/j.cma.2014.10.040.

[19] C. Peskin, Flow patterns around heart valves: A numerical method, J. Comput. Phys. 10 (2) (1972) 252–271, http://dx.doi.org/10.1016/0021-9991(72)90065-4.

[20] C. Peskin, D. McQueen, A three-dimensional computational method for blood flow in the heart i. immersed elastic fibers in a viscous incompressible fluid, J. Comput. Phys. 81 (2) (1989) 372–405, http://dx.doi.org/10.1016/0021-9991(89)90213-1.

[21] L. Zhu, C. Peskin, Interaction of two flapping filaments in a flowing soap film, Phys. Fluids 15 (7) (2003) 1954–1960, http://dx.doi.org/10.1063/1.1582476.

[22] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. 227 (10) (2008) 4825–4852, http://dx.doi.org/10.1016/j.jcp.2008.01.028.

[23] M. Davari, R. Rossi, P. Dadvand, Three embedded techniques for finite element heat flow problem with embedded discontinuities, Comput. Mech. 59 (2017) 1003–1030, http://dx.doi.org/10.1007/s00466-017-1382-7.

[24] J. Hoffman, J. Jansson, C. Johnson, New theory of flight, J. Math. Fluid Mech. 18 (2015) http://dx.doi.org/10.1007/s00021-015-0220-y.

[25] P. Dadvand, R. Rossi, E. Oñate, An object-oriented environment for developing finite element codes for multi-disciplinary applications, Arch. Comput. Methods Eng. 17 (3) (2010) 253–297, http://dx.doi.org/10.1007/s11831-010-9045-2.

[26] P. Dadvand, R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S. Idelsohn, E. Oñate, Migration of a generic multi-physics framework to HPC environments, Comput. & Fluids 80 (2013) 301–309, http://dx.doi.org/10.1016/j.compfluid.2012.02.004.

[27] J. Cotela, R. Rossi, E. Oñate, A fic-based stabilized finite element formulation for turbulent flows, Comput. Methods Appl. Mech. Engrg. 315 (2017) 607–631, http://dx.doi.org/10.1016/j.cma.2016.11.020.

[28] A. Brooks, T. Hughes, Streamline upwind/petrov-galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations, Comput. Methods Appl. Mech. Engrg. 32 (1) (1982) 199–259, http://dx.doi.org/10.1016/0045-7825(82)90071-8.

[29] T. Hughes, Multiscale phenomena: Greens function, the dirichlet to neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations, Comput. Methods Appl. Mech. Engrg. 127 (1) (1995) 387–401, http://dx.doi.org/10.1016/0045-7825(95)00844-9.

[30] T. Hughes, G. Feijóo, L. Mazzei, J. Quincy, The variational multiscale method - a paradigm for computational mechanics, Comput. Methods Appl. Mech. Engrg. 166 (1) (1998) 3–24, http://dx.doi.org/10.1016/S0045-7825(98)00079-6, advances in Stabilized Methods in Computational Mechanics.
[31] R. Codina, A stabilized finite element method for generalized stationary incompressible flows, Comput. Methods Appl. Mech. Engrg. 190 (20) (2001) 2681–2706, http://dx.doi.org/10.1016/S0045-7825(00)00260-7.
[32] R. Codina, Stabilized finite element approximation of transient incompressible flows using orthogonal subscales, Comput. Methods Appl. Mech. Engrg. 191 (39) (2002) 4295–4321, http://dx.doi.org/10.1016/S0045-7825(02)00337-7.
[33] R. Codina, J. Principe, O. Guasch, S. Badia, Time dependent subscales in the stabilized finite element approximation of incompressible flow problems, Comput. Methods Appl. Mech. Engrg. 196 (21) (2007) 2413–2430, http://dx.doi.org/10.1016/j.cma.2007.01.002.
[34] A. Meurer, C. Smith, M. Paprocki, O. Čertík, S. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. Moore, S. Singh, T. Rathnayake, S. Vig, B. Granger, R. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. Curry, A. Terrel, v. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, A. Scopatz, Sympy: symbolic computing in python, PeerJ Comput. Sci. 3 (2017) e103, http://dx.doi.org/10.7717/peerj-cs.103.
[35] D. Baumgärtner, J. Wolf, R. Rossi, P. Dadvand, R. Wüchner, A robust algorithm for implicit description of immersed geometries within a background mesh, Adv. Model. Simul. Eng. Sci. 5 (1) (2018) 21, http://dx.doi.org/10.1186/s40323-018-0113-8.
[36] C. Felippa, Introduction to Finite Element Methods, University of Colorado, USA, 2001.
[37] R. Ausas, F. Sousa, G. Buscaglia, An improved finite element space for discontinuous pressures, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1019–1031, http://dx.doi.org/10.1016/j.cma.2009.11.011.
[38] R. Ausas, F. Sousa, S. Idelsohn, A statically condensable enrichment for pressure discontinuities in two-phase flows, Mec. Comput. 30 (4) (2011) 175–191.
[39] S. Idelsohn, J. Gimenez, N. Nigro, Multifluid flows with weak and strong discontinuous interfaces using an elemental enriched space, Internat. J. Numer. Methods Fluids 86 (12) (2018) 750–769, http://dx.doi.org/10.1002/fld.4477.
[40] J. Anderson, Fundamentals of Aerodynamics, sixth ed., McGraw-Hill, 2017.

Chapter 5

Discontinuous Nitsche-based formulation for the imposition of the Navier-slip condition

5.1 Introduction

This chapter advances further in the investigation of embedded CFD methods for thin-walled bodies. Hence, taking as starting point the purely-slip formulation presented in Chapter 4, this chapter introduces some enhancements in order to extend the applicability range of the method to any wall behavior from the slip to the stick (no-slip) limits. This is achieved by substituting the slip BC by a general Navier-slip BC, which can be somehow understood as a linear wall law. In addition, the BC enforcement is also improved by substituting the penalty-based method by a Nitsche imposition.

The chapter is organized as follows. First of all the discontinuous Nitsche-based method for the Navier-slip BC imposition is presented. Secondly, the new proposal is validated by solving several examples, which involve different wall behaviors ranging from the slip to the no-slip limits. Moreover, a potential industrial application is also presented. The last section collects the conclusions and further enhancements.

The formulation in this chapter has been submitted for its publication in Computer Methods in Applied Mechanics and Engineering (CMAME). The current status of the paper is under review.

5.2 Embedded Navier-slip boundary condition

This section presents a novel formulation for the imposition of the Navier-slip BC over embedded thin-walled bodies. Hence, the substitution of the standard FE space by the Ausas discontinuous one is also introduced in order to represent the discontinuities arising from the immersion of a (possibly thin-walled) body. However, the space substitution is combined in this case with the Navier-slip Nitsche method presented in [110]. Although this might seem a simple mixture of the ideas in Chapter 4 with the ones in [110], it is not obvious that the combination of the Ausas FE space with a stick wall behavior works out of the box owing to the worse interpolation properties featured by the Ausas FE space. In the following, the properties of the Ausas FE space are briefly recalled to ease the comprehension of the rest of the chapter. The next subsections present the Nitsche method for the imposition of the Navier-slip BC and a more accurate alternative to calculate the drag force when the Ausas FE space is involved.

5.2.1 The Ausas Finite Element space

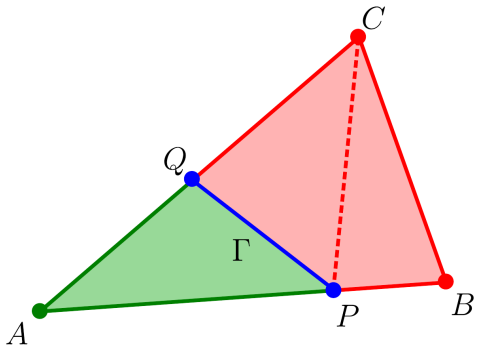
The Ausas FE space was initially conceived to represent the discontinuities arising from the resolution of two-phase flow problems [2, 3]. In this thesis, the ability of the Ausas FE space to disconnect the solution fields in the two sides of the level set intersection is exploited to directly embed thin-walled bodies without requiring any interface operation.

A particularly interesting feature of the Ausas FE space is that it is conforming with the standard FE one. Hence, no special treatment of the blending elements (the elements neighbouring the intersected ones) is required, thus leading to a purely local formulation. This becomes in a great advantage for the extension of the formulation to distributed memory platforms.

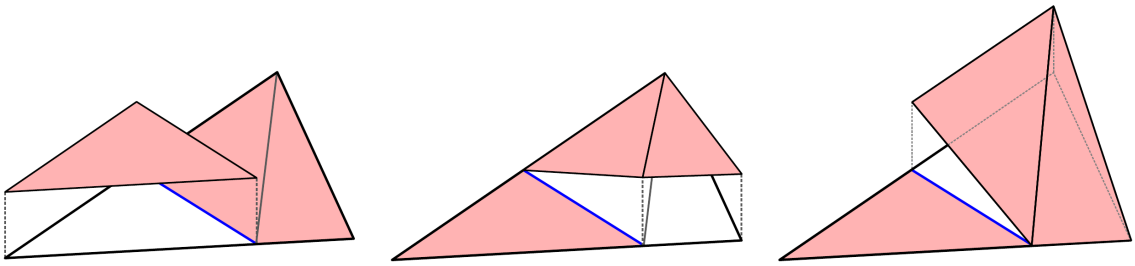
Aiming to describe the main geometrical features of the Ausas FE space, Fig. 5.1 depicts the same intersected element example that the original authors used in their proposal. By inspecting the Ausas shape functions that are obtained from the corresponding division in subelements, it can be observed that:

- the shape function values along an intersected edge are constant. Note that in the green side the shape function at the intersection points Q and P takes the same value as in A (Fig. 5.1b). On the other hand, in the red side the value at the intersection point P takes the same value as in B (Fig. 5.1c). The same can be observed for Q and C (Fig. 5.1d).

- in consequence, the shape functions values are constant in one of the split sides (the green one in Fig. 5.1a), and thus lead to an always constant interpolation with no gradient. Hence, only piecewise constant functions can be exactly reproduced.
- in the non-constant side, the shape function gradients are approximatively null in the intersection normal direction.



(a) Partition of ABC following the interface PQ.



(b) N_A (c) N_B (d) N_C

Figure 5.1: Ausas shape functions for the intersected finite element ABC (source [2]).

5.2.2 Nitsche imposition of the Navier-slip Boundary Condition

This subsection presents the general Navier-slip BC as well as the Nitsche method that is employed for its imposition. As it has been already mentioned, this technique is based on the work by Winter et. al. in [110], which is adapted to be used in combination with the aforementioned Ausas FE space.

In short, the Navier-slip BC can be understood as a sort of linear wall law whose behaviour is regulated by the slip length parameter ε as

$$\varepsilon \frac{\partial \mathbf{u}^t}{\partial x^n} = \mathbf{u}^t \tag{5.1}$$

where $\mathbf{u}^t$ and x^n denote the tangential wall velocity and the normal to the wall direction. As it can be observed, the Navier-slip BC approaches the completely stick condition as $\varepsilon \to 0$ while it tends to the slip limit as $\varepsilon \to \infty$.

From a numerical perspective, the general Navier-slip BC is defined as the conjunction of a no penetration constraint in the normal direction and a shear force imposition in the tangential one. While the first one is enforced on the normal projection of the velocity, the second is a boundary value imposition of the tangential projection of the viscous traction, meaning that the general Navier-slip condition is indeed a Robin-type BC. Accordingly, the Navier-slip BC can be split in a normal and a tangential contribution as

$$\mathbf{P}^n \left(\mathbf{u} - \mathbf{g}\right) = 0 \tag{5.2a}$$

$$\mathbf{P}^t \left(\varepsilon \left(\left[\mathbb{C} : \left(\boldsymbol{\nabla}^s \mathbf{u}\right)\right] \cdot \mathbf{n} - \mathbf{h}\right) + \mu \left(\mathbf{u} - \mathbf{g}\right)\right) = 0 \tag{5.2b}$$

in which $\mathbf{g}$ and $\mathbf{h}$ are the velocity and the tangential traction to be imposed over the embedded boundary. $\mathbf{P}^n$ and $\mathbf{P}^t$ are the normal and tangential projection operators, which can be computed as $\mathbf{P}^n = \mathbf{n} \otimes \mathbf{n}$ and $\mathbf{P}^t = \mathbf{I} - \mathbf{n} \otimes \mathbf{n}$.

The imposition of the BC in Eq. 5.2 is achieved by using a stabilized Nitsche method in both the normal and tangential directions. The Nitsche imposition of the normal component reads

$$\begin{aligned} \left\langle \frac{\mu}{\gamma h} \left(\mathbf{u}_h - \mathbf{g}\right) \mathbf{P}^n, \mathbf{w} \right\rangle_\Gamma - \left\langle \left(\mathbf{u}_h - \mathbf{g}\right) \mathbf{P}^n, \left(q\mathbf{I} + \zeta \mathbb{C} : \boldsymbol{\nabla}^s \mathbf{w}\right) \mathbf{n} \right\rangle_\Gamma + \\ \left\langle \frac{\phi_u}{\gamma h} \left(\mathbf{u}_h - \mathbf{g}\right) \mathbf{P}^n, \mathbf{w} \right\rangle_\Gamma \end{aligned} \tag{5.3}$$

while the tangential one is

$$\begin{aligned} &\left\langle \frac{1}{\varepsilon + \gamma h} \left(\varepsilon \left(\mathbb{C} : \boldsymbol{\nabla}^s \mathbf{u}_h \mathbf{n} - \mathbf{h}\right) + \mu \left(\mathbf{u}_h - \mathbf{g}\right)\right) \mathbf{P}^t, \mathbf{w} \right\rangle_\Gamma - \\ &\zeta \left\langle \frac{\gamma h}{\varepsilon + \gamma h} \left(\varepsilon \left(\left[\mathbb{C} : \left(\boldsymbol{\nabla}^s \mathbf{u}_h\right)\right] \cdot \mathbf{n} - \mathbf{h}\right) + \mu \left(\mathbf{u}_h - \mathbf{g}\right)\right) \mathbf{P}^t, \left(\boldsymbol{\nabla}^s \mathbf{w}\right) \cdot \mathbf{n} \right\rangle_\Gamma \end{aligned} \tag{5.4}$$

γ is the penalty constant while $\zeta \in \{-1, 1\}$. The choice $\zeta = -1$ leads to the adjoint inconsistent Nitsche formulation, which enjoys improved inf-sup stability for any value of γ [110]. Even though optimal convergence is not guaranteed for the velocity L^2-error in this case, the adjoint inconsistent formulation is taken owing to its reported better

stability properties. ϕ_u is a stabilization constant defined as

$$\phi_u = \mu + \rho\|\mathbf{u}\|h + \frac{\rho}{\Delta t}h^2 \tag{5.5}$$

It is important to remark that these two constraints need to be integrated in both sides of the level set interface after the ability of the Ausas FE space to completely disconnect the positive and negative sides of the intersection. Hence, the problem somehow translates to impose the same constraint in two different but overlapping interfaces.

In addition, it is also important to point out that the Ausas FE space reduces the approximation properties within the cut elements, so exactly where the Nitsche terms are active. This effect is particularly relevant since, depending on the splitting pattern, the gradient of the velocity and of the test functions is kinematically forced to be zero on one of the sides of the cut. In this context, the verification of the correct imposition of the Navier-slip BC is fundamental.

5.2.3 Drag calculation

Owing to the worse interpolation properties featured by the Ausas FE space, the boundary traction calculation might be inaccurate, specially under the presence of important wall shear effects. This section aims to address such undesired situation, which might limit the extension of the method to FSI problems.

In general terms, the computation of the embedded boundary traction can be performed by integrating the formula

$$\mathbf{t} = \mathbf{P}^n\left((\mathbb{C} : \boldsymbol{\nabla}^s\mathbf{u})\cdot\mathbf{n}\right) + \mathbf{P}^t\left((\mathbb{C} : \boldsymbol{\nabla}^s\mathbf{u})\cdot\mathbf{n}\right) - p\mathbf{n} \tag{5.6}$$

over the level set intersections. Alternatively, it can be assumed that the Navier-Slip condition requires $t_{\|} = -\frac{\mu}{\varepsilon}\mathbf{P}^t(\mathbf{u}-\mathbf{g})$, or equivalently that $\mathbf{P}^t\left((\mathbb{C} : \boldsymbol{\nabla}^s\mathbf{u})\cdot\mathbf{n}\right) \approx -\frac{\mu}{\varepsilon}\mathbf{P}^t(\mathbf{u}-\mathbf{g})$, and rewrite Eq. 5.6 in the form

$$\mathbf{t} \approx \mathbf{P}^n\left((\mathbb{C} : \boldsymbol{\nabla}^s\mathbf{u})\cdot\mathbf{n}\right) - \frac{\mu}{\varepsilon}(\mathbf{u}-\mathbf{g})\,\mathbf{P}^t - p\mathbf{n} \tag{5.7}$$

These two expressions are equivalent in the continuum and when a "standard" FE formulation is employed. Not so however in the case of using Ausas shape functions, since in presence of zero velocity gradients would predict a zero skin friction in the first while the second would correctly estimate a skin friction provided that the tangential

velocity is non-zero on the cut boundary (even when the gradient is zero in the immediate vicinity of the intersection).

In order to explain such inconsistency, the use of Ausas shape functions can be re-interpreted as the use of the standard FE space subjected to the additional constraint that the solution needs to be piecewise constant along each cut edge. The action of such geometric constraint is fundamental in the way equilibrium is established in the vicinity of the immersed objects. Indeed, the constraint allows forces to be transmitted from the cut boundary to the surrounding nodes without the need of producing a gradient in the solution, and is the reason for which the effective tangential stress can be non-zero in the vicinity of the embedded boundaries.

5.3 Validation

This section collects the benchmarking of the proposed formulation. To ensure that all the examples can be reproduced, the geometry, BCs and simulation settings are detailed prior to the results assessment.

In the following all the examples are listed with a brief description of their motivation

- 2D straight channel: this is a simple test case to check the implementation and convergence rates of the formulation considering three different wall behaviors (slip, intermediate and no-slip).
- 2D flow inside a ring: to obtain the convergence rates of the formulation when dealing with faceted approximations of curved geometries. The influence of the Nitsche constant γ is also evaluated with this example.
- 2D flow around pressurized cylinder: to prove the utility of the pseudo-compressible term to keep the pressure bounded when isolated fluid cavities appear.
- 2D elbow with internal wall: to test the performance of the formulation with a more complex geometry.
- 2D flow around thin-walled arch profile: to assess the performance of the proposed technique to solve high Re flows. This example is conceived as a 2D idealization of the final 3D example.
- 3D flow around a sailboat: this example presents a potential industrial application that consists in the CFD analysis of a sailboat.

5.3.1 2D straight channel

The objective of this very first example is to preliminary assess the correctness and convergence rates of the proposed formulation. Thus, the geometry and boundary conditions have been selected to be as simple as possible.

The geometry consists in a 2D 1.1x2m rectangular channel. The discontinuous level set function is set such that the effective width of the channel is 1m. Regarding the wall behaviour, the bottom edge is considered as a no-slip boundary while the embedded top one is tested with both slip and no-slip BCs. Moreover, an extra intermediate case in which the embedded skin ε equals 10^{-1}m is also solved.

The material properties are selected such that the Re number is 100. This is achieved by setting the density to 1kg/m^3 and the viscosity to and 10^{-2}kg/ms.

A second order Backward Differentiation Formula (BDF2) scheme is used for the time discretization. To ensure that the steady state solution is reached, 20 time steps of 100s are solved.

The problem is solved for a set of structured meshes, whose characteristic element size and edge subdivisions are collected in Table 5.1.

Table 5.1: 2D straight channel. Mesh refinement settings.

	Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5
Element size [m]	0.2	0.0667	0.0333	0.0016	0.0083	0.0042
Short edge divisions	5	15	30	65	120	241
Long edge divisions	10	30	60	130	240	480

Slip interface

In this case the flow is induced by applying a volume force of 0.5m/s^2 in the horizontal direction. The problem reaches a steady state that can be characterized by the rigid body movement

$$v_x = 50y\left(1 - \frac{y}{2}\right) \tag{5.8a}$$

$$v_y = 0 \tag{5.8b}$$

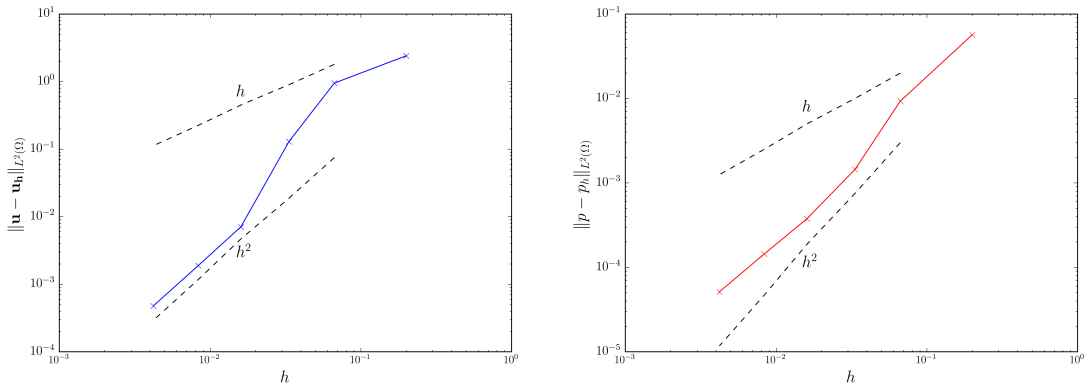
$$p = 0 \tag{5.8c}$$

Table 5.2 collects the $L^2(\Omega)$-norm of both the velocity and pressure errors. Fig. 5.2a and Fig. 5.2b depict the previous results. As it can be observed, the convergence order is h^2 for the velocity field while it is superlinear for the pressure one, meaning that the obtained convergence rates match the linear FE ones even though the Ausas FE space is limited to a $h^{3/2}$ convergence rate [2]. Such improved convergence is a particularity of the current test case, which is not affected by the null gradient limitation of the Ausas shape functions since the analytical solution gradient is zero in the top embedded boundary.

Finally, Fig. 5.3 presents the velocity field for three different mesh refinement levels (meshes 1, 3 and 5 in Table 5.1). The obtained solution matches the parabolic rigid body movement in Eq. 5.8a in all cases. It can be also observed that, as expected, the gradient tends to zero in the region close to the embedded boundary as the mesh is refined.

Table 5.2: 2D straight channel (slip interface). Velocity and pressure error norms.

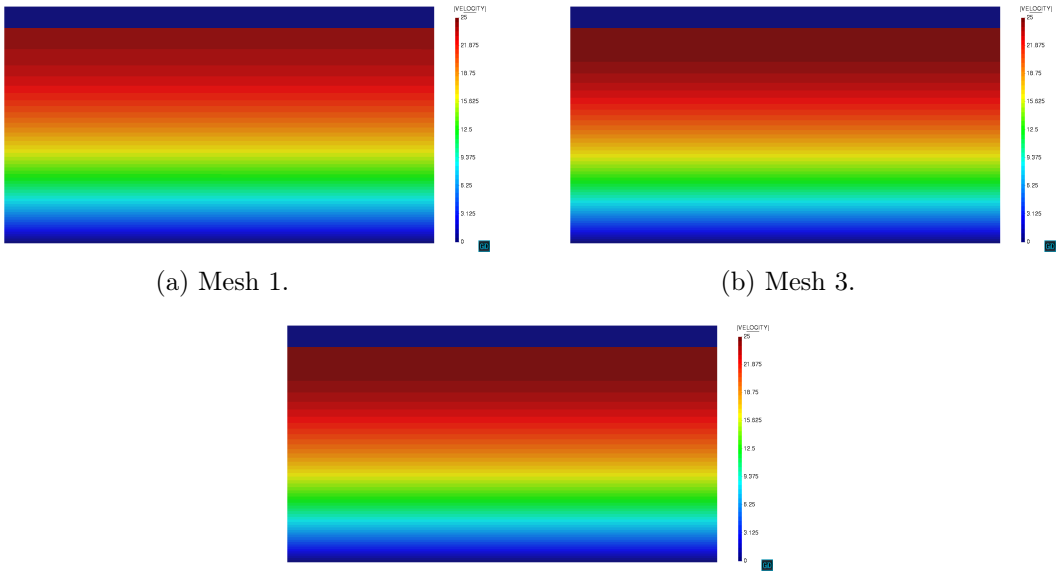
	Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5
$\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$	2.41253	0.950292	0.129366	0.007083	0.001888	0.000479
$\|p-p_h\|_{L^2(\Omega)}$	0.056811	0.009313	0.001444	0.000376	0.000144	0.000051



(a) Velocity convergence.

(b) Pressure convergence.

Figure 5.2: 2D straight channel (slip interface). Solid lines represent the obtained results. Dashed lines represent h and h^2 convergence rates.



(a) Mesh 1.

(b) Mesh 3.

(c) Mesh 5.

Figure 5.3: 2D straight channel (slip interface). Coarse, medium and fine meshes $\|\mathbf{u}\|$ field.

No-slip interface

Instead of using a body force to induce the flow, in this case a pressure gradient between the inlet and the outlet is applied. Hence, the pressure is fixed to 1Pa in the left edge and to 0Pa in the right one. This, in combination with the no-slip behaviour in both top and bottom boundaries, leads to the well known pressure driven flow between two steady plates benchmark, also known as Poiseuille flow. The steady state analytical solution of the Poiseuille flow can be particularised for the above mentioned material properties as

$$v_x = 25y\,(1-y) \tag{5.9a}$$

$$v_y = 0 \tag{5.9b}$$

$$p = \frac{2-x}{2} \tag{5.9c}$$

Table 5.3 collects the $L^2\,(\Omega)$-norm of the velocity and pressure errors. The results exhibit a convergence rate around h for both velocity and pressure fields (Figs. 5.4a and 5.4b).

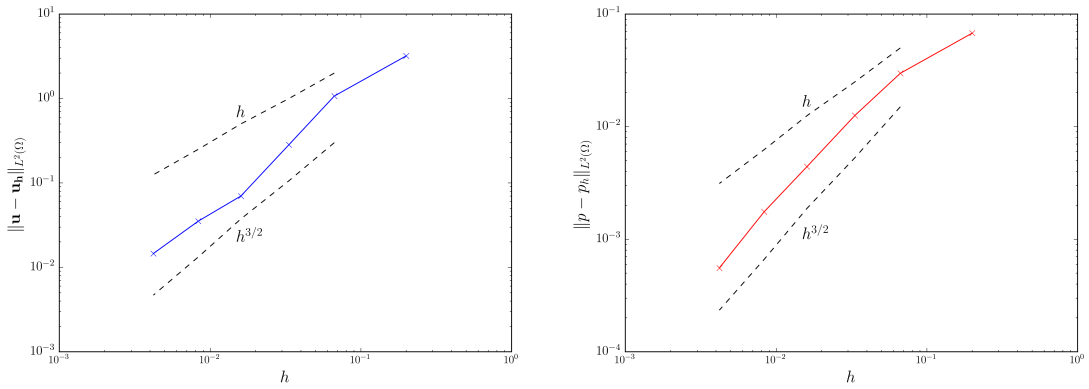
Fig. 5.5 depicts the obtained solution for three different mesh refinement levels (meshes 1, 3 and 5 in Table 5.1). The expected parabolic velocity profile can be noted in all cases.

However, the peak value in the channel mid axis is far from the analytical solution value (6.25m/s) in the coarsest mesh. Regarding the pressure, the horizontal linear gradient can be observed in all cases.

Furthermore, it is worth to comment about the perturbations in the solution close to the embedded boundary, which can be clearly noted in the coarsest mesh pressure field (Fig. 5.4b). These are the direct consequence of the inability of the Ausas FE space to represent a gradient in one side of the intersected elements, which forces the solution in the embedded boundary vicinity to accommodate according to such extra constraint.

Table 5.3: 2D straight channel (no-slip interface). Velocity and pressure error norms.

	Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5
$\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$	3.1982	1.06859	0.28365	0.07003	0.035238	0.014591
$\|p-p_h\|_{L^2(\Omega)}$	0.067731	0.02977	0.012579	0.004407	0.001754	0.000556

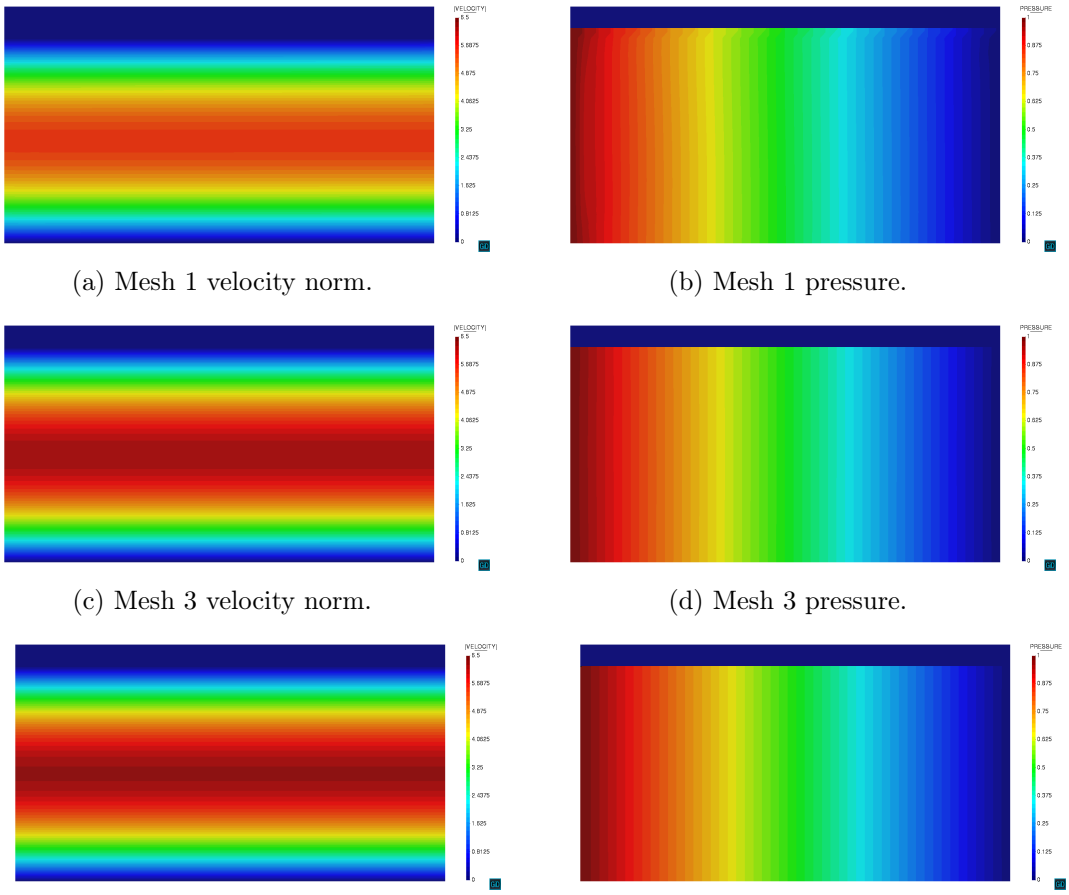


(a) Velocity convergence. (b) Pressure convergence.

Figure 5.4: 2D straight channel (no-slip interface). Solid lines represent the obtained results. Dashed lines represent h and $h^{3/2}$ convergence rates.

Slip length 10^{-1} interface

This last variant of the straight channel test checks the performance of the proposed technique when it is applied in a wall law fashion. This means to introduce a controlled wall friction such that the embedded skin behavior is between the slip and no-slip limits. Hence, the slip length is set ε to 10^{-1}m in the embedded boundary. The inlet profile



(a) Mesh 1 velocity norm.

(b) Mesh 1 pressure.

(c) Mesh 3 velocity norm.

(d) Mesh 3 pressure.

(e) Mesh 5 velocity norm.

(f) Mesh 5 pressure.

Figure 5.5: 2D straight channel (no-slip interface). Coarse, medium and fine meshes solution.

$v_x = 1 - (1/(1+\varepsilon))y$ is accordingly imposed in the left edge while the bottom edge velocity is fixed to [1,0]m/s.

These settings yield the analytical velocity and pressure steady state solution

$$v_x = 1 - \left(\frac{1}{1+\varepsilon}\right) y \tag{5.10a}$$

$$v_y = 0 \tag{5.10b}$$

$$p = 0 \tag{5.10c}$$

which exerts a 0.0182N total horizontal drag force over the embedded boundary.

Table 5.4 collects the $L^2(\Omega)$-norm of both the velocity and pressure errors as well as the $L^2(\Gamma)$-norm of the horizontal drag force error. These results are presented for

three different values of the penalty constant γ. As it can be observed in Fig. 5.6, the convergence rate of the three analysed magnitudes is in all cases around h, which is in line with the results previously obtained for the pure no-slip case. Besides that, it can be also noted that the convergence rate of the formulation is not affected by the γ value. As expected, the error constant minimizes for smaller γ values.

Complementary, Fig. 5.7 shows the obtained velocity fields for three mesh refinement levels (meshes 1, 3 and 5 in Table 5.1). The expected linear velocity profile can be clearly identified in all cases.

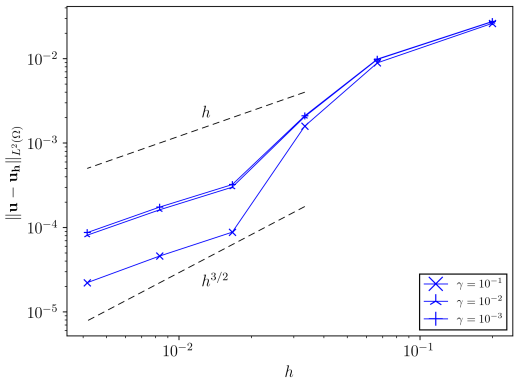
Finally, it has to be highlighted that, even though this test involves a very simple geometry, the obtained results are a very first, but essential, proof of convergence required prior to the application of the proposed technique to more realistic scenarios.

Table 5.4: 2D straight channel (Slip length 10^{-1} interface). Velocity and pressure error norms.

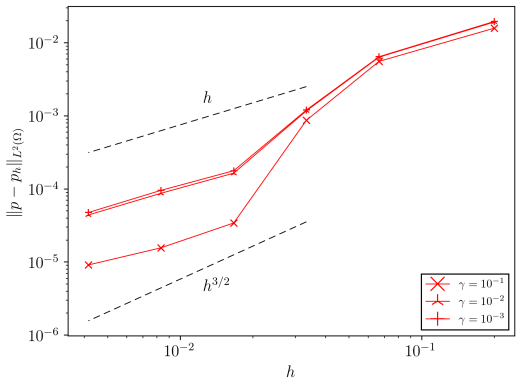
		Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5
$\gamma = 10^{-1}$	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$	0.026176	0.008904	0.001587	8.78378e-05	4.5824e-05	2.21285e-05
	$\|p - p_h\|_{L^2(\Omega)}$	0.015735	0.005540	0.000865	3.40915e-05	1.56027e-05	9.09062e-06
	$\|f_x - f_{x_h}\|_{L^2(\Gamma)}$	0.034265	0.023253	0.019279	0.018424	0.018315	0.018247
$\gamma = 10^{-2}$	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$	0.027379	0.009778	0.002054	0.00023	0.000162	8.08048e-05
	$\|p - p_h\|_{L^2(\Omega)}$	0.019098	0.006327	0.001165	0.000164	8.69214e-05	4.38793e-05
	$\|f_x - f_{x_h}\|_{L^2(\Gamma)}$	0.033049	0.022813	0.019053	0.018319	0.018256	0.018219
$\gamma = 10^{-3}$	$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$	0.027548	0.009871	0.002102	0.000322	0.000174	8.67592e-05
	$\|p - p_h\|_{L^2(\Omega)}$	0.019457	0.006407	0.001196	0.000177	9.52864e-05	4.74011e-05
	$\|f_x - f_{x_h}\|_{L^2(\Gamma)}$	0.032919	0.022768	0.01903	0.018308	0.01825	0.018216

5.3.2 2D flow inside a ring

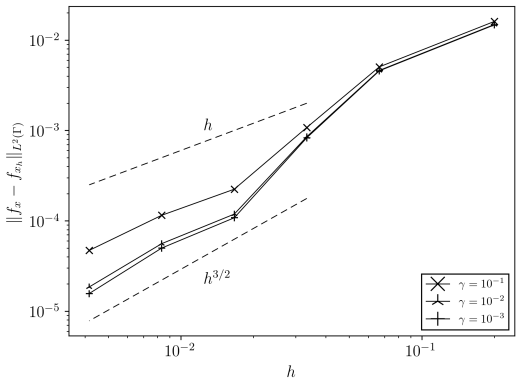
The aim of this test is to obtain the convergence rates of the presented formulation, in both the slip and no-slip limits, when dealing with polygonal approximations of curved boundaries. It is known that the imposition of a slip BC over faceted approximations of curved boundaries can be problematic [106]. This is explained by the fact that the slip BC requires the velocity to be parallel to each face of the discrete polygonal approximation. In consequence, the unique solution that fulfills such requirement in the vertices that are shared between faces is the null velocity. As it is observed in [106, 119], this can lead to a wrong velocity approximation in the slip wall, specially in those cases that present an insufficient discretization of the curved boundary.



(a) Velocity convergence.



(b) Pressure convergence.



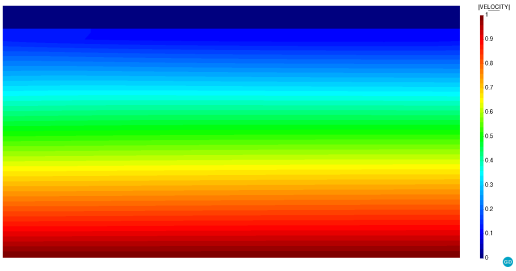
(c) Drag convergence.

Figure 5.6: 2D straight channel (Slip length 10^{-1} interface). Solid lines represent the obtained results. Dashed lines represent h and $h^{3/2}$ convergence rates.

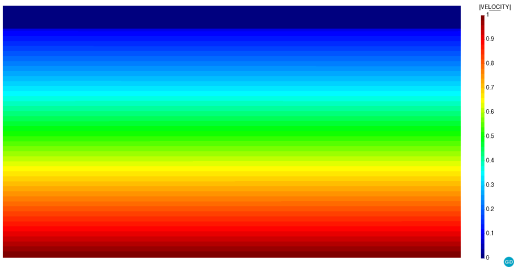
The geometry of the problem consists in two concentric cylinders separated by an interior fluid cavity. The inner cylinder radius is 0.5m while the outer one is 1.0m, meaning that the fluid domain Ω can be described as $(x, y) \in \mathbb{R}^2 : \ 0.25 \leq x^2 + y^2 \leq 1$. The fluid is initially at rest and a unit angular velocity $\bar{v}_\Theta$ is imposed in the outer cylinder to induce the flow. The pressure is fixed to zero along the outer cylinder boundary to obtain a radially symmetric solution.

With regard to the material properties, the dynamic viscosity μ and density ρ are 1.0kg/m^3 and 10^{-3}kg/ms. A second order BDF2 scheme is used for the time discretization. To ensure that the steady state solution is reached, 10 time steps of 200s are solved.

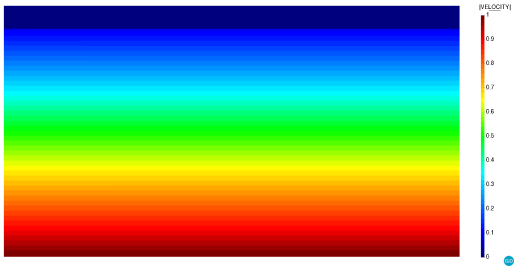
A centered structured mesh is used in all the refinement levels of the study. Note that



(a) Mesh 1.



(b) Mesh 3.



(c) Mesh 5.

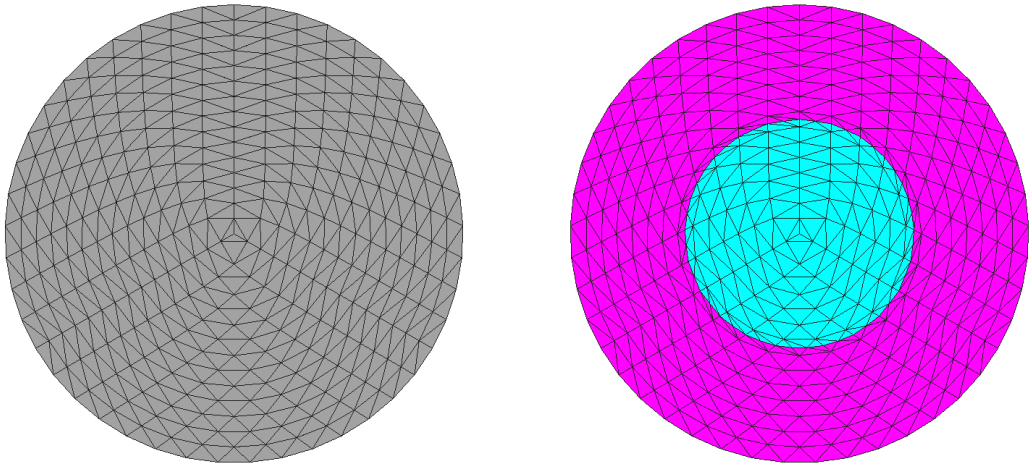
Figure 5.7: 2D straight channel (Slip length 10^{-1} interface). Coarse, medium and fine meshes velocity field.

the outer cylinder is completely meshed disregarding the inner one, which is represented by a radial discontinuous distance function. The characteristic element size together with the number of radial and perimeter subdivisions are collected in Table 5.5. Fig. 5.8 shows the Mesh 1 refinement level in Table 5.5. More specifically, Fig. 5.8a depicts the centered structured pattern while Fig. 5.8b details the discontinuous level set function intersection pattern.

In the following, the convergence of the obtained solutions when a slip and no-slip BC is imposed over the inner boundary is studied by means of the analytical solutions, which can be easily obtained from the mass and momentum conservation equations written in polar coordinates.

Table 5.5: 2D flow inside a ring. Mesh refinement settings.

	Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5	Mesh 6
Element size [m]	0.1428	0.06667	0.03448	0.01754	0.00884	0.00444	0.00223
Radial divisions	7	15	29	57	113	225	449
Perimeter divisions	21	43	85	169	337	673	1345



(a) Mesh capture. (b) Level set intersection pattern.

Figure 5.8: 2D flow inside a ring test case geometry (mesh 1).

Slip interface

In order to ensure a slip-like behavior, the slip length ε is set to 10^8 in all the cases presented in this subsection. The Nitsche penalty constant γ is 0.1 in all the examples. The analytical solution can be expressed in polar coordinates as

$$v_r = 0 \tag{5.11a}$$

$$v_\Theta = r \tag{5.11b}$$

$$p = \frac{\rho}{2}\left(r^2 - 1\right) \tag{5.11c}$$

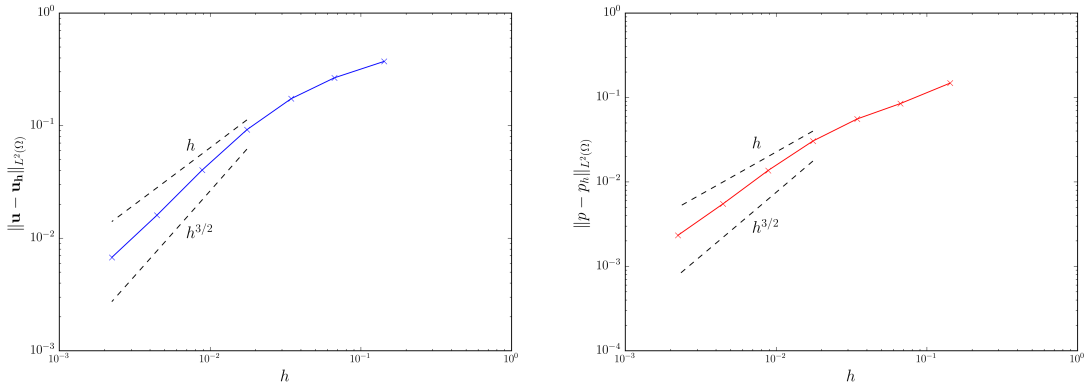
Table 5.6 collects the $L^2(\Omega)$-norm of the velocity and pressure errors. For both velocity and pressure fields, the results display a superlinear convergence rate close to $h^{3/2}$ (Figs. 5.9a and 5.9b), which is in line with slip straight channel observations as well as with the results in [2, 119].

The obtained velocity and pressure fields are shown for three of the mesh refinement levels (mesh 0, 3 and 6) in Fig. 5.11. At first sight, it can be easily observed that the coarsest mesh solution if far from the expected slip behavior. However, the method converges to the slip solution as the mesh is refined. These observations are in agreement with the ones in [106] about the resolution of a similar Stokes problem using a Nitsche imposition of the slip BC. The same issues are also reported in Chapter 4 when solving the same test but using the Navier-Stokes embedded discontinuous purely slip formula-

tion. Having said this, the deterioration of the convergence rates to a superlinear rate below the expected $h^{3/2}$ one is presumably associated to the polygonal approximation of curved boundaries rather than to the Nitsche imposition of the slip BC.

Table 5.6: 2D flow inside a ring (slip interface). Velocity and pressure error norms.

	Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5	Mesh 6
$\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$	0.372217	0.265875	0.173553	0.092081	0.040315	0.016052	0.006761
$\|p-p_h\|_{L^2(\Omega)}$	0.148217	0.084728	0.055737	0.030523	0.013681	0.005508	0.002329



(a) Velocity convergence.

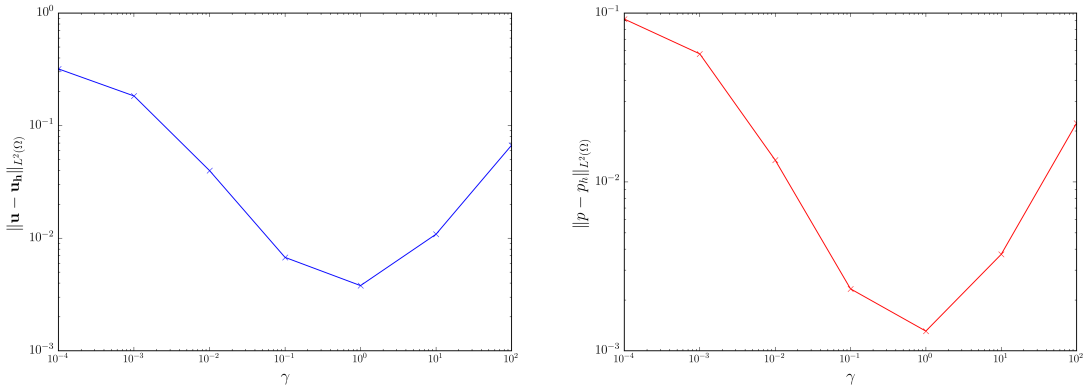
(b) Pressure convergence.

Figure 5.9: 2D flow inside a ring (slip interface). Solid lines represent the obtained results. Dashed lines represent h and $h^{3/2}$ convergence rates.

Complementary, the sensitivity of the formulation to the value of the penalty constant γ when $\varepsilon \to \infty$ is also assessed. To that purpose, the same example is solved but using the finest mesh and a set of γ values ranging from from 10^{-4} to 10^{2}. Table 5.7 collects the velocity and pressure error $L^2(\Omega)$-norms for all the γ values. These results are also depicted in Fig. 5.10. It can be observed that both error norms minimize for γ values between 0.1 and 10, being the minimum values around the unit γ value.

Table 5.7: 2D flow inside a ring (slip interface). γ sensitivity analysis.

γ	10^2 0	10^1	10^0	10^{-1}	10^{-2}	10^{-3}	10^{-4}
$\|\mathbf{u}-\mathbf{u}_h\|_{L^2(\Omega)}$	0.066914	0.010858	0.003796	0.006761	0.03991	0.183505	0.319047
$\|p-p_h\|_{L^2(\Omega)}$	0.022297	0.003725	0.0013086	0.002329	0.013493	0.057347	0.092092



(a) Velocity error norm.

(b) Pressure error norm.

Figure 5.10: 2D flow inside a ring (slip interface). Error norms for different γ values.

No-slip interface

In this case the slip length ε is set to 0 to enforce a no-slip wall behavior in the embedded interface. The Nitsche penalty constant γ is kept as 0.1. As in the slip case, the results and convergence assessment are based on the analytical solution, which in in polar coordinates reads as

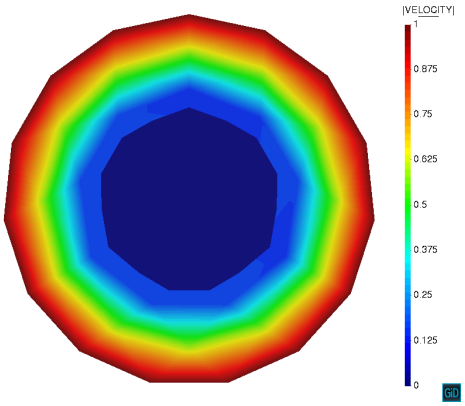
$$v_r = 0 \tag{5.12a}$$

$$v_\Theta = \frac{\bar{v}_\Theta}{1-0.5^2}\left(\frac{r^2-0.5^2}{r}\right) \tag{5.12b}$$

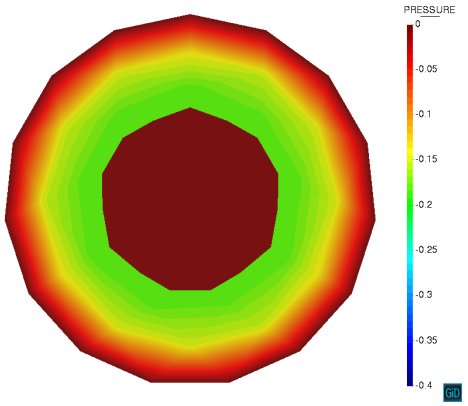
$$p = \frac{\bar{v}_\Theta \rho}{2(1-0.5^2)^2}\left(r^2 - log(r) - \frac{0.5^2}{r^2} - 0.9375\right) \tag{5.12c}$$

Table 5.8 collects the velocity and pressure error $L^2(\Omega)$-norms. These values are also depicted in Figs. 5.9a and 5.9b. It can be observed that both velocity and pressure fields exhibit a linear convergence rate. Such deteriorated convergence rate can be again associated to the inability of the Ausas FE space to properly capture the shear effects within the intersected elements.

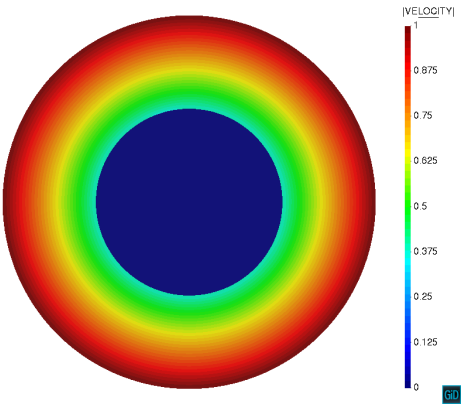
Fig. 5.11 depicts the solution for three of the mesh refinement levels (meshes 0, 3 and 6). It can be observed that the solution tends to the analytical one as the mesh is refined.



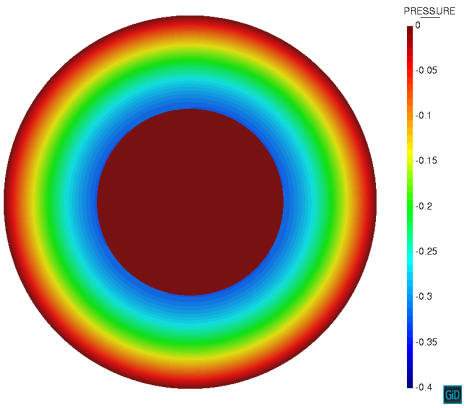
(a) Mesh 0 velocity.



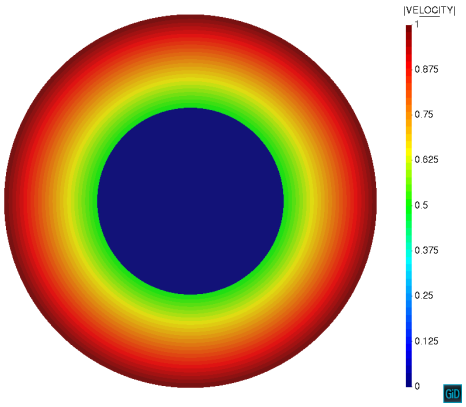
(b) Mesh 0 pressure.



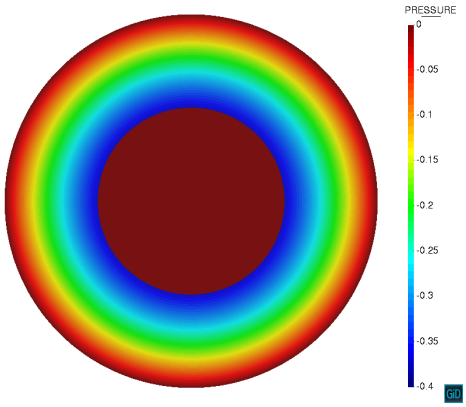
(c) Mesh 3 velocity.



(d) Mesh 3 pressure.



(e) Mesh 6 velocity.

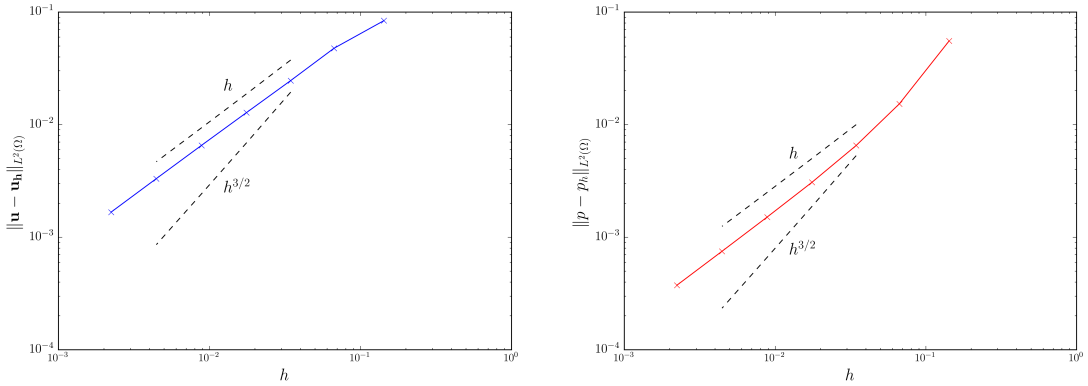


(f) Mesh 6 pressure.

Figure 5.11: 2D flow inside a ring (slip interface). Coarse, medium and fine meshes solutions.

Table 5.8: 2D flow inside a ring (no-slip interface). Velocity and pressure error norms.

	Mesh 0	Mesh 1	Mesh 2	Mesh 3	Mesh 4	Mesh 5	Mesh 6
$\|\mathbf{u} - \mathbf{u}_h\|_{L^2(\Omega)}$	0.083878	0.047655	0.024639	0.012780	0.006550	0.003318	0.001670
$\|p - p_h\|_{L^2(\Omega)}$	0.055474	0.015289	0.006531	0.003074	0.001510	0.000751	0.000374



(a) Velocity convergence.

(b) Pressure convergence.

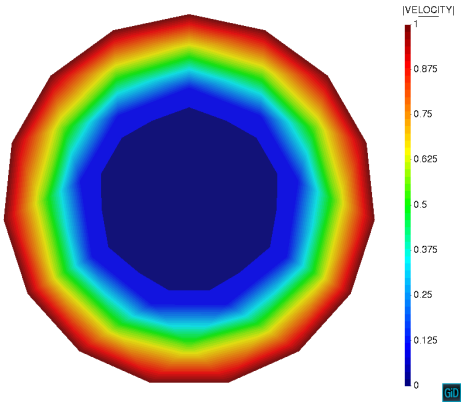
Figure 5.12: 2D flow inside a ring (no-slip interface). Solid lines represent the obtained results. Dashed lines represent h and $h^{3/2}$ convergence rates.

5.3.3 2D flow around pressurized cylindrical membrane

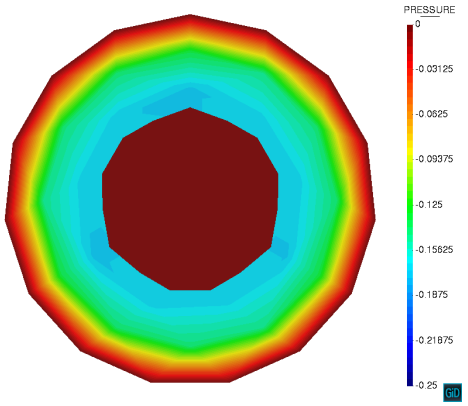
This example is specifically conceived to prove the capability of the pseudo-compressible term in Eq. 3.6b to prevent the pressure field to blow up when isolated cavities of fluid appear.

To that purpose this test solves the fluid flow around an embedded cylindrical membrane that is filled with fluid. After the ability of the Ausas FE space to disconnect both sides of the level set, the embedded cylinder becomes in an isolated fluid cavity with no pressure constraint, and thus requires the pseudo-compressibility term in the mass conservation equation to keep the pressure bounded.

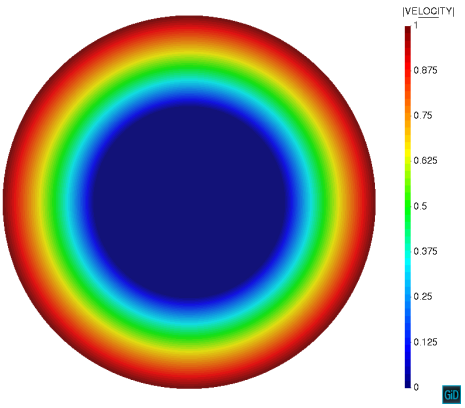
The problem geometry consists in a 1x5m channel containing a 0.1m radius cylinder. Assuming that the bottom left corner of the channel coincides with the origin, the center of the cylinder is located in the (1.25,0.5)m coordinates. With regard to the fluid properties, ρ and μ are set to 1.0kg/m^3 and 10^{-2}kg/ms. The speed of sound velocity c is equal to 10^3m/s, which is a feasible value for a liquid medium. Inside the cylinder (Figs. 5.14a and 5.14b), a volume force $\mathbf{b}$ equal to $[0.0, -10.0]$m/s^2 is applied to ensure that the initial solution (zero velocity and pressure) is not in equilibrium.



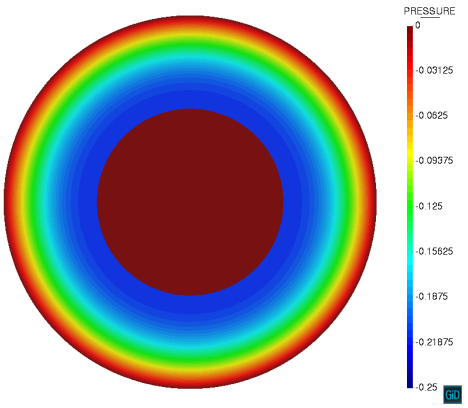
(a) Mesh 0 velocity.



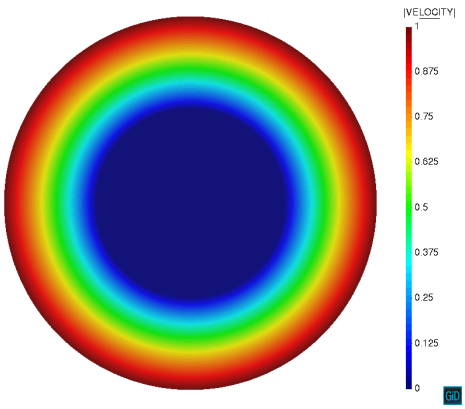
(b) Mesh 0 pressure.



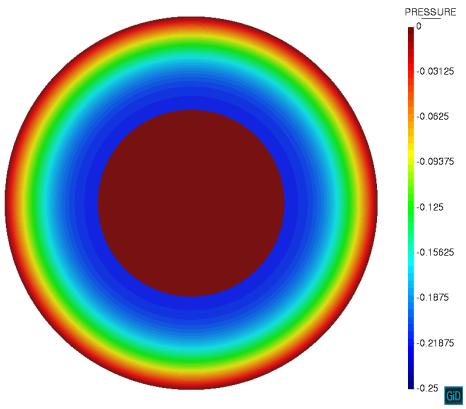
(c) Mesh 3 velocity.



(d) Mesh 3 pressure.



(e) Mesh 6 velocity.



(f) Mesh 6 pressure.

Figure 5.13: 2D flow inside a ring (no-slip interface). Coarse, medium and fine meshes solutions.

A constant inlet velocity profile of 1m/s is imposed in the left edge of the channel while a symmetry condition (zero normal velocity and free tangential velocity) is enforced in the top and bottom walls. In order to get a steady state solution, the cylinder is modeled as a slip body by setting the slip length to 10^8. Furthermore, it is important to clearly state that the pressure is not fixed in any point of the computational domain.

The time discretization is done using a BDF2 scheme with a constant time step equal to 1s. To ensure that the steady solution is reached, the simulation is run for a total time of 100s. The channel is meshed with a structured grid made with around 200k linear triangular elements, whose average size is 7e-3m.

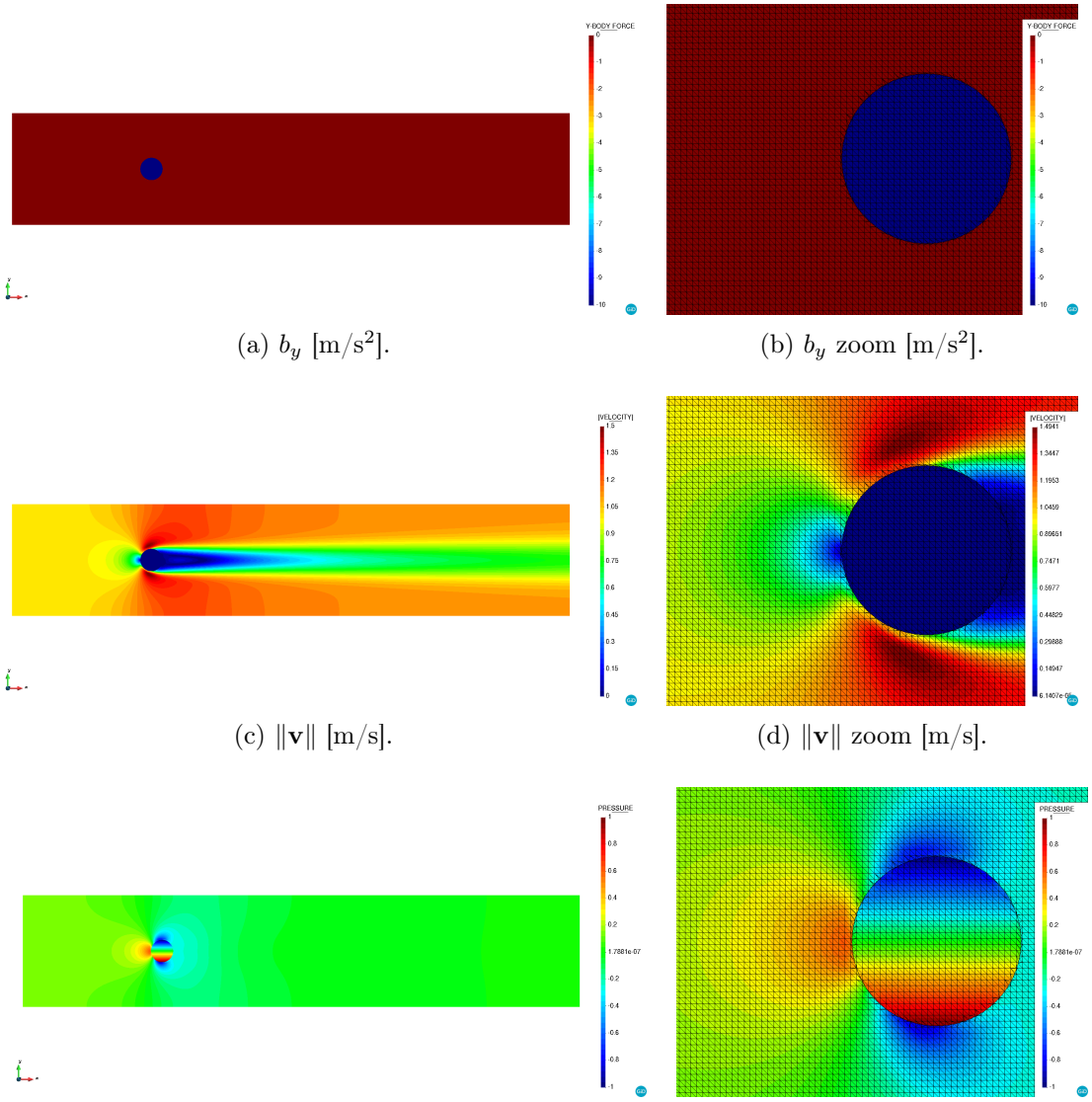
Figs. 5.14c and 5.14e depict the obtained solution. As expected, the flow around the cylinder presents an upstream stagnation point and a symmetric steady wake. Inside the cylinder (Fig. 5.14d and 5.14f), the expected hydrostatic solution can be clearly identified too. These results prove that the addition of a slight compressibility is enough to keep the pressure bounded, and therefore to ensure convergence, despite the presence of isolated fluid cavities with no pressure constraint.

Finally, it is important to spend some words on the relevance that these results have for the application of the proposed technique to real life problems involving "dirty" input geometries. Although in this case the unbounded pressure issue could be immediately tackled by fixing the pressure in any point inside the cylinder, this is likely impossible to be done in a real scenario since complex geometries can be rarely described using an analytical level set functions as the cylinder case. Hence, the use of the quasi-incompressible form of the N–S equations is of great utility for the extension of the proposed method to either real geometries or complex moving domains as it makes possible to automatically handle such unbounded pressure cavities regardless their a priori unknown location.

5.3.4 2D elbow with internal wall

This example, which was firstly proposed in [54], consists in a 90° curved 2D pipe conforming an elbow shape (Fig. 5.15). A zero-thickness wall is placed inside the curved pipe, generating two separated fluid ducts whose cross section area varies in space. Even though for such a simple geometry the zero-thickness wall could be alternatively modelled by duplicating the interface nodes, in this case it is represented using a discontinuous level set function.

The previously described elbow geometry is solved considering a slip interface, as was



(a) b_y [m/s^2]. (b) b_y zoom [m/s^2].

(c) $\|\mathbf{v}\|$ [m/s]. (d) $\|\mathbf{v}\|$ zoom [m/s].

(e) p [Pa]. (f) p zoom [Pa].

Figure 5.14: 2D flow around pressurized cylindrical membrane. Body force, $\|\mathbf{v}\|$ and p fields for c 10^3m/s. In the left hand side a general view of the entire domain is presented. In the right hand side a zoom on the cylinder region with the background mesh, including the sub-triangles arising from the level set intersection, is shown.

done by the original authors in [54], as well as a no-slip one. The obtained solutions are compared with the ones obtained with a body fitted reference solver as well as with the literature data. Furthermore, the analytical values, which can be easily obtained from the mass conservation principle, are also considered in the results assessment.

With regard to the fluid properties, they are selected such that the Re number is unitary.

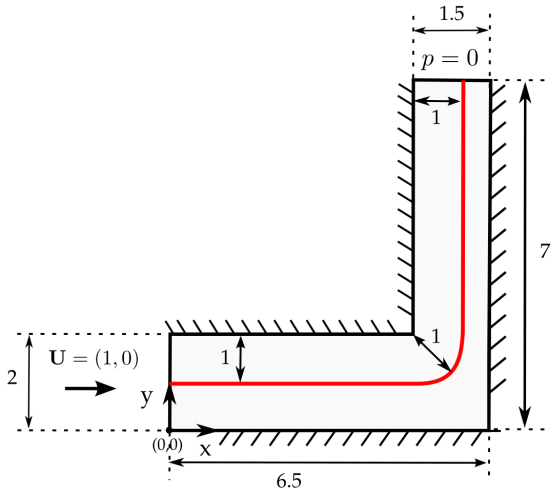


Figure 5.15: 2D elbow with internal wall. Problem geometry (source [54]).

Considering the width of each one of the ducts as reference length, the unit Re number can be achieved by setting the density ρ to 1kg/m^3, the dynamic viscosity μ to 1kg/ms and a constant inlet velocity u_x of 1m/s. In all cases the outer walls are assumed to have null velocity (no-slip condition) and the pressure is fixed to zero along the outlet. The BDF2 time scheme is used for the time discretization with a constant time time step of 10^{-2}s. The total simulation time is 1s, which is enough to reach a steady solution.

Slip interface

For the sake of a fair comparison, the meshes employed in this case are set as similar as possible to the reference ones in [54]. Hence, four structured grids (denoted as coarse, medium, fine and very fine) are used. Table 5.9 collects the number of elements of each one of the refinement levels together with the reference ones in [54].

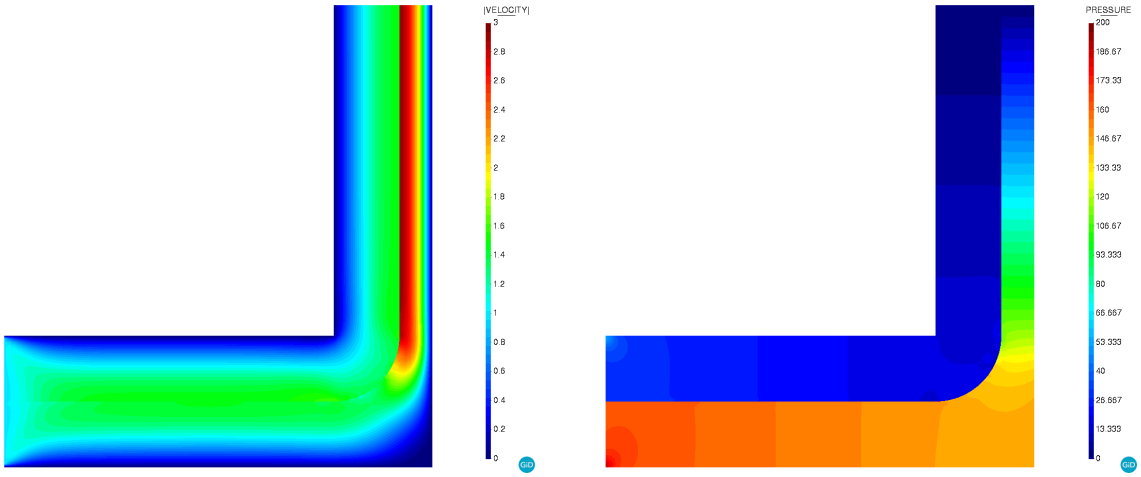
Table 5.9: 2D elbow with internal wall. Number of elements for different refinement levels.

Mesh	Reference	Present work
Coarse	2400	2300
Medium	9600	8900
Fine	38400	36200
Very fine	153600	147000

Considering that the reference solution assumes a slip behavior of the intermediate wall,

the slip length ε is set to 10^8, which is a large enough value to yield a wall behavior close to the slip limit.

Prior to any convergence assessment, the solution obtained with the medium refinement mesh is presented in Fig. 5.16. As it can be observed, the flow discontinuities arising from the introduction of the zero-thickness intermediate wall are perfectly captured by the formulation. As expected, a pressure gradient appears in the right duct due to the contraction generated by the wall. This turns into a flow acceleration to preserve the inlet flow rate. On the contrary, the left duct pressure gradient is almost constant since there is no variation in the cross section. The minor differences between the inlet and outlet pressure values are due to the local energy losses that occur when the flow changes its direction in the elbow.



(a) Velocity modulus [m/s].

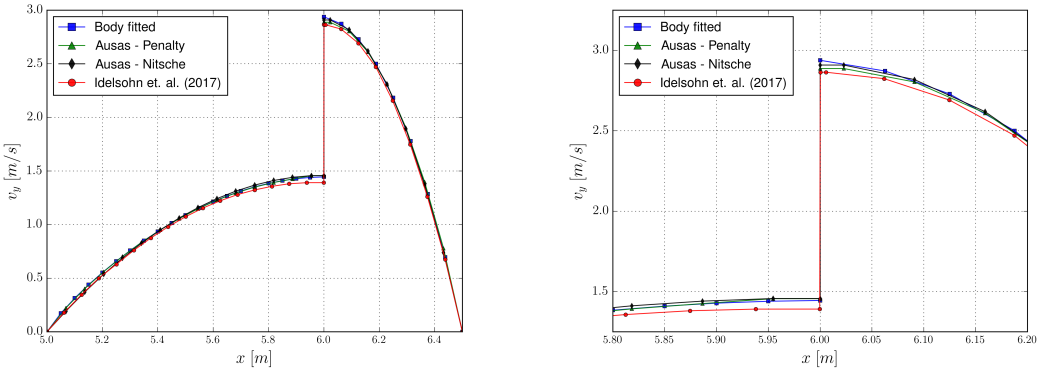
(b) Pressure [Pa].

Figure 5.16: 2D elbow with internal wall (slip interface). Medium mesh solution.

Figs. 5.17 and 5.18 collect the results assessment. On the one hand, Fig. 5.17 compares the obtained medium mesh outlet y-velocity with the reference, body fitted and Chapter 4 ones. A good correlation can be observed between all the values. Minor differences can only be observed between the penalty-based formulation and the Nitsche-based one, being the Nitsche-based formulation slightly more accurate in the values close to the embedded wall. This is more noticeable in the right duct due to the higher fluid velocity.

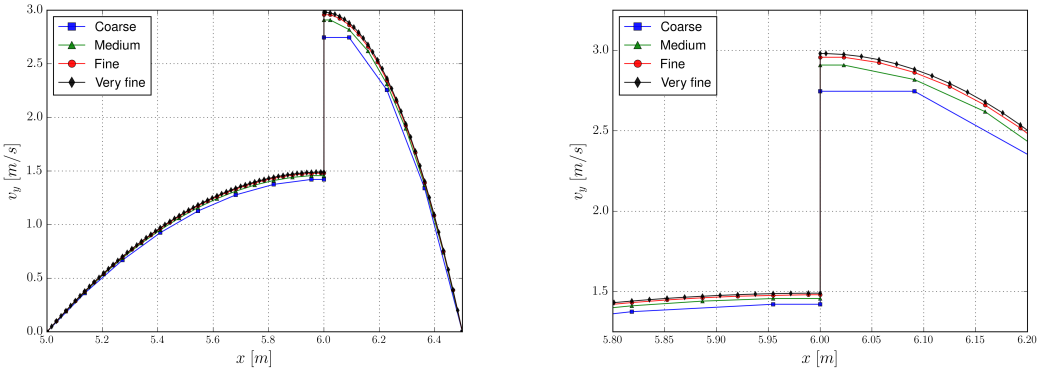
On the other hand, Fig. 5.18 depicts the convergence of the outlet y-velocity when the mesh is refined. It can be noted that the obtained values converge to the reference ones. This is even more evident in the values close to the embedded wall (Table 5.10), which

significantly improve upon refinement.



(a) General view. (b) Interface region zoom.

Figure 5.17: 2D elbow with internal wall (slip interface). Outlet v_y comparison [m/s].



(a) General view. (b) Interface region zoom.

Figure 5.18: 2D elbow with internal wall (slip interface). Outlet v_y convergence [m/s].

Finally, Table 5.10 compares the outlet maximum y-component velocity values with the analytical solution ones. According to [54], if it is considered that the flow is parabolic in both sides of the intermediate wall, it can be proven that the analytical maximum velocity is 1.5m/s in the left duct and 3m/s in the right one. The absolute errors that are obtained taking these two values as reference are collected in Table 5.10. Once again, it can be observed that the values converge to the expected solution as the mesh is refined.

Table 5.10: 2D elbow with internal wall (slip interface). Outlet maximum velocity y-component for different refinement levels [m/s].

	$\|\mathbf{v_y}\|_{max}^{left}$	abs. err.	$\|\mathbf{v_y}\|_{max}^{right}$	abs. err.
Expected	1.5	-	3.0	-
Coarse	1.4213	0.0787	2.7459	0.2541
Medium	1.4569	0.0431	2.9089	0.0911
Fine	1.4805	0,0196	2.9574	0,0426
Very fine	1.4902	0,0098	2.9804	0,0196

No-slip interface

In this section the 2D elbow test case is modified to consider a no-slip behavior in the intermediate embedded wall. Although the geometry is the same, it is convenient to modify the inlet to consider a parabolic profile in both ducts according to the no-slip wall behaviour. However, preliminary tests showed that the flow rate might be influenced by the level intersection pattern in the inlet owing to the Ausas FE space interpolation properties. Therefore, the inlet function is not imposed along the entire left edge but along the 75% of the length of each duct to ensure that the intersection pattern does not affect the inflow. Thus, the bottom duct inlet function reads as

$$v_x(y) = y(-12.642y + 9.4812) \tag{5.13}$$

while the top one is

$$v_x(y) = -12.642y^2 + 41.0864y - 31.6049 \tag{5.14}$$

Note that these parabolic functions preserve the original benchmark flow rates.

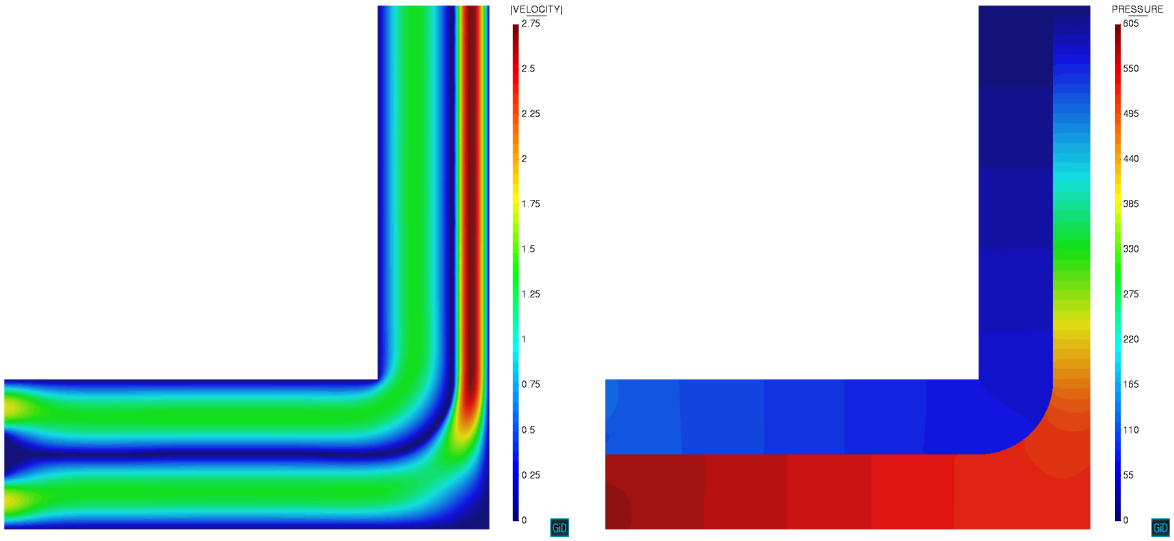
Taking into account the influence that the no-slip boundary condition has in the solution, a new set of finer meshes is used to properly capture the expected parabolic velocity profile. The new mesh sizes are collected in Table 5.11.

Fig. 5.19 depicts the medium mesh embedded solution. As expected, the imposition of the no-slip constraint in the intermediate wall yields a symmetric parabolic profile in both ducts. Furthermore, the pressure gradient required to keep the inlet flow rate also appears in both ducts.

Fig. 5.20 compares the three mesh refinements with the reference body fitted solution,

Table 5.11: 2D elbow with no-slip internal wall. Element size for different refinement levels.

Mesh	Element size
Body fitted	0.01
Embedded (coarse)	0.07
Embedded (medium)	0.0175
Embedded (fine)	0.01



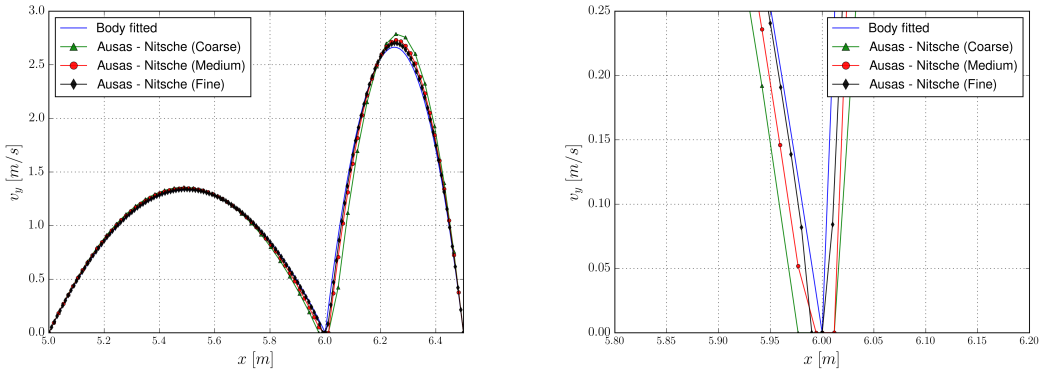
(a) Velocity modulus [m/s].

(b) Pressure [Pa].

Figure 5.19: 2D elbow with internal wall (no-slip interface). Medium mesh solution.

proving that the Nitsche embedded solution converges to the reference one as the mesh is refined. It is worth mentioning that the lower accuracy solutions have a higher peak velocity value. This is explained by the inability of the Ausas FE space to properly capture the solution in the elements intersected by the no-slip embedded wall. Considering that no velocity gradient can be captured in one side of these elements, the velocity becomes null (in a variational sense) over the entire element (Fig. 5.20b). In consequence, the flow needs to accelerate in the rest of the duct to keep the inlet flow rate.

Similarly to what has been done for the slip case, Table 5.12 compares the reference analytical values, which can be easily obtained considering a parabolic flow and mass conservation, with the three refinement levels ones. Again, it can be observed that the no-slip solution converges to the reference one as the mesh is refined.



(a) General view. (b) Interface region zoom.

Figure 5.20: 2D elbow with internal wall (no-slip interface). Outlet v_y [m/s].

Table 5.12: 2D elbow with internal wall (no-slip interface). Outlet maximum velocity y-component for different refinement levels [m/s].

	$\|\mathbf{v_y}\|^{left}_{max}$	abs. err.	$\|\mathbf{v_y}\|^{right}_{max}$	abs. err.
Expected	1.3332	-	2.6664	-
Coarse	1.3577	0.0245	2.7874	0.121
Medium	1.3492	0.016	2.7304	0.064
Fine	1.3370	0.0038	2.7011	0.0347

5.3.5 2D flow around thin-walled arch profile

This example is conceived to be a transition case between the more academic examples presented so far and the real application in the next subsection. Hence, the objective is to assess the performance of the formulation when very large Re are involved but still keeping an elementary 2D geometry.

The problem geometry consists in a 20x100m rectangular channel in which a 4m chord and 1m height arch geometry is placed. Note that the embedded arch geometry is thought to be reasonably similar to the cross-section of a real 3D thin-walled structure (e.g. a sail or lightweight roof cover). Assuming that the bottom left corner of the channel is placed in the (0,0)m coordinates, the arch can be created from the three points (-2,-0.5), (0,0.5) and (2,-0.5)m. These geometry settings yield a 5% blockage coefficient. The material properties are set such that the Re is 10^5. Taking the height

of the arch as reference length for the Re calculation, this can be achieved by setting ρ and μ to 1kg/m^3 and 10^{-5}kg/ms.

The BCs are rather simple. A constant inlet velocity of 1m/s is imposed in the left edge of the channel while the pressure is fixed to 0Pa in the right one. A symmetry BC (zero normal velocity and free tangential velocity) is imposed in the top and bottom edges. Concerning the embedded body BC, two scenarios are considered. The former is an ideal slip case, in which the slip length is set to 10^8m, whereas the latter entails a stick condition, whose slip length is equal to 10^{-3}m. Note that this value is of the same order of magnitude as the corresponding theoretical boundary layer thickness.

Both the embedded and the reference body fitted meshes have around 320k linear triangular elements. It is important to point out that in the body fitted case the boundary edges representing the analysed arch geometry are duplicated in order to model the velocity and pressure discontinuities. Concerning the time discretization, the BDF2 scheme is used again. In all cases the time step is 0.05s.

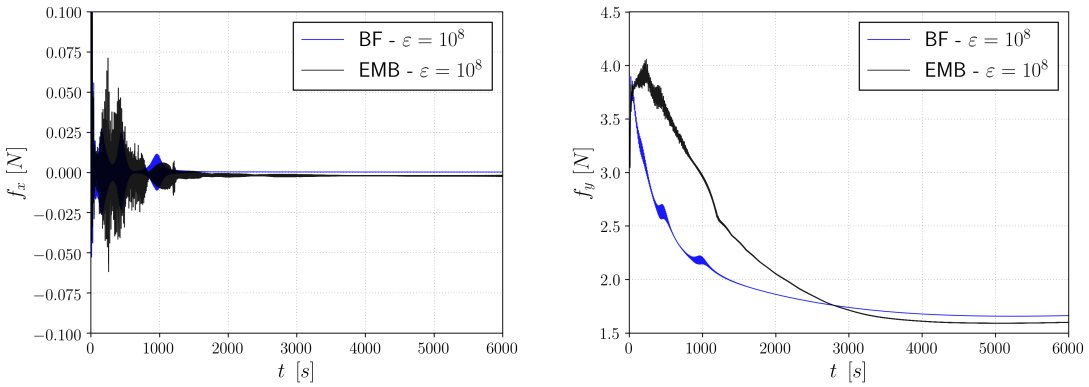
Slip interface

Fig. 5.21 presents the time evolution of the drag and lift forces. As it can be observed, both the body fitted and the embedded solutions reach a steady state solution with null horizontal drag force. Concerning the vertical lift force, the embedded solution one is 1.60N while body fitted reference one is 1.66N. Although the steady values are similar, it can be observed that the embedded solution is more chaotic during the initial transient phase, and thus requires more time to approach the steady state.

Complementary, Figs. 5.22 and 5.23 compare the body fitted and embedded steady state solutions, which are in remarkably good agreement. Besides, it can be observed the steady solution presents a symmetric pattern that, according to the d'Alembert's paradox, explains the null horizontal drag component.

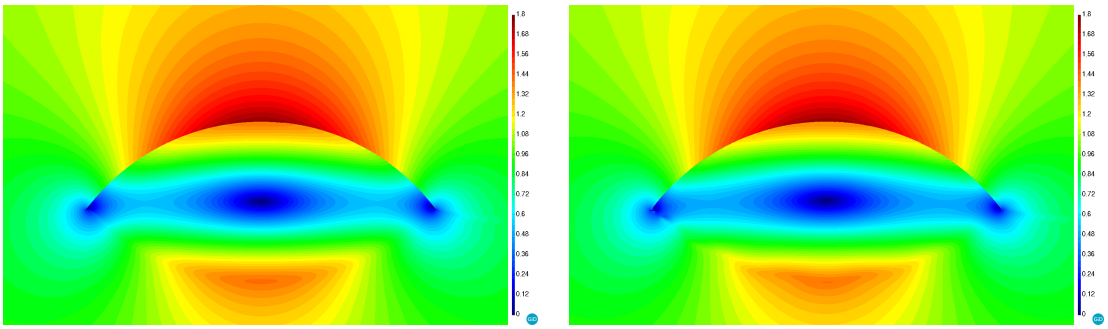
No-slip interface

As it is done in the slip interface case, Fig. 5.24 presents the time evolution of the drag and lift forces. Since in this case the solution is no longer steady, this information is accompanied with the corresponding histograms. It can be observed that the embedded and body fitted drag force histograms are in really good agreement, being the average drag value around the 0.2N (Fig. 5.24b). To what concerns the lift force, the embedded



(a) Drag force [N]. (b) Lift force [N].

Figure 5.21: 2D flow around thin-walled arch profile (slip interface). Drag and lift evolution.



(a) Body fitted. (b) Embedded.

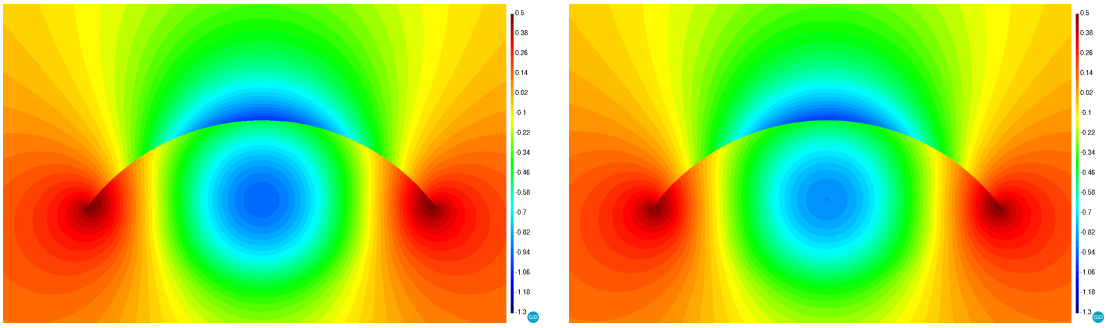
Figure 5.22: 2D flow around thin-walled arch profile (slip interface). Velocity modulus [m/s].

values are more disperse than the body fitted ones (Fig. 5.24d).

Fig. 5.25 depicts a velocity and pressure snapshot of the obtained embedded solution. The presence of a stagnation point in the front end of the arch can be noted as well as the velocity and pressure discontinuities that arise after embedding the arch geometry.

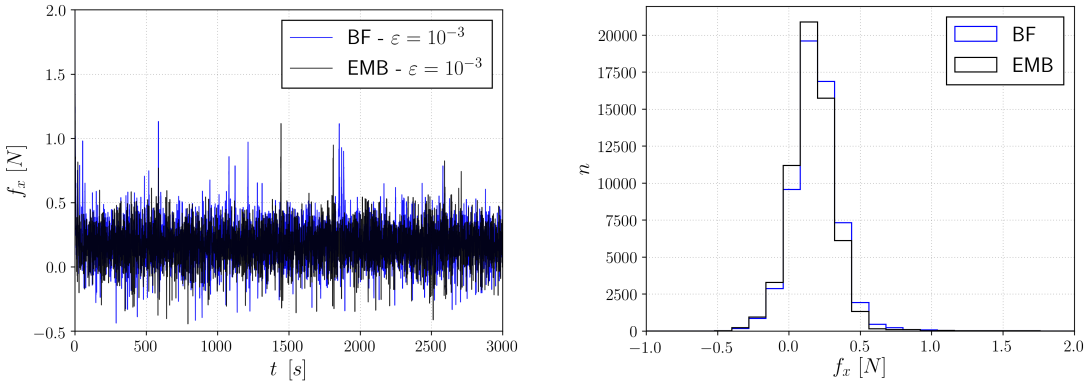
Moreover, the turbulent nature of the flow, which separates in the top part of the arch, can be also observed. Small vortexes appear after the separation point, whose location is approximately at one quarter of the total arch length. Besides these, a larger vortex appears in the trailing part of the arch.

Last but not least, it is important to highlight the capability of the method to correctly impose not only the slip condition but a wall law-like one. This capability would be of great utility for the extension of the proposed technique to real engineering applications, in which the resolution of the boundary layer is normally not affordable.

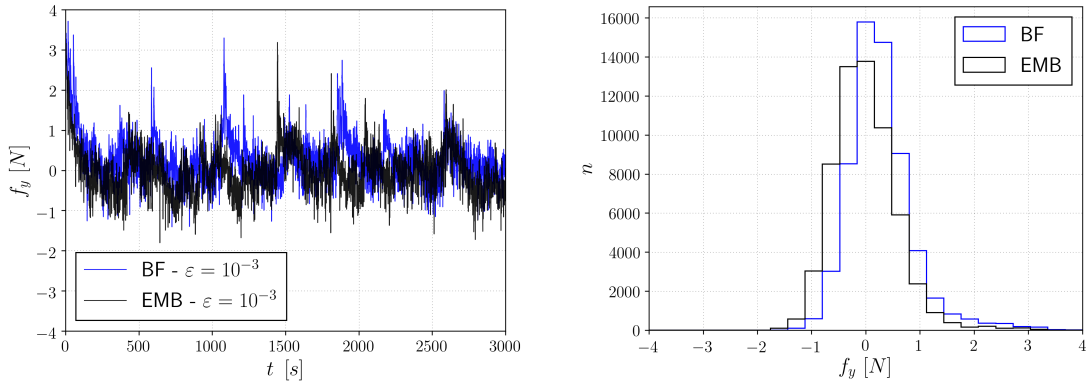


(a) Body fitted. (b) Embedded.

Figure 5.23: 2D flow around thin-walled arch profile (slip interface). Pressure [Pa].

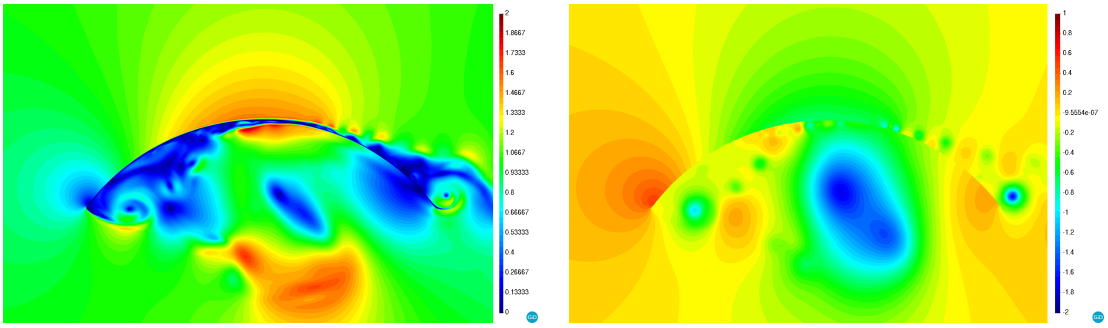


(a) Drag force evolution[N]. (b) Drag force histogram [N].



(c) Lift force evolution [N]. (d) Lift force histogram [N].

Figure 5.24: 2D flow around thin-walled arch profile (no-slip interface). Drag and lift force.



(a) Velocity modulus [m/s]. (b) Pressure [Pa].

Figure 5.25: 2D flow around thin-walled arch profile (no-slip interface). Embedded solution.

5.3.6 3D flow around a sailboat

The objective of this last example is to showcase the capabilities of the discontinuous Navier-slip formulation to efficiently solve real engineering problems. In order to exploit the ability of the method to conveniently deal with membrane-like bodies, the chosen geometry consists in a real 3D sailboat. It is due mentioning that no reference solution of any type is available for the problem at hand. Hence, this example only represents a proof-of-concept of a potential industrial application.

Figs. 5.26a and 5.27a depict the geometry of the sailboat, which has been provided by Juan Kouyoumdjian Naval Architects. The length and beam of the hull are 18.3m and 5.75m while the height of mast, whose axis is placed in the origin, is around 27m.

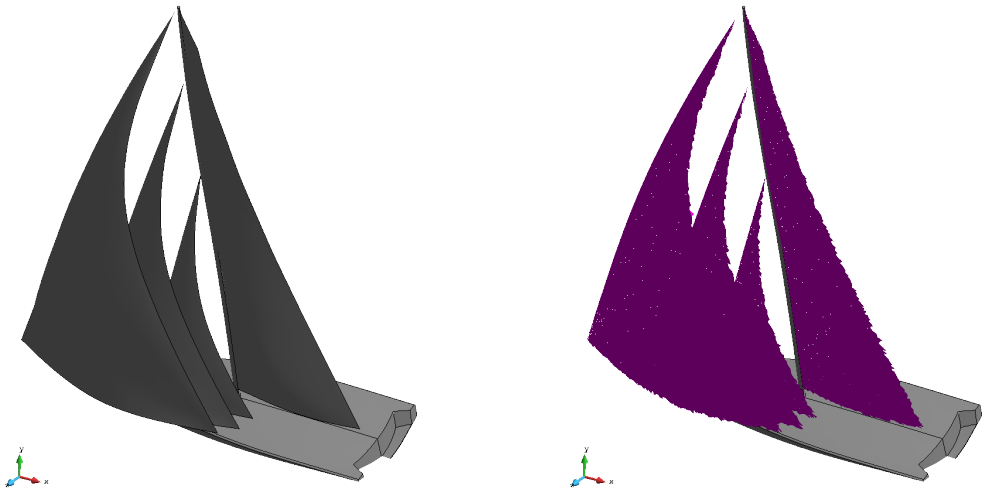
The computational domain is a 80m radius and 70m height cylinder centered in the (13.8,0,16.5)m coordinates (Fig. 5.28). The lateral surface of the cylinder is divided in two regions. The light green one represents the outlet, which comprises one quarter of the total skin, while the light pink one is the inlet.

Aiming to exploit the features of the method, the input geometrical entities are separated in two sets according to their volume nature. On the one hand, those entities that conform bodies with well defined internal volume (i.e. the hull and the mast), that is to say, those ones that can be straightforwardly represented using a body conforming discretization, are collected in one set. On the other hand, another set containing all the surfaces that represent the membrane sails is created.

By doing so it is possible to take advantage of the implicit representation to avoid the volume meshing of the thin-walled sails. Hence, the mainsail as well as the headsails are considered as four independent embedded volumeless bodies which are represented

by a discontinuous level set function.

Unlike the previous examples, in which the level set can be computed with a simple analytical function, this example requires the use of a level set calculation algorithm [10]. Figs. 5.26 and 5.27 compare the input boat geometry to the one used in the resolution of the problem, which includes the level set representation of the sails. Although the this is in general terms very accurate, the boundaries of the reconstructed geometry exhibit a "saw-toothed" pattern. This behavior, which is also reported for a similar case in [119], is a consequence of the nature of the level set method, which cannot represent the partial intersections that do not split the element in two complete parts.

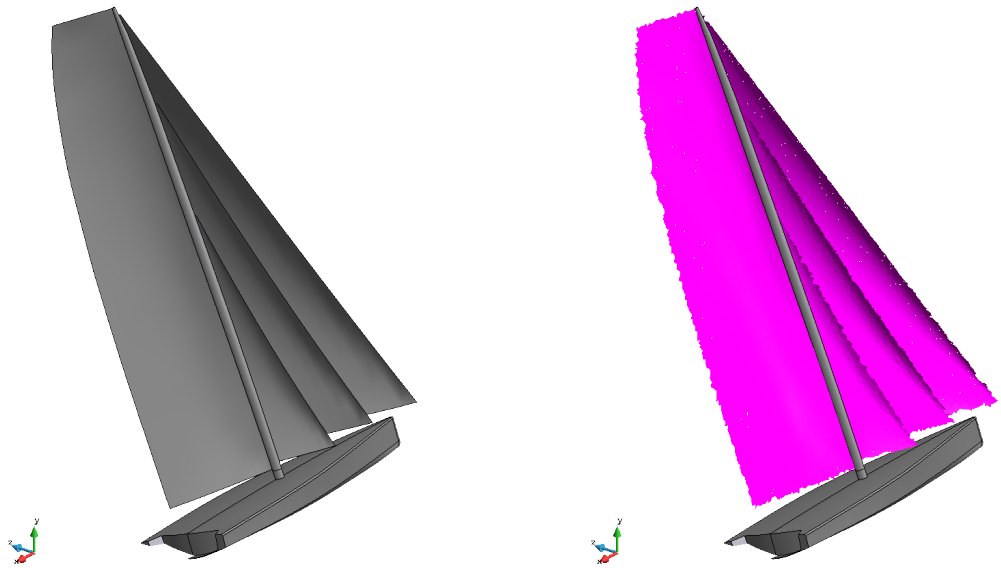


(a) Original geometry. (b) Computational model.

Figure 5.26: 3D flow around a sailboat. Sailboat top-side view.

The boundary conditions are rather simple. Both the hull and the mast are slip. The same behaviour is also imposed in the top and bottom surfaces of the computational domain. In the outlet surface the pressure is fixed to 0Pa. In the inlet surface a wind velocity of 15knots (7.72m/s) is imposed in the (1,0,1) direction, which is rotated 45° with respect to the alignment of the hull. With regard to the Navier-slip BC in the embedded sails, the slip length ε is set to 10^{-2}m while the penalty constant γ is equal to 0.25. The problem is run for 20s using a time step of 0.025s. The computational domain is meshed with 3.3M linear tetrahedra. Fig. 5.29 shows a detail the body conforming volume mesh surrounding the hull and the mast.

Despite the lack of reference results, the obtained velocity and pressure fields are con-



(a) Original geometry S (b) Computational model.

Figure 5.27: 3D flow around a sailboat. Sailboat top-rear view.

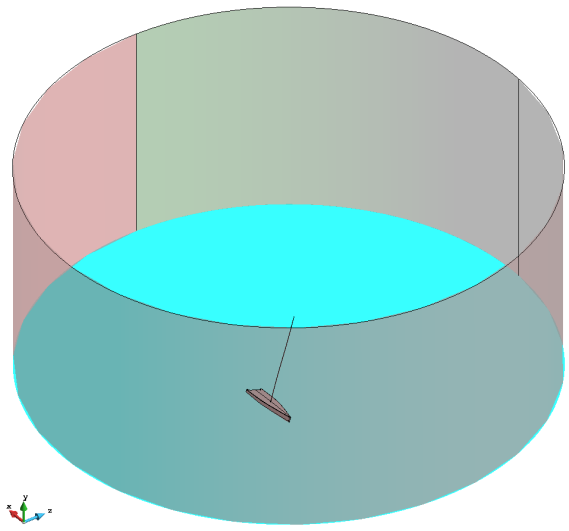


Figure 5.28: 3D flow around a sailboat. Computational domain.

vincing. As it can be observed in Fig. 5.30c, the discontinuous Navier-slip formulation is capable of representing the pressure discontinuity between the two sides of the embedded sails. More specifically, a positive overpressure appears in the windward region of the sails (Fig. 5.30a) while a suction occurs in the leeward one (Fig. 5.30b). Such

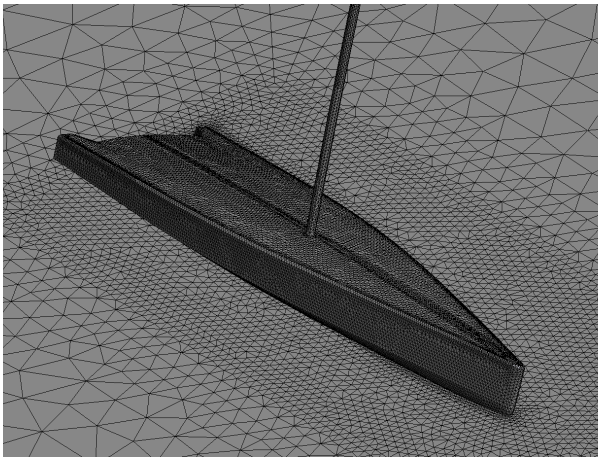


Figure 5.29: 3D flow around a sailboat. Hull and mast mesh detail.

pressure discontinuity generates the wind thrust of the sailboat.

Figs. 5.31, 5.32, 5.33 and 5.34 present a set of horizontal cross sections of the velocity vector field. It is interesting to comment the effect that the slip length has in the solution, which can be clearly noted in the front sail, specially in the plane y equal to 5m cross section (Fig. 5.32). Even though the flow separation that occurs in the front edge of the sail, the velocity is neither null nor maximum in the sail surface. Instead, it behaves as a wall law, which is expected to be the nature of the Navier-slip BC. Despite the lack of a more rigorous study about the effect that more feasible values of the ε could have in the solution, the results of these preliminary experiments appear to be promising from the industrial application perspective.

5.4 Conclusion

This chapter presents a novel embedded formulation for the weak imposition of the Navier-slip BC in viscous incompressible Navier-Stokes flows. The proposal is valid not only for volumetric bodies but also for membrane-like ones that feature no internal volume.

The new formulation relies on the use of the Ausas discontinuous FE space in combination with a Nitsche-based imposition of the Navier-slip BC. Provided a discontinuous level set function representing the immersed objects, the Ausas FE space is able to capture the velocity and pressure jumps inside the intersected elements. On top of that, the Navier-slip BC that is enforced in the level set intersections makes possible to represent

any wall behaviour from the slip to the no-slip limits.

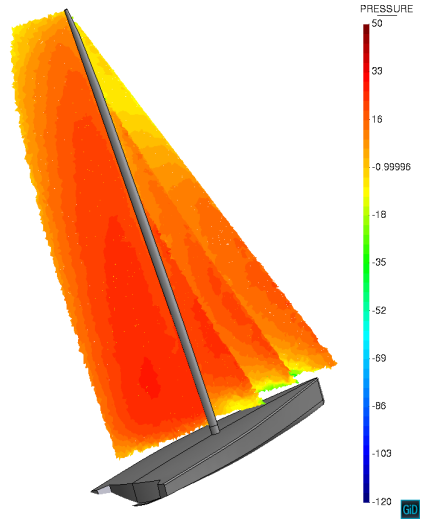
Furthermore, the proposed technique has neither blending elements nor extra degrees of freedom. While the former becomes relevant when working in distributed memory environments (MPI), the latter avoids the need of reconstructing the system matrix graph each time the level set is updated.

The convergence and accuracy of the method is assessed by solving several examples, which consider different values of the slip length ranging from the slip to the no-slip limits. For ε values close to the slip limit, the velocity and pressure convergence rates are around $h^{3/2}$, which is in accordance with the interpolation properties of the Ausas FE space [2, 119]. However, the convergence rates deteriorate to h when the slip length approaches the no-slip limit. Such degradation can be perfectly expected owing to the inability of the Ausas FE space to properly represent the gradients in the intersected elements. Nonetheless, the wall shear effects are roughly recovered upon mesh refinement as the thickness of the constant solution (or null gradient) region is reduced.

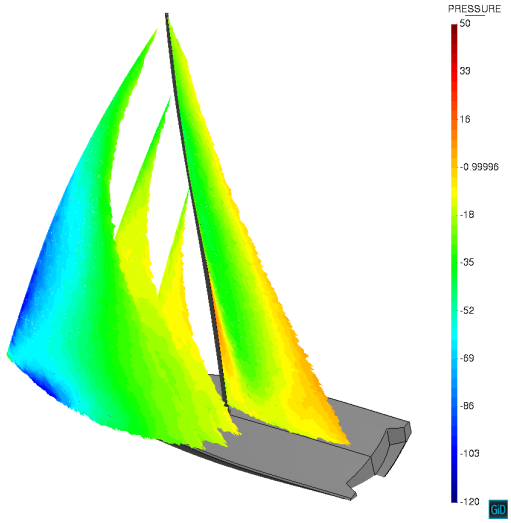
Complementary, the ability of the compressibility term in the mass conservation equation to keep the pressure bounded is proved by solving one example involving isolated fluid cavities with no Neumann BC. These results are crucial for the application of the method either to ill-conditioned level set functions coming from "dirty" input geometries or to complex moving boundary problems.

The experimental campaign is completed with two examples involving high Re turbulent flows. While the first one involves a simple 2D geometry, the second one is a feasible industrial application that evinces the utility of the method to completely get rid of the pre-processing difficulties that arise when trying to create a volume mesh from a thin-walled structure. In these two examples the viscous Navier-slip approach is proved to be a reasonable alternative to the more expensive boundary layer mesh plus no-slip condition combination.

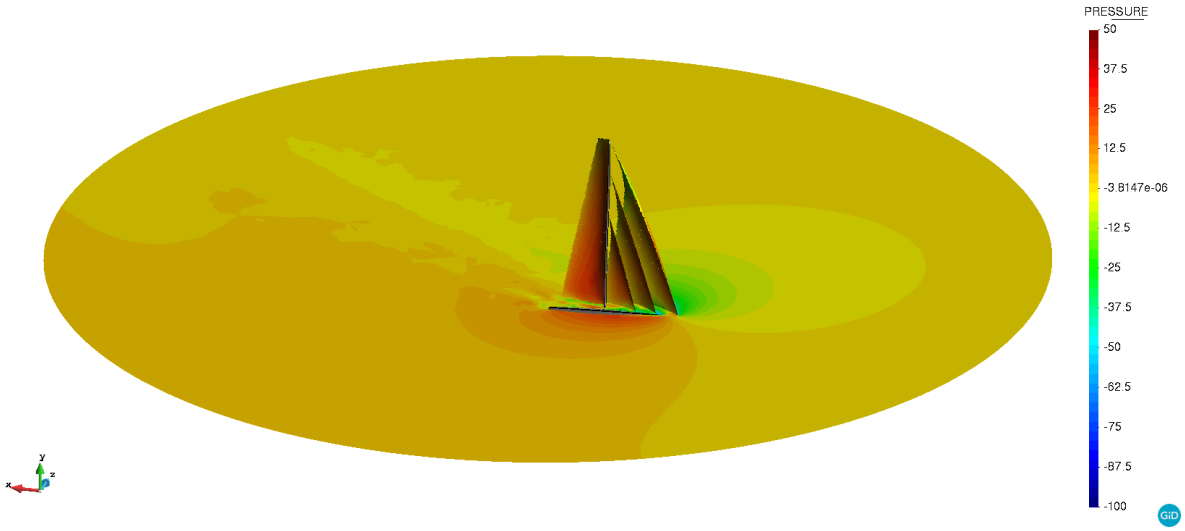
Finally, it is interesting to point out the further enhancements and arising work lines. The first one is to couple the proposed tool with a structural mechanics solver to build a FSI framework for the analysis of highly flexible membrane and shell structures. Another interesting application in the biomedical engineering context is the modelling of the fluid flow around thin biological tissues [122]. There is also room for more theoretical advances. In this regard, the method could be enhanced to work in combination with a discontinuous edge-based level set function. This would help to handle some of the partial element intersection issues reported in the sailboat example.



(a) Boat detail (windward).



(b) Boat detail (leeward).



(c) Computational domain cross section.

Figure 5.30: 3D flow around a sailboat. Pressure field [Pa].

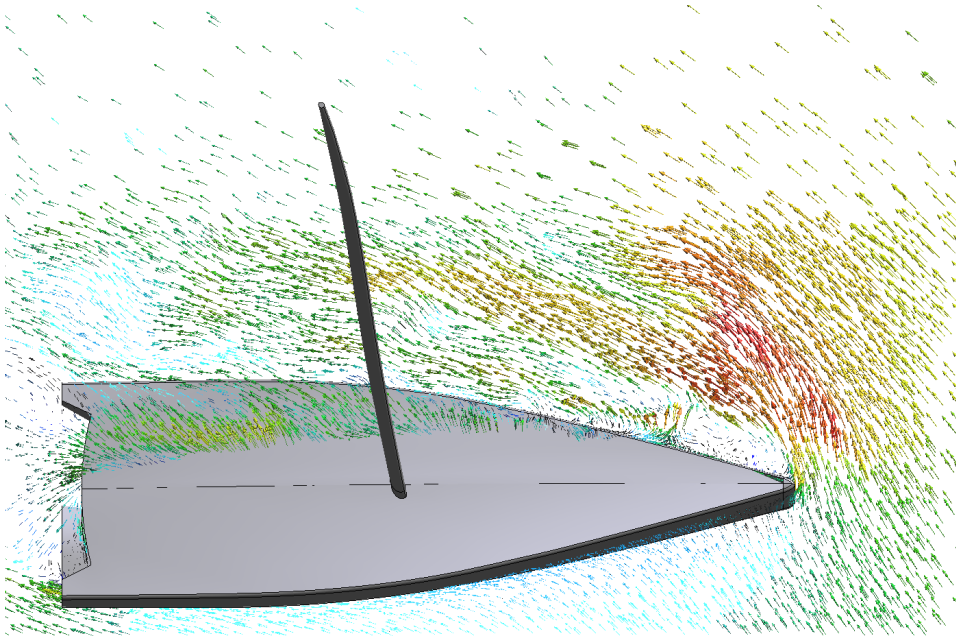


Figure 5.31: 3D flow around a sailboat. Velocity field cross section at height 1m [m/s].

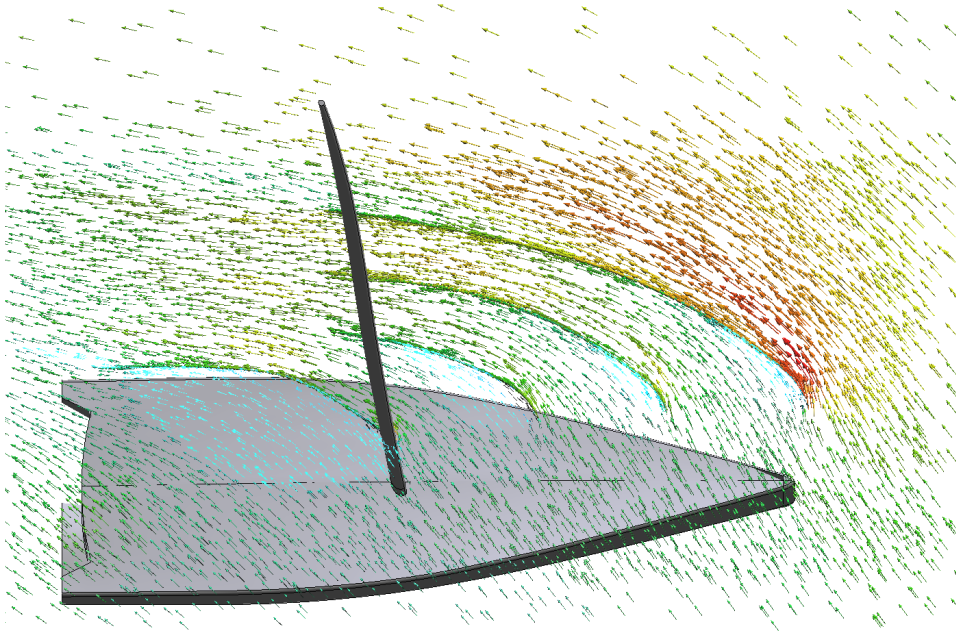


Figure 5.32: 3D flow around a sailboat. Velocity field cross section at height 5m [m/s].

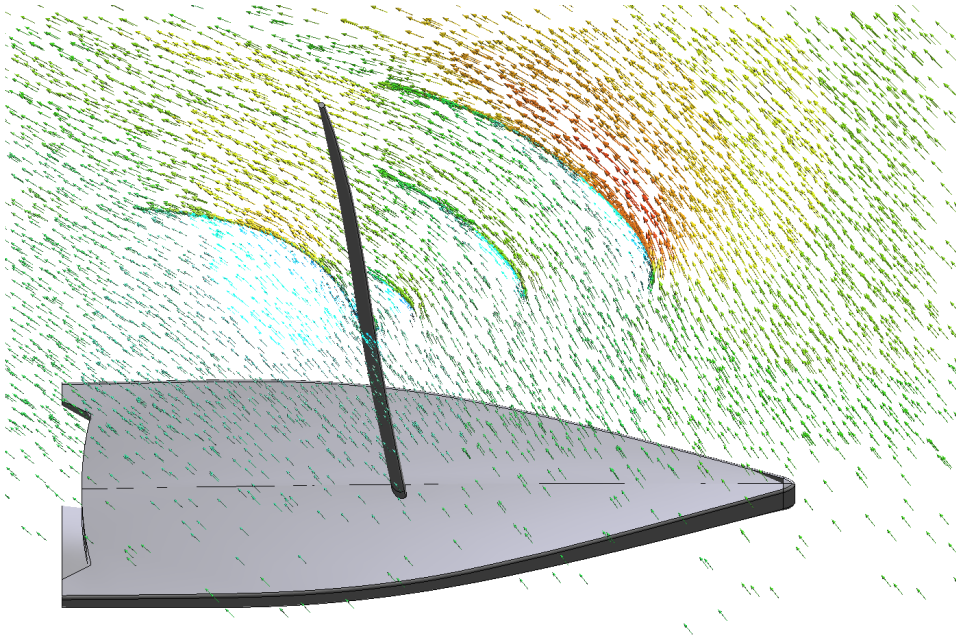


Figure 5.33: 3D flow around a sailboat. Velocity field cross section at height 12.5m [m/s].

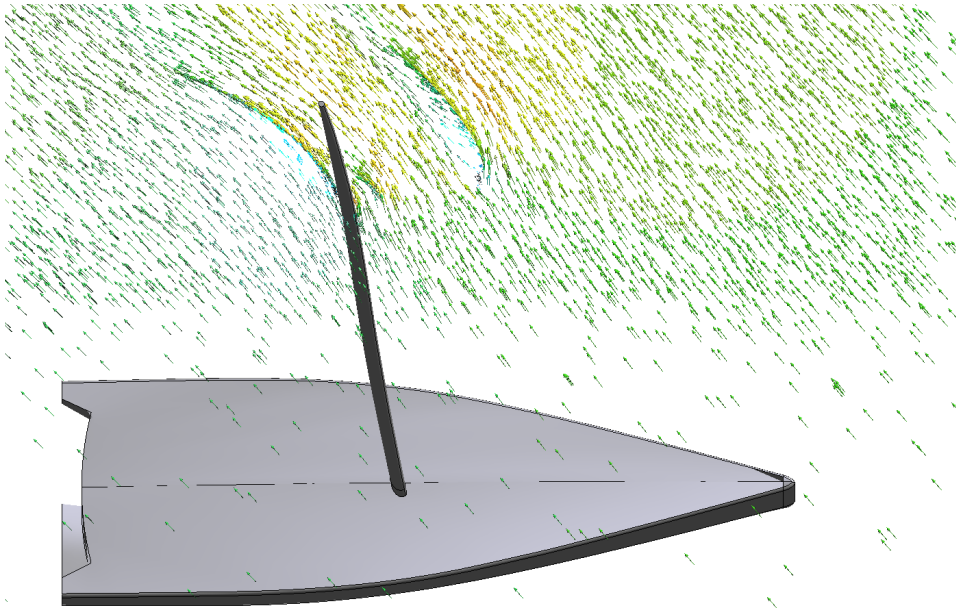


Figure 5.34: 3D flow around a sailboat. Velocity field cross section at height 20m [m/s].

Chapter 6

Computational modeling of the fluid flow in type B aortic dissection using a modified Finite Element embedded formulation

6.1 Article data

Title: Computational modeling of the fluid flow in type B aortic dissection using a modified Finite Element embedded formulation
Authors: R. Zorrilla, E.Soudah and R. Rossi
Journal: Biomechanics and Modelling in Mechanobiology (2020)
Received: 2 September 2019 / Accepted: 14 January 2020 / Available online: 23 January 2020
DOI: 10.1007/s10237-020-01291-x

6.2 Scientific contribution

This chapter presents the second paper of the compendium. The objective of the article is to further validate the formulation developed in Chapter 5 by solving the flow in a type B Aortic Dissection (AD) phantom model. Although this application is far from

the civil engineering field, it has been chosen since the data from previous numerical simulations and *in vitro* experiments is available. Furthermore, this article also aims at applying the methods developed in this thesis to other engineering disciplines. More specifically, this paper proves that the proposed discontinuous embedded formulation can be efficiently employed for the resolution of CFD problems involving thin biological tissues.

After a brief revision of the state of the art about CFD modeling of the AD disease, the article presents the *in vitro* experiment that is reproduced. The original *in vitro* experiment presents four variants of the phantom model [92], which differ in the number and size of the tears in the intimal flap. The results show that the embedded solutions obtained with the formulation in Chapter 5 are in really good agreement with both the experimental data in [92] as well as with the body fitted CFD simulation results presented in [102].

Additionally, the article also describes a simple procedure that might help the clinicians in the decision making before the AD fenestration surgery. Although the idea might seem rather simple, it can serve as a complementary tool to assess which is the optimal position for the fenestration re-entry tears when used in combination with patient specific geometries. Finally, it is important to highlight that the findings of this chapter can be considered as a first step towards the application of the proposed techniques to more complex biomedical problems involving thin-walled bodies such as the FSI analysis of hearth valves.

Biomechanics and Modeling in Mechanobiology
https://doi.org/10.1007/s10237-020-01291-x

ORIGINAL PAPER

Computational modeling of the fluid flow in type B aortic dissection using a modified finite element embedded formulation

Rubén Zorrilla[1,2,3] · **Eduardo Soudah**[1,2] · **Riccardo Rossi**[1,2]

Received: 2 September 2019 / Accepted: 14 January 2020

Abstract
This work explores the use of an embedded computational fluid dynamics method to study the type B aortic dissection. The use of the proposed technique makes it possible to easily test different intimal flap configurations without any need of remeshing. To validate the presented methodology, we take as reference test case an in vitro experiment present in the literature. This experiment, which considers several intimal flap tear configurations (number, size and location), mimics the blood flow in a real type B aortic dissection. We prove the correctness and suitability of the presented approach by comparing the pressure values and waveform. The obtained results exhibit a remarkable similarity with the experimental reference data. Complementary, we present a feasible surgical application of the presented computer method. The aim is to help the clinicians in the decision making before the type B aortic dissection surgical fenestration. The capabilities of the proposed technique are exploited to efficiently create artificial reentry tear configurations. We highlight that only the radius and center of the reentry tear need to be specified by the clinicians, without any need to modify neither the model geometry nor the mesh. The obtained computational surgical fenestration results are in line with the medical observations in similar clinical studies.

Keywords Aortic dissection · Surgical fenestration · Phantom model · CFD · Embedded mesh methods · Level set methods · Bio-mechanics · Open source

1 Introduction

The aortic dissection (AD) is a cardiovascular disease that results from the tearing of the inner layer of the aortic wall. The blood flow emerging through the tear causes the dissection of the inner (intima) and middle (media) layers of the aorta. The AD is a very severe pathology which often becomes fatal when the blood-filled channel silently ruptures through the outer (adventitia) aortic wall. An early diagnosis and aortic wall resistance assessment are therefore required to prevent the AD to reach such catastrophic scenario.

ADs are classified according to the location of the dissection. Hence, those ADs located in either the ascending region of the aorta or the aortic arch are denoted as ascending ADs or type A ADs. On the contrary, those ADs located in the descending region of the aorta are denoted as descending ADs or type B ADs. Type A ADs typically require surgical interventions, while type B ADs commonly become in a chronic disease with a high long-term morbidity and mortality rates due to the recurrence, the progressive dilatation of lumen and the aortic rupture (Fattori et al. 2011). Type B ADs survival rate is around the 50–80% at 5 years and diminishes to the 30–60% at 10 years. This decrease in the survival expectation is associated with the progressive dilatation of the AD, which commonly has multiple causes. Some of these are the number, location and size of the intimal tears, the elasticity of the intimal flap (IF), the pressure difference between the true (TL) and false lumen (FL) or the high intraluminal pressure (Hartnell and Gates 2005) that may result in the rupture of the aortic tissue (Tolenaar et al. 2013).

Due to the severe, and possibly lethal, nature of the AD disease, it is crucial to quantify the pressure peaks, as well as to locate the points where the dissection tear jets impact. Unfortunately, it is very challenging to clinically measure

✉ Rubén Zorrilla
rzorrilla@cimne.upc.edu

[1] Centre Internacional de Mètodes Numèrics en Enginyeria, CIMNE, Barcelona, Spain

[2] Universitat Politècnica de Catalunya, UPC, Barcelona, Spain

[3] Present Address: CIMNE, C. Gran Capità s/n, Edifici B0, Campus Nord, UPC, 08034 Barcelona, Spain

Published online: 23 January 2020

these magnitudes as it requires the use of invasive techniques. In this context, numerical simulation and ex vivo experiments can play an important role in the study of the chronic AD evolution in order to prevent its possible lethal outcomes (Nauta et al. 2015). The hemodynamics of the AD disease have been extensively studied using either ex vivo models (Faure et al. 2014; Qing et al. 2012) or experimental phantoms (Tsai et al. 2008; Rudenick et al. 2013; Peelukhana et al. 2017; Marconi et al. 2017). However, numerical methods have been proved to be a valuable tool to assess the hemodynamics in AD phantom models (Soudah et al. 2015; Ben Ahmed et al. 2016) or to simulate real patient-specific AD scenarios (Wan AB Naim et al. 2014; Alimohammadi et al. 2015; Tse et al. 2011; Chen et al. 2016; Rudenick et al. 2010a; Ryzhakov et al. 2019). The majority of these works only consider the fluid dynamics problem, excluding the elastic effects in both the aortic walls and the IF (Soudah et al. 2015; Ben Ahmed et al. 2016; Tse et al. 2011).

A natural extension of these works is therefore to consider the IF as a flexible body, leading to the resolution of the AD as a coupled FSI problem. Despite the fact that this approach has been successfully applied in Alimohammadi et al. (2015), Chen et al. (2016) and Ryzhakov et al. (2019), its extension to real medical applications is limited as it is hardly possible to obtain, or even estimate, the mechanical properties of patient-specific IF tissues.

However, it is nowadays possible to efficiently provide a patient-specific aortic wall displacement field by using medical imaging techniques as the 2D and 4D-PC-MRI (Sherrah et al. 2017; Liu et al. 2018; Dillon-Murphy et al. 2016). This information can be used as input data to model the AD as a moving boundaries CFD problem. This means that the IF (and aortic wall) displacements are considered as boundary conditions (BCs) of the CFD problem, whose solution is thus influenced by the 'a priori' known boundary movement. The common approach to handle boundary movements is to use an arbitrary Lagrangian–Eulerian (ALE) framework (Hirt et al. 1974; Donea et al. 2004). Despite the ALE is possibly the most straightforward approach in these cases, it is known to suffer from large element distortion, or even inversion, when large boundary movements are involved (Bhardwaj and Mittal 2012).

It is also due consideration that the IF is an extremely thin wall with possibly space varying thickness. Hence, it makes sense to model the IF as a negligible thickness body. Unfortunately, dealing with such zero-thickness structures is particularly challenging from the preprocessing point of view as they require to duplicate their surface mesh in order to consider the 'positive' and 'negative' pressure sides. Moreover, the mesh generation in these cases is prone to fail as the polygonal representation of the surface mesh may intersect one of its duplicated counterpart due to the machine precision range operations.

Considering all the above-mentioned reasons, we decided to model the AD using an embedded mesh method, meaning that the negligible thickness IF is implicitly described by a level set function. This simplification implies that the entire AD volume of fluid is meshed disregarding the IF geometry, skipping thus all the previously mentioned preprocessing complexities (e.g., node and surface duplication) (Löhner et al. 2008). After the AD fluid volume has been meshed (background mesh), the IF geometry (skin) is superimposed to it to compute the level set function that implicitly represents it.

One of the main advantages of this technique is that there is no necessity to solve any mesh problem if the spatial configuration of the IF evolves in time. The unique requirement is thus to recompute the level set function according to the new IF configuration, which is assumed to be known from a position-time data set. This feature opens the possibility to easily model the flow of patient-specific AD scenarios with moving IF as it is nowadays straightforward to obtain the time evolution of the IF by using noninvasive medical imaging techniques such as the 4D-PC-MRI (François et al. 2013).

An extra motivation to use the presented embedded approach in the AD modeling scenario is the possibility to simplify the fluid flow assessment during the decision making before the fenestration surgical intervention. Aortic fenestration is a method to decrease high pressures in the hypertensive FL. This treatment is performed by creating an artificial tear in the IF with the objective of making possible the return flow from the FL to the TL. This change in the IF geometry reduces the FL intraluminal pressure, decreasing thus the dilatation risk of the FL. The use of the embedded approach offers the possibility of testing as many reentry tears as needed by only updating the level set function that describes it, requiring no modification nor remeshing in the background mesh. Thanks to this feature of the presented method, the new modified IF models can be straightforwardly set up.

This work is the first part of a new research line that aims to exploit embedded mesh methods for the more efficient assessment of the AD hemodynamics. One of the objectives of the present work is to prove that the AD CFD problem can be alternatively solved using embedded mesh methods, being their solution comparable to traditional body conforming (also known as body fitted) approaches in terms of accuracy and efficiency. To that purpose, we take as reference the work described in Rudenick et al. (2013) and Iwai et al. (1991), where several in vitro AD experiments are reproduced. It is due mentioning that there exists previous studies (Soudah et al. 2015; Ben Ahmed et al. 2016; Rudenick et al. 2010b) that also reproduced the in vitro experiments in Rudenick et al. (2013) which we take as reference. In all these publications the authors used an in-silico body

conforming CFD numerical approach. It is important to remark that in these in-silico experiments, the IF was considered to be a rigid body. This assumption explains the minor differences than can be appreciated when these results are compared with the experimental ones in Rudenick et al. (2013). The fact that the our reference works consider the IF as a rigid body as well as the absence of the IF displacement data in Rudenick et al. (2013) lead us to also take the rigid IF assumption.

Complementary, this work also aims to exploit the capability to easily generate different IF geometries to study surgical fenestration technique. Thus, we also present a handy method to artificially generate (and assess) the reentry tears by only setting the center and the radius of the new fenestration tears.

The paper is organized as follows. Section 2 describes the main concepts of the level set approach and presents the governing equations of the embedded CFD model used for the simulation of the AD. Next, in Sect. 3, geometry, fluid properties and boundary conditions corresponding to the considered AD cases are presented. In Sect. 4.1, the most significant results of the numerical simulations are summarized and compared with the reference data. In Sect. 4.2, we study the influence on the hemodynamics of a hypothetical AD surgical fenestration. Finally, Sect. 5 collects the main achievements and conclusions of the work and states the future work lines.

All the cases presented in this work are solved using Kratos Multiphysics, an Open Source C++ object oriented finite element framework for the resolution of multiphysics problems (Dadvand et al. 2010, 2013).

2 Formulation

2.1 Level set-based embedded approach

The most distinguishable feature of any embedded (or immersed) approach is the capability to decouple the geometry of the analyzed objects from the problem domain mesh. As opposite to the traditional body conforming (also known as body-fitted) approaches, these kind of formulations do not solve the problem using a discrete representation of the analyzed bodies but using a mathematical representation of them, which is typically achieved by using a level set technique.

The level set method, which was firstly proposed by Osher and Fedkiew in Osher and Fedkiw (2003), makes it possible to implicitly describe physical discontinuities occurring in a background mesh. This method has been successfully applied in a vast amount of engineering problems, ranging from the representation of the fluid interface in two-fluid (Sussman et al. 1994; Codina and Soto 2002) and free-surface (Rossi et al. 2013) problems to the tracking of crack paths in mechanical ones (Gravouil et al. 2002).

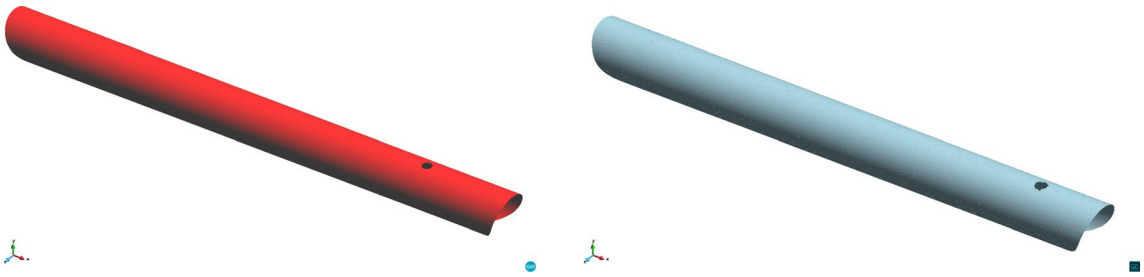
In our case, the level set method is used to represent the flow discontinuity generated by the presence of the IF between the TL and the FL. One of the advantages of this approach is that the model and mesh generation are incredibly simplified. Such simplification comes from the fact that the TL and FL are treated as a unique volume of fluid, disregarding the presence of the intimal flap, which is treated in a completely independent manner during the entire preprocess stage. This makes it possible to set up and test as many IF configurations as needed by only updating the IF geometry, without even remeshing the volume TL and FL domain. This feature is particularly interesting for the decision making before surgery since it makes possible to easily set up new IF configurations to assess their affectation to the blood flow.

The level set approach is also useful to get rid of the mesh update problems when the problem is solved considering the movement of the IF. Unlike in the FSI case, in which the aortic walls movement is a consequence of the fluid flow, in this case, the objective is to obtain the fluid flow but considering a known IF movement, which can be easily obtained by using medical imaging techniques. Taking into account the uncertainties associated with the IF thickness and mechanical behavior, this becomes a feasible robust alternative to the fully coupled FSI simulation of the AD.

Unfortunately, all the previous advantages come at the price of introducing some extra complexities that are not present in the common body-fitted approaches. On the one hand, it is required to use a robust auxiliary algorithm (Baumgärtner et al. 2018) to compute the level set function in accordance with the IF position. Besides, although the level set a robust and always defined operation, some details of the original geometry (e.g., sharp edges or corners) might be lost when computing its level set approximation.

With regard to the continuity of the level set function, it is possible to distinguish between two types of level set functions depending on the nature of the embedded body. If the geometry to be analyzed has a closed internal volume, a continuous level set function is commonly employed, although it is possible to use a discontinuous one. On the contrary, the bodies without internal volume, such as the IF, can only be represented by a discontinuous level set (Fig. 1). This means that in these cases the level set function is computed element by element, which implies that the same node can have different sign and values depending on the element considered. A much more detailed discussion about the continuity of the level set can be found in Zorrilla et al. (2019).

Once the embedded body (in this case the IF) is represented by a level set function, it is needed to impose proper boundary conditions over it. According to the nature of such boundary condition imposition, we can roughly divide the unfitted mesh methods in two families: immersed and



(a) Origin geometry **(b)** Level set intersections

Fig. 1 Intimal flap level set function representation

embedded mesh methods. Since the distinction between these two typologies of methods appears to be blurred in the literature, from now on, we will denote as immersed those formulations that use the nodes in the interior of the level set to impose the analyzed body movement. On the contrary, we denote as embedded those formulations that impose the boundary condition by using a variational (weak) technique over the zero isosurface of the level set. Considering that these works aim to represent the IF as a zero-thickness body, it is only possible to impose the boundary condition by using an embedded mesh method. The technique used for such boundary condition imposition is detailed in Sect. 2.3.

In the following subsections, we present the governing equations of the fluid dynamics problem as well as the level set boundary condition imposition technique.

2.2 Governing equations

The governing equations of the fluid dynamics problem are the conservation of linear momentum and the conservation of mass. In an Eulerian reference framework, these equations are

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} - \boldsymbol{\nabla}\cdot\boldsymbol{\sigma} = \mathbf{f}_{\text{ext}} \tag{1a}$$

$$\frac{D\rho}{Dt} + \rho\boldsymbol{\nabla}\cdot\mathbf{v} = 0 \tag{1b}$$

where $\mathbf{v}$ is the velocity, $\mathbf{f}_{\text{ext}}$ is an external volume force (e.g., gravity) and ρ is the density. $\boldsymbol{\sigma}$ denote the Cauchy stress tensor which is computed as $\boldsymbol{\sigma} = -p\mathbf{I} + \mathbb{C} : \boldsymbol{\nabla}^{\text{s}}\mathbf{v}$, where p is the pressure, $\boldsymbol{\nabla}^{\text{s}}$ the symmetric gradient operator and $\mathbb{C}$ the constitutive tensor describing the viscous (shear) behavior.

Taking into account that the perfusion liquid used in the in vitro experiments reproduced in this work has similar properties to the water ones, we decided to model it as a viscous Newtonian fluid. Hence, the viscous constitutive tensor $\mathbb{C}$ in Eq. 1a can be substituted by the Newtonian one. The Newtonian constitutive tensor only depends on the dynamic viscosity μ and can be expressed in Voigt notation as the second-order tensor

$$\mathbb{C} = \begin{bmatrix} \frac{4\mu}{3} & -\frac{2\mu}{3} & -\frac{2\mu}{3} & 0 & 0 & 0 \\ -\frac{2\mu}{3} & \frac{4\mu}{3} & -\frac{2\mu}{3} & 0 & 0 & 0 \\ -\frac{2\mu}{3} & -\frac{2\mu}{3} & \frac{4\mu}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \tag{2}$$

By substituting the previous definition of $\boldsymbol{\sigma}$ into the linear momentum conservation equation (Eq. 1a), it yields the well-known viscous Navier–Stokes equations

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} - \boldsymbol{\nabla}\cdot(\mathbb{C} : \boldsymbol{\nabla}^{\text{s}}\mathbf{v}) + \boldsymbol{\nabla}p = \mathbf{f}_{\text{ext}} \tag{3a}$$

$$\frac{D\rho}{Dt} + \rho\boldsymbol{\nabla}\cdot\mathbf{v} = 0 \tag{3b}$$

For purely incompressible fluids, it is commonly assumed that the density time derivative $D\rho/Dt$ in Eq. 1b is zero, leading to the conventional viscous incompressible Navier–Stokes equation. This assumption is however not taken into account in here. Instead, we consider the simplified state equation $p = \rho c^2$, leading to the so-called viscous pseudo-compressible Navier–Stokes equations. This simplified equation of state relates the pressure with the density through the fluid speed of sound c. The purpose of adding this pseudo-compressibility is to tackle those cases in which isolated fluid cavities with no Neumann boundary condition appear. The apparition of such ill-defined cavities is commonly related either to dirty input skin geometries or to the movement of the embedded skin. If we further assume that the density fluctuations are negligible, the final form of the governing equations reads

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v}\cdot\boldsymbol{\nabla})\mathbf{v} - \boldsymbol{\nabla}\cdot(\mathbb{C} : \boldsymbol{\nabla}^{s}\mathbf{v}) + \boldsymbol{\nabla}p = \mathbf{f}_{\text{ext}} \tag{4a}$$

$$\frac{1}{\rho c^2}\frac{\partial p}{\partial t} + \boldsymbol{\nabla}\cdot\mathbf{v} = 0 \tag{4b}$$

In all the cases presented in this work, we set the speed of sound c to 10^{12} m/s. This implies that the term $1/\rho c$ in Eq. 4b is reduced beyond the machine precision (10^{-16}), recovering thus the fully incompressible form of the Navier–Stokes equations. This is intentionally done since no isolated fluid cavities are expected due to the simplicity and steady nature of the level set function that represents the IF. However, it is due mentioning that this term might play an important role as soon as the problem is extended to consider the movement of the IF.

2.3 Modified finite element formulation and embedded imposition

Once the immersed body (IF) is represented by the level set function, proper boundary conditions need to be imposed over it. To that purpose, we rely on the use of a modified finite element (FE) space together with a Nitsche-based general Navier imposition.

On the one hand, we substitute the standard FE space by an alternative discontinuous one in those elements intersected by the level set. Such alternative space (henceforth named Ausas FE space after the name of the original author) was firstly proposed in Ausas et al. (2010). The main feature of the Ausas FE space is the capability to represent discontinuities inside an element. This makes it possible to capture the velocity and pressure fields discontinuities that arise between the TL and the FL due to the presence of the IF. Unfortunately, such capability comes at the price of having worse interpolation properties than the common FE space. Despite this limitation, the capability of the Ausas FE space to solve viscous incompressible fluid problems, ranging from the low to the high Reynolds (Re) scenario, is proved in Zorrilla et al. (2019).

On the other hand, a Robin-type general Navier condition (Eq. 5) is imposed over the level set intersections by using a Nitsche-based technique as it is described in Winter et al. (2018). The general Navier condition is composed by a Dirichlet no penetration condition in the normal direction to the level set zero isosurface (Eq. 5a) and a Robin condition to impose a shear force in the tangential direction to the level set zero isosurface (Eq. 5b). The normal and tangential contributions of the general Navier condition can be split in two equations as

$$(\mathbf{v} - \bar{\mathbf{v}})\mathbf{P}^{n} = \mathbf{0} \tag{5a}$$

$$\left(\varepsilon\left(\left[\mathbb{C} : (\boldsymbol{\nabla}^{s}\mathbf{v})\right]\cdot\mathbf{n} - \mathbf{h}\right) + \mu(\mathbf{v} - \bar{\mathbf{v}})\right)\mathbf{P}^{t} = \mathbf{0} \tag{5b}$$

where $\bar{\mathbf{v}}$ is the immersed body (IF) velocity, $\mathbf{n}$ is the unit normal vector to the level set and $\mathbf{h}$ an imposed shear traction. $\mathbf{P}^{n}$ and $\mathbf{P}^{t}$ are the normal and tangential projection operators obtained as $\mathbf{P}^{n} = \mathbf{n}\otimes\mathbf{n}$ and $\mathbf{P}^{t} = \mathbf{I} - \mathbf{n}\otimes\mathbf{n}$. ε is the so-called slip length, which regulates the tangential behavior of the boundary condition. Therefore, the general Navier condition behaves as a slip condition (no tangential velocity constraint) as $\varepsilon \to \infty$, while it approximates to a stick condition (null tangential velocity) as $\varepsilon \to 0$.

More details on the implementation, convergence and stability analysis of the general Navier condition and the Nitsche constraint can be found in Winter et al. (2018).

3 Experiment: aortic dissection model

3.1 Geometry

As one of the main objectives of this work is to reproduce the in vitro experiments in Rudenick et al. (2013), the phantom models used in this work are taken as reference geometry for the present study. Far from being patient-specific AD cases, these phantom models consist in a simplified geometry made of silicon and latex connected to a closed circuit that pumped water into it to emulate the AD blood flow. The geometry of these AD phantom models is simplified to two straight and constant diameter tubes (the aortic arch is not taken into account) whose thickness is assumed to be infinitesimal. The IF is also considered to be an infinitesimal thin wall (Figs. 2, 3). The two straight tubes axes are aligned with the X-axis of the global coordinates system, whose origin is located in the center of the inlet face.

As was originally done in Rudenick et al. (2013), we distinguish the different study cases according to the proximal and distal tears diameter. Table 1 collects each one of the

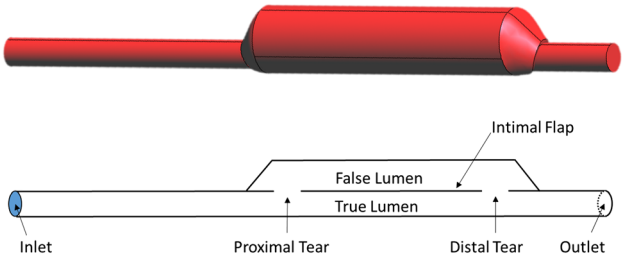
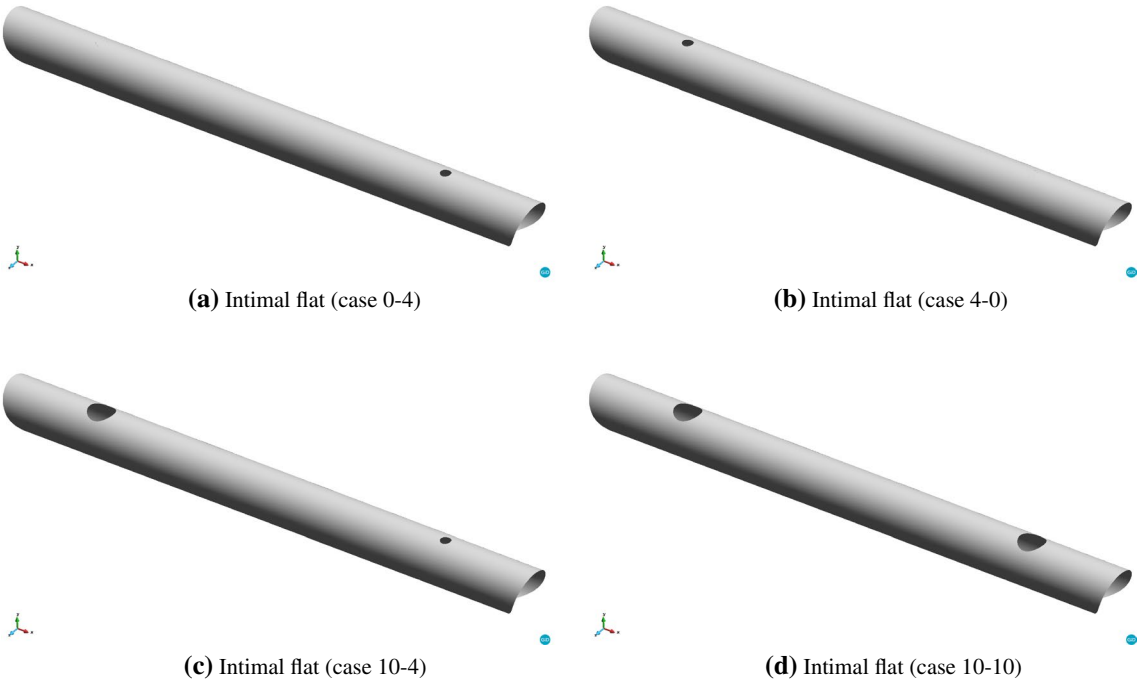


Fig. 2 Three-dimensional geometry model of the type B aortic dissection and longitudinal cross-sectional plane. The diameter of the tube representing the TL is 16 mm, while its length is 390 mm. On the other hand, the diameter of the tube representing the FL is 37 mm. The dissection length is 160 mm, and it is set up by aligning the inferior parts of the TL and FL tubes. The proximal and distal tears were simulated by creating two holes, which diameters range from 0 to 10 mm, in the FL tube. The center of the proximal and distal tears is 177.5 mm and 322.5 mm from the inlet



(a) Intimal flat (case 0-4)

(b) Intimal flat (case 4-0)

(c) Intimal flat (case 10-4)

(d) Intimal flat (case 10-10)

Fig. 3 Intimal flat geometries for the four validation cases studied. These are the input geometries for the computation of the level set function used in the resolution of the embedded CFD problems

study cases together with their proximal and distal tears diameter. Henceforth, we will refer to the cases as Case *proximal tear diameter—distal tear diameter* (e.g., Case 0–4 stands for no proximal tear and 4 mm distal tear diameter).

It is worth mentioning that thanks to the embedded approach used in this work the geometry and the mesh generation are significantly simplified compared with the common body-fitted approach. The key advantage is that a unique volume mesh needs to be generated for all the cases to be studied. This volume mesh is built from the outer skin, which at the same time is conformed by the aggregation of the inlet and outlet tubes skin together with the enlarged tube one representing the FL. Once the volume mesh is generated, each simulation case can be defined by simply dropping the surface mesh representing its IF into the volume mesh and computing its level set function, which is obtained from the intersections between the AD volume mesh and IF surface one.

We would like to stress again the potential of the presented method to be exploited when multiple IF configurations need to be efficiently tested (e.g., decision making before surgery). We also highlight that the complex preprocess operations commonly associated with the CFD simulation of volumeless geometries (e.g., edge and surface duplication) are completely avoided as the volume mesh is meshed with no regard of the immersed volumeless structures.

3.2 Material properties

The perfusion liquid used in the original in vitro experiments is water at room temperature, which is less viscous than blood. According to the original authors, no noticeable differences in the velocity values can be observed if they are compared with the ones obtained by using a glycerin-water solution, whose viscosity and density are similar to blood ones (Rudenick et al. 2013). Taking this into consideration, we decided to set the fluid material properties to be similar to the blood ones. Therefore, the density is set to 1050 kg/m^3 , while the dynamic viscosity is set to $3.5 \cdot 10^{-3}$ kg/ms.

3.3 Boundary conditions

The boundary conditions are selected to emulate each one of the reference in vitro experiments in Rudenick et al. (2013), meaning that each AD scenario has its own velocity and pressure waveform. On the one hand, the fluid flow generated by the pumping system of the in vitro experiments is reproduced by setting the inlet velocity function $v_M(t)$. On the other hand, the outlet pressure waveform $p_M(t)$ is taken directly from experiments measurements and imposed in the outlet region. Both inlet velocity function and outlet pressure waveform are directly extracted from the reference in vitro experiments datasets.

The above-mentioned inlet velocity function v_M is applied to the inlet surface as a parabolic profile in such a way that the reference in vitro experiments flow rate is kept. Thus, the velocity in each one of the inlet nodes can be computed as

$$\mathbf{v}^{\text{in}}(r,t) = 2.0 v_M(t)\left(1 - \frac{r}{R^2}\right)\mathbf{n}_{\mathbf{in}} \tag{6}$$

where r denotes the in-plane distance from the inlet center point to the considered inlet node, R the radius of the inlet surface, whose value is 0.008 m, and $\mathbf{n}_{\mathbf{in}}$ the vector normal to the inlet surface pointing in the inward direction.

The outlet pressure is fixed to the $p_M(t)$ waveform values. To fulfill the force equilibrium in the outlet region, it is also required to set the external traction that corresponds to the $p_M(t)$ values. This can be computed as $\mathbf{t} = p_M(t)\mathbf{n}$.

We consider that the rest of boundary surfaces that conform the AD computational volume have a stick (no-slip) behavior, something that turns into a null velocity constraint. The same no-slip behavior is assumed in the embedded IF. Therefore, the slip length (ε) in Eq. 5b is set to 0 in all the cases.

3.4 Simulation settings

The problem is discretized in time using a second-order backward differentiation formula (BDF2). The available data sets in Rudenick et al. (2013) comprehend a period of 0.88 s, which is enough to represent a normal hemodynamic cycle of the human aorta: the peak pressure (reached at 0.25 s) and a biphasic diastolic period. Based on this, we set the total simulation time to 0.8 s, which is also enough to represent all the aortic cycle features, and the time step of $2.5 \cdot 10^{-3}$ s.

The use of the BDF2 formula results in an implicit time discretization, which preferred because of its well-known stability properties. However, such enhanced stability comes at the price of requiring to solve a nonlinear system of equations at each time step. This is achieved by using a Newton–Raphson strategy, with linearized convective velocity, in combination with a mixed pressure and velocity convergence criterion. The relative and absolute tolerances are 10^{-4} and 10^{-6} for both the pressure and velocity residuals.

As it is highlighted above, one of the remarkable features of the presented approach is that the same background mesh is always used with no regard of the analyzed IF configuration. Therefore, a unique volume mesh conformed by approximately 7M linear tetrahedrons is used in all the presented cases. All the simulations are run in our internal high-performance computing (HPC) facilities (Acuario cluster) using a parallel distributed memory environment (MPI parallelism). For all the cases, we take four nodes with 16 Intel Xeon E5-2670 processors and 64 GB of memory. This amounts a total of 64 processors and 256 GB of memory. The elapsed wall clock time in all cases is less than 12 h.

4 Results

4.1 Validation

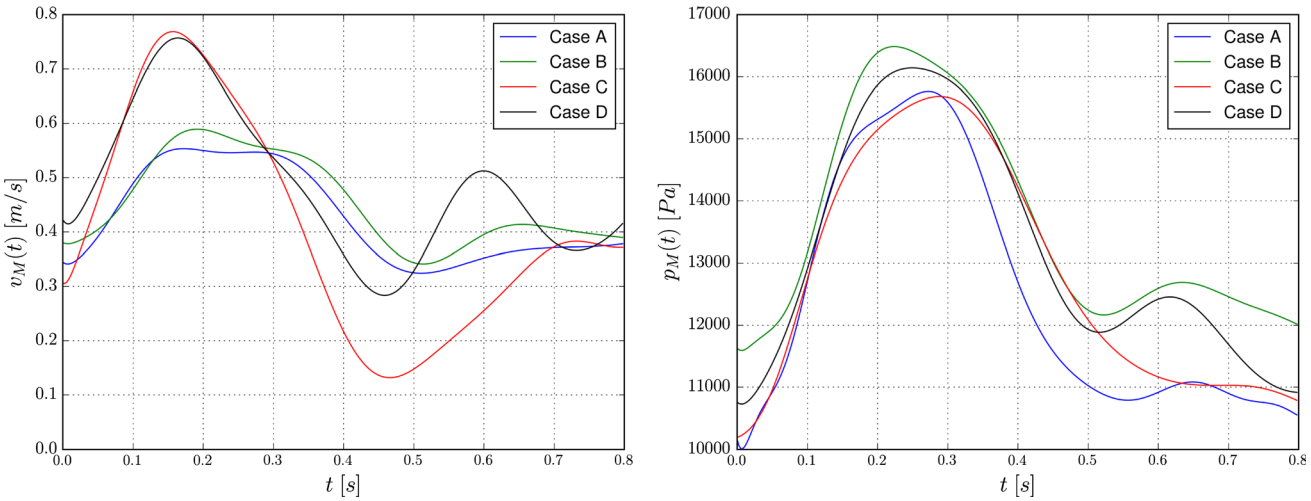
This section presents the results of the AD scenarios described in the previous section. In all the cases, we compare the results obtained by using the presented embedded approach with the results obtained from the reference in vitro experiments in Rudenick et al. (2013). The AD cases with a single tear (case 4–0 and case 0–4) are also validated with the CFD simulation results in Soudah et al. (2015). The results validation mainly consists in comparing the pressure time evolution in the four control points stated in Rudenick et al. (2013) and Soudah et al. (2015). The coordinates of these pressure measurement points are collected in Table 2.

4.1.1 1 Tear cases

For assessing the effect of the tears number upon the fluid flow, next we consider a configuration with a unique tear: Case 4–0 (A) and Case 0–4 (B). In both cases, the proximal or distal tear has a diameter of 4 mm. The inlet and outlet boundary conditions are identical to the ones used in the in vitro experiments (Fig. 4). Before any further discussion, it is interesting to remark that the in vitro pressure measurements were done using retrograde catheterization. This could have induced a partial obstruction of the tears, reducing thus their effective diameter, something that could have altered FL pressure measurements (especially in the small tear cases). Furthermore, it is important to mention that in these particular cases, the experimental data measurements in the FL region appear to have been incorrectly taken (Fig. 3a, b). Taking this into consideration, we decided to also use the numerical results in Soudah et al. (2015) to properly validate these AD scenarios.

Figure 5 shows the pressure evolution at the reference points for case 4–0. Figure 6 presents the same results for case 0–4. As it can be observed, pressures at the TL (proximal and distal locations) are nearly identical in both cases. For both case 4–0 and case 0–4, a similar pressure waveform can be observed. However, the peak pressure reported in Soudah et al. (2015) is higher than the result obtained in this work. Considering that the presented results better fit the experimental ones, we believe that these differences are associated with the different discretization levels (1.5 M elements vs 7 M) as well as to the IF geometry modeling. Moreover, in this work, the IF is considered to have an infinitesimal thickness, while in Soudah et al. (2015), the thickness is 2 mm.

It is also interesting to remark that single-tear models show near zero FL flow (Figs. 7, 8). In both cases, there is



(a) Inlet velocity (m/s)

(b) Outlet pressure (Pa)

Fig. 4 Inlet velocity and outlet pressure waveform used in the four AD validation cases. On the left side, the inlet velocity time evolution is shown. On the right one, the outlet pressure time evolution is shown. These data have been extracted from reference in vitro experiments in Rudenick et al. (2013)

Table 1 Cases studied

Case	Case Id.	Proximal tear		Distal tear	
		Diameter (mm)	Coordinates (mm)	Diameter (mm)	Coordinates (mm)
Case 4–0	A	4	(177.5, 9.0, 0.0)	0	(No tear)
Case 0–4	B	0	(No tear)	4	(320.5, 9.0, 0.0)
Case 10–4	C	10	(177.5, 9.0, 0.0)	4	(320.5, 9.0, 0.0)
Case 10–10	D	10	(177.5, 9.0, 0.0)	10	(320.5, 9.0, 0.0)

Table 2 Control points coordinates

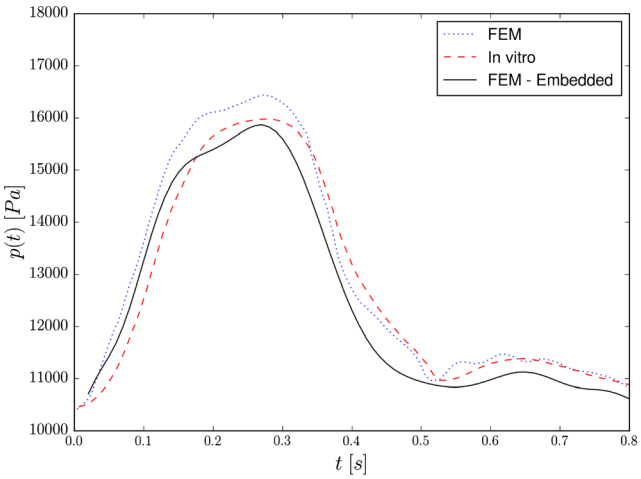
	Proximal tear		Distal tear	
	True lumen	False lumen	True lumen	False lumen)
Coordinates (mm)	(177.5, 0.0, 0.0)	(177.5, 17.5, 0.0)	(320.5, 0.0, 0.0)	(320.5, 17.5, 0.0)

only a small jet entering into the FL, while a slightly wider jet in the opposite direction enters into the TL during the peak systole. At peak diastole, it can be observed that the flow behaves in the opposite manner. In these cases, we observe large velocity decreases in the FL, leading to an negligible flow rate. This low fluid velocity in the FL favors the thrombus formation in single-tear AD as it is observed in the daily clinical routines.

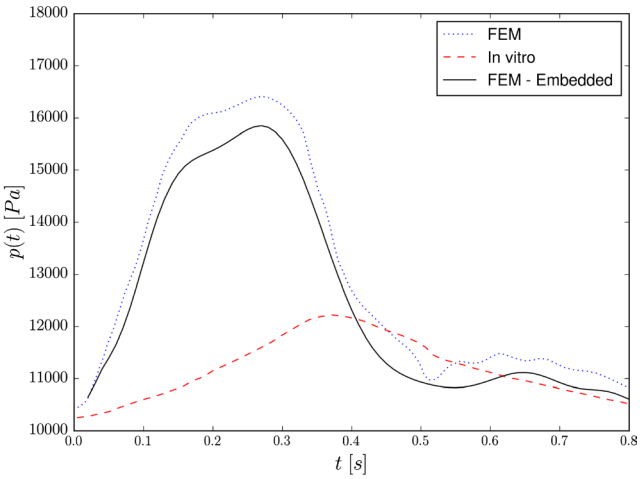
4.1.2 2 Tears cases

In this section, we analyze the double (proximal and distal) tear configuration scenarios: Case 10–4 (C) and Case 10–10 (D). Case 10–4 involves a proximal tear of 10 mm and a distal tear of 4 mm, while Case 10–10 involves two tears of equal diameter (10 mm). The inlet and outlet boundary conditions are identical to the ones used in the reference in vitro experiments (Fig. 4).

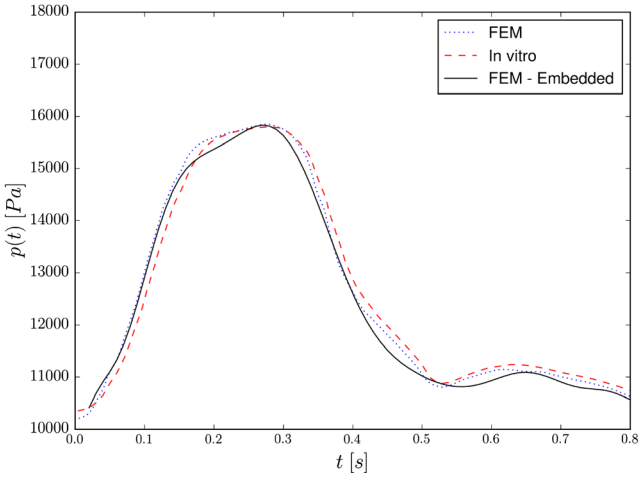
Again, we compare the waveform and values of the pressure recorded in the different control points with the in vitro experiments reference ones (Figs. 9, 10). With regard to the pressure waveforms, bare differences can be observed between the numerical and experimental approaches. Concerning the pressure values, specially the systolic peak ones, the TL and FL obtained values are apparently equal to the reference ones in the Case 10–10. However, it can be observed that the TL and FL systolic peak values in the Case 10–4 are slightly lower than the ones reported in Rudenick et al. (2013), being the maximum error around the 5% (FL proximal tear control point). These differences are possibly



(a) Case A - TL proximal pressure



(b) Case A - FL proximal pressure



(c) Case A - TL distal pressure

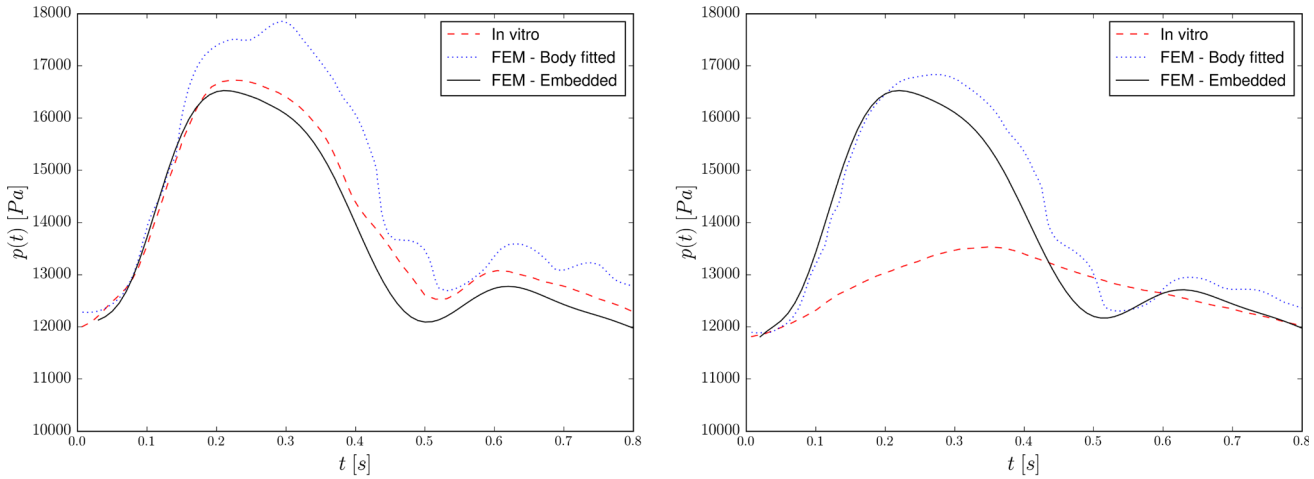
Fig. 5 Case A. Pressure evolution at various locations. Comparison with the results in Rudenick et al. (2013) and Soudah et al. (2015)

associated with the fact that our model considers the IF as a rigid body, while it was flexible in the original in vitro experiments. A similar trend in the results can be also observed in Soudah et al. (2015) and Ben Ahmed et al. (2016).

The equal pressure evolution in the control points proves that the presence of a distal tear prevents large pressure gradients between FL and TL to occur. We also observe that large distal tears increase the pressure in the FL, leading to a diminution in the pressure gradient within the AD. It is also interesting to comment that the presence of two tears in the IF yields a flow recirculation in the FL, which is around a 20% higher in the Case 10–10 if we compare to the small distal tear Case 10–4. Therefore, during the systole phase, the flow enters from the TL to the FL through the proximal tear, while it leaves the AD through the distal one. Conversely, the opposite behavior occurs during the diastole phase. A natural consequence of such recirculation is that the flow in the FL becomes more chaotic, leading thus to the generation of some observable vorticity effects. Such increment in the vorticity is known to increase the wall shear stress, which is directly related to the redissection, or even rupture, risk.

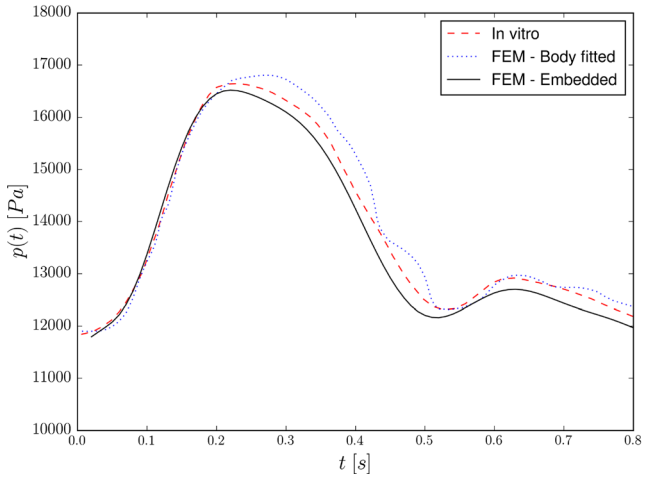
4.2 Surgical application: reentry tear aortic fenestration

In this section, we want to show the capability of the presented methodology to easily generate new computational models to study the AD surgical fenestration technique. Surgical aortic fenestration represents an effective alternative for the treatment of ischemic complications of the AD, particularly in those patients with no aortic dilatation. This surgical procedure is less invasive than the thoracic endovascular aortic repair (TEVAR) (Nauta et al. 2017) as

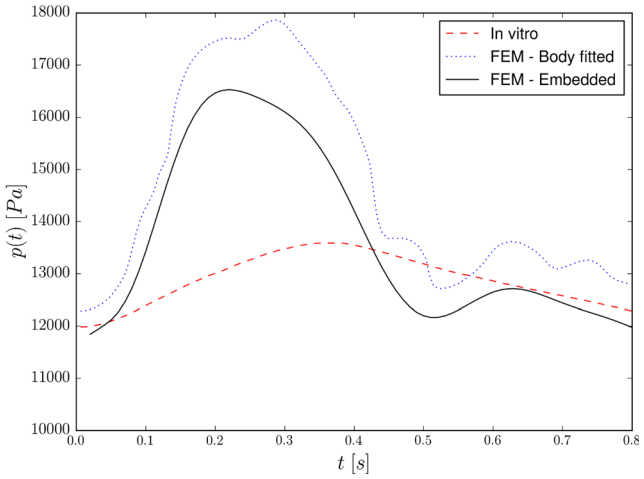


(a) Case B - TL proximal pressure

(b) Case B - FL proximal pressure



(c) Case B - TL distal pressure



(d) Case B - FL distal pressure

Fig. 6 Case B. Pressure evolution at various locations. Comparison with the results in Rudenick et al. (2013) and Soudah et al. (2015)

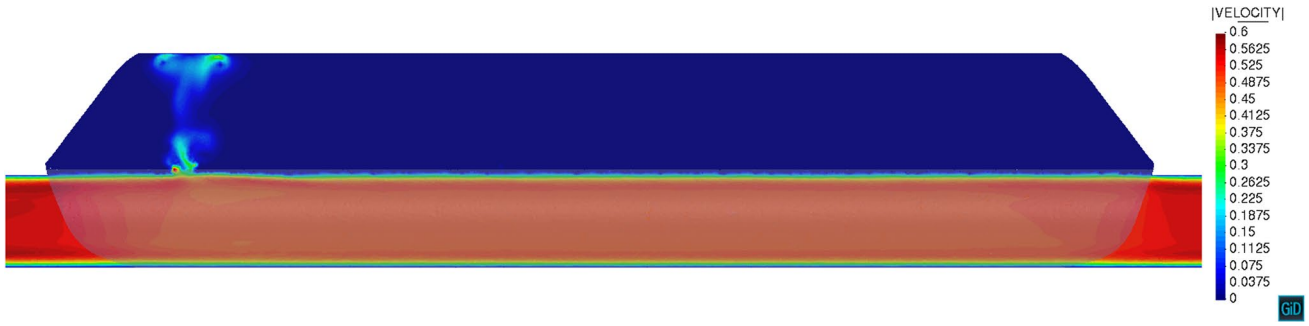
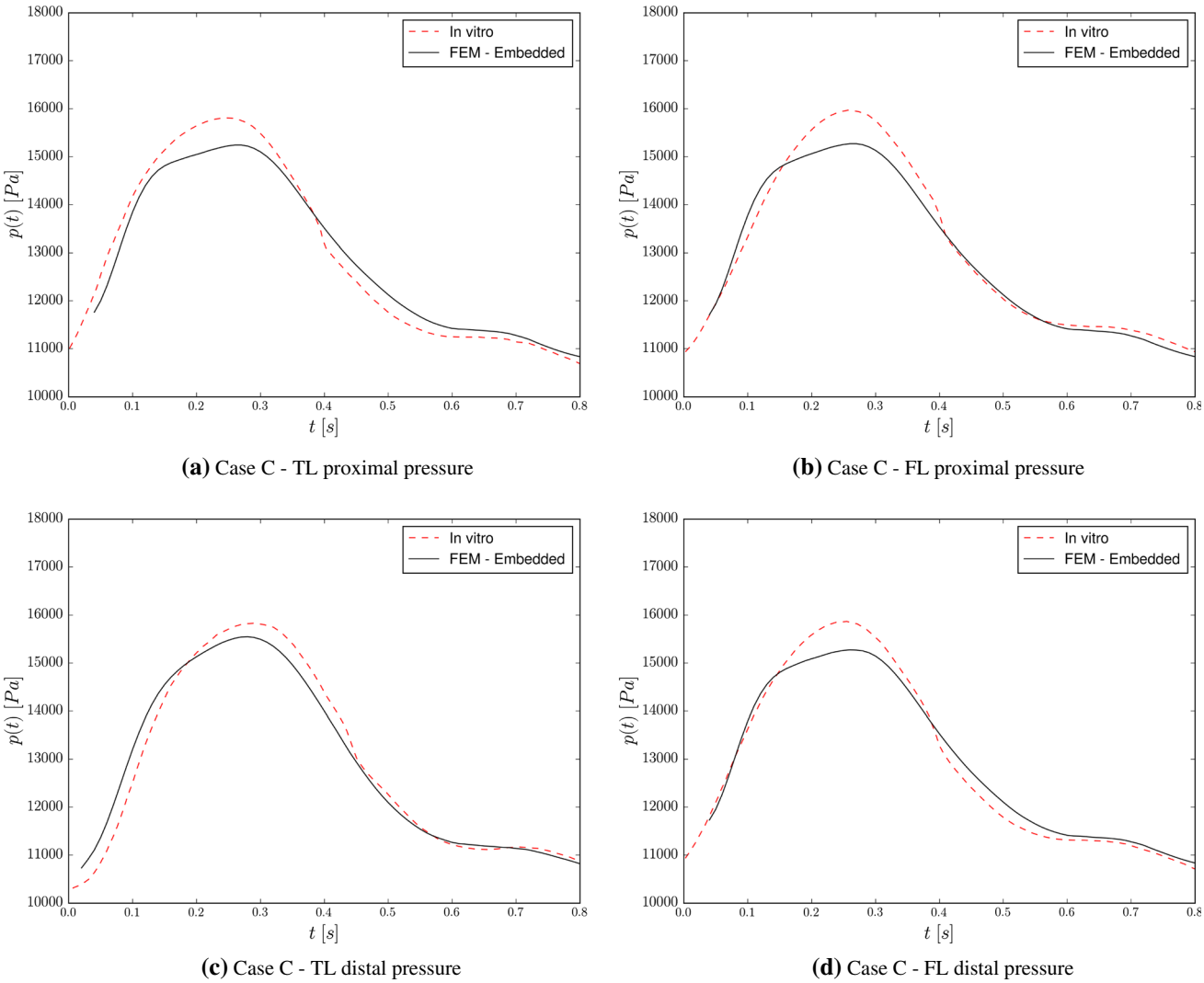


Fig. 7 Case A. Velocity cross section at $t = 0.09$ s. IF level set representation shadowed over the cross section

it only requires a puncture of the IF from the TL to the FL using a needle. This technique corrects the malperfusion at the suprarenal or infrarenal levels by creating a single aortic lumen intended to restore the organ flow. The aortic fenestration is also used to decompress the hypertensive FL thanks to the creation of the new reentry tear in the distal part of the IF (Hartnell and Gates 2005; Canchi et al. 2018). Considering that the mortality rate of patients without

Fig. 8 Case B. Velocity cross section at $t = 0.03$ s. IF level set representation shadowed over the cross section



(a) Case C - TL proximal pressure

(b) Case C - FL proximal pressure

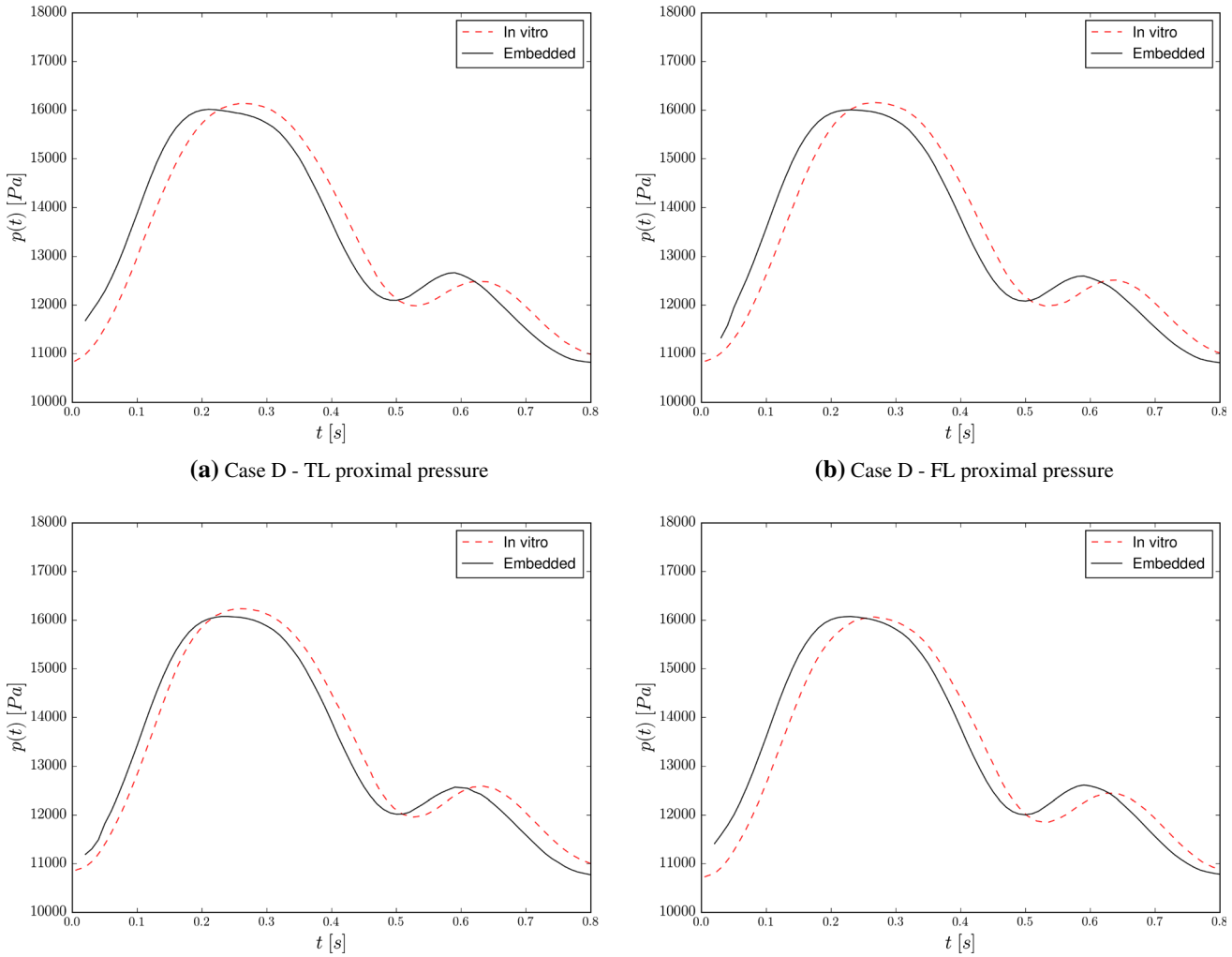
(c) Case C - TL distal pressure

(d) Case C - FL distal pressure

Fig. 9 Case C. Pressure evolution at various locations. Comparison with the results in Rudenick et al. (2013)

fenestration is higher than the one in patients with exit or reentry tears, Tsai et al. (2008), evinces the importance of having noninvasive tools to a priori assess the consequences of the surgical fenestration before the clinical intervention.

We took case A as application case. Starting from the original IF geometry, we generate an artificial reentry tear in the midpoint of the IF top line (Table 3) to assess the



(a) Case D - TL proximal pressure

(b) Case D - FL proximal pressure

(c) Case D - TL distal pressure

(d) Case D - FL distal pressure

Fig. 10 Case D. Pressure evolution at various locations. Comparison with the results in Rudenick et al. (2013)

Table 3 Fenestration cases studied

Case	Case Id.	Proximal tear	Mid tear(s)	Distal tear
Case 4–10–0	A_F	(177.5, 9.0, 0.0)	(249.0, 9.0, 0.0)	(No tear)

affectation of the surgical fenestration to the velocity and pressure fields.

4.2.1 Embedded reentry tear generation

First of all, it is important to highlight that the creation of the new reentry tear is automatically done without any geometrical operation. The methodology implemented to create the new re-tear is described in algorithm 1. This algorithm only requires the user to introduce the radius (r^{fen}) and center ($\mathbf{x}_{\mathbf{c,fen}}$) of the new reentry tear. After the definition of these magnitudes, the routine flags the IF skin elements lying inside the fenestration radius. Once the marked elements are identified, they are deactivated to recompute the level set function with this new modified skin mesh. We would like to highlight again the simplicity of the methodology from the end-user perspective, as it avoids any preprocessing operation (including remeshing) thanks to the capabilities of the embedded approach. Besides, the presented approach has no limitation in the number of reentry tears, so could be possible to consider more than one.

ALGORITHM 1

Algorithm to place a fenestration tear in the intimal flap mesh.

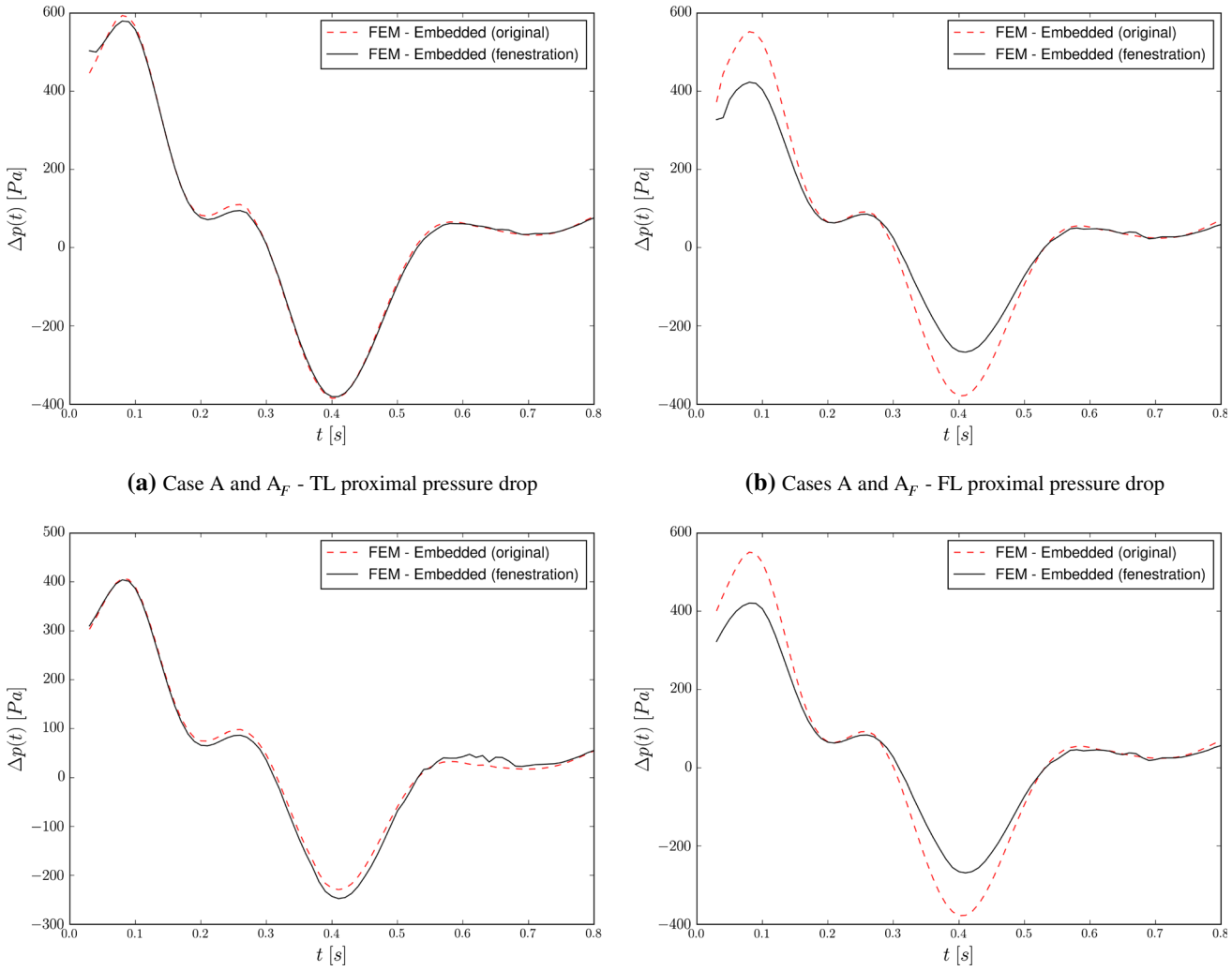
```
r_fen                                                  ▷ Fenestration radius
x_c,fen                                                ▷ Fenestration center point
fluid_mesh                                             ▷ Fluid mesh container
if_skin_mesh                                           ▷ Intimal flap mesh container

for if_skin_mesh.Elements() do
    x_c,elem = if_element.Center()                     ▷ Intimal flap element center
    dist = ||x_c,fen − x_c,elem||                      ▷ Distance from x_c,fen to x_c,elem
    if dist ≤ r_fen then                               ▷ Check if the current entity belongs to the fenestration
        if_element.Set(TO_ERASE)
    end if
end for

if_skin.RemoveElements(TO_ERASE)                       ▷ Remove the selected intimal flap entities
CalculateDiscontinuousDistance(fluid_mesh,if_skin_mesh) ▷ Calculate the level set to the intimal flap mesh
```

4.2.2 Embedded surgical fenestration results

Figure 11 compares the results obtained after the surgical fenestration (case A_F) with the original ones with no fenestration (case A). Prior to any assessment, it is due mentioning that in this case, the results are not presented for the absolute pressure but for the pressure drop of the control points. Since we assume identical outlet pressure values in both cases (with and without fenestration), the pressure drop is computed as the pressure difference between the control points and the outlet.

Table 4 shows the maximum pressure differences at the proximal, mid (new reentry tear) and distal points for case A and A_F. Even though the surgical fenestration seems to barely affect the TL pressure evolution, Fig. 11 clearly shows that the surgical fenestration reduces the systole and diastole pressure peak values in the FL (difference between the TL and FL pressure values) (Fig. 12). Figure 12 also proves that the pressure difference between the TL and FL is almost negligible in the midpoint. Moreover, the pressure difference between the TL and FL decreases 120 Pa in the distal region of the AD. These results are in line with the medical protocols, which have traditionally promoted the use of surgical fenestration in one tear scenarios.

However, the creation of the distal reentry tear increases the flow rate in the FL (Fig. 13c). This artificially generated flow pattern complicates the generation of the thrombus in the AD. We also observe that the pressure difference grows in the proximal region. This is possibly due to the fact that the flow between the TL and FL increases, with the consequent increase in the energy local losses of the proximal tear. Clinical findings increasingly suggest, that patients with a partially thrombosed FL are at a higher risk of rupture (Tsai et al. 2007). Conversely, completely thrombosed FL have a better prognosis (Song et al. 2011). Indeed, it has been suggested that 'complete thrombosis of the residual FL might be a sign of aortic wall healing and remodeling' (Tsai et al. 2008). Therefore, we can conclude that the FL artificial flow pattern generated after the aortic fenestration is directly related to the thrombosis of FL (Yazdani et al. 2018).



(a) Case A and A_F - TL proximal pressure drop

(b) Cases A and A_F - FL proximal pressure drop

(c) Cases A and A_F - TL midpoint pressure drop

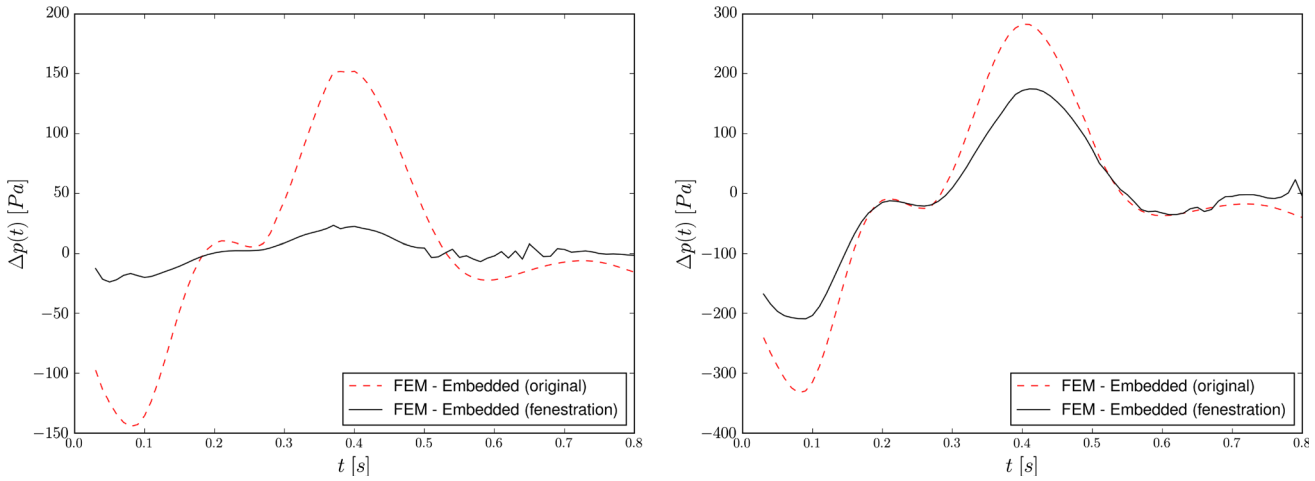
(d) Cases A and A_F - FL midpoint pressure drop

Fig. 11 Case A_F. Pressure drop evolution at various locations. Comparison with Case A (before fenestration) pressure drop results

5 Conclusion

In this work, we propose a new numerical FE-based embedded technique to study the fluid flow in biomedical applications, in particular to AD scenarios. The FE embedded technique is developed and implemented by the authors in the Kratos Multiphysics framework (Dadvand et al. 2010). The presented method is validated by reproducing the in vitro experiment described in Rudenick et al. (2013). After the successful validation, and with the objective of showing the advantages of the proposed approach, we reproduced a surgical fenestration technique. The presented embedded technique makes it possible to easily modify the IF geometry without the necessity of modifying the volume mesh.

It is known that pressure differences between TL and FL may indicate conditions of AD growth (Tolenaar et al. 2013). Taking this into consideration, the validation is based on measuring the pressures in a set of control points in the TL and FL. After solving multiple cases, we dare say that the



(a) Midpoint **(b)** Distal point

Fig. 12 Pressure difference between TL and FL. Positive values means higher pressure in TL, whereas a negative value means higher pressure in FL

presented methodology is capable to reliably represent the fluid flow inside an AD. We also notice that the maximum pressure peaks occur for the single-tear cases. This leads us to consider the introduction of a new reentry tear in the IF, mimicking a surgical fenestration intervention. Such new tear introduction yields a diminution in the TL and FL pressure difference as shown in Fig. 12. The comparison between the two tear cases (C and D) shows that a small distal tear reduces the pressure difference. On the contrary, large distal tears result in negligible pressure difference between the TL and FL. This observation confirms that a small tear size yields an insufficient reduction in the FL pressure. These results are in line with the conclusions in Berguer et al. (2015), which state that the pressure difference between the TL and the FL is inversely proportional to the cross-sectional area of the exit tear. It can be also observed that the tear jet becomes relevant only in those cases involving two tears (Figs. 14, 15). These fluid jets affect the FL wall and could eventually generate a high wall stress. Furthermore, these cases also show that a small distal tear induces a reduced jet impact in the proximal region of the FL (Fig. 14). A detailed analysis of the flow through the tears reveals that the jets are typically shifted in the direction of the flow (downstream).

We would also like to highlight that all the AD cases have been solved with a unique volume mesh. This is thanks to the capability of the method to represent the IF by using a discontinuous level set function. Note that the level set movement could eventually be considered by simply updating the level set function. This feature becomes crucial for the application of the presented approach in the biomedical field since it would make it possible to solve a FSI problem without any knowledge of the mechanical properties of the tissue, whether it is healthy or not (Alimohammadi et al. 2015; Chen et al. 2016). In this particular case, the movement of the IF can be obtained using 4D PC-MRI techniques (Liu et al. 2018; Sherrah et al. 2017; Soudah et al. 2011). 4D-PC-MRI techniques can provide the characteristic features of the flow in the entire aortic volume for any specific time window (Dillon-Murphy et al. 2016). For each time frame, the displacement of the IF could be recorded and used as a boundary condition in our method. As a second step, the mechanical properties of the IF could be eventually estimated by performing an inverse analysis after computing the pressure distribution in the TL and FL. Although the presented approach is already prepared to be used in combination with any imposed movement of the IF, there are still some complexities to be tackled. These are mainly related with the input data acquirement. Handling these data could be also a issue as it is required to do, store and manage the IF segmentation, together with the tear(s) location.

The capability of the model to easily create artificial tears is also proved. Then, it could be possible to use the presented technique to efficiently study the flow pattern in AD patients after stent-grafting repair. The location of the stent-graft into

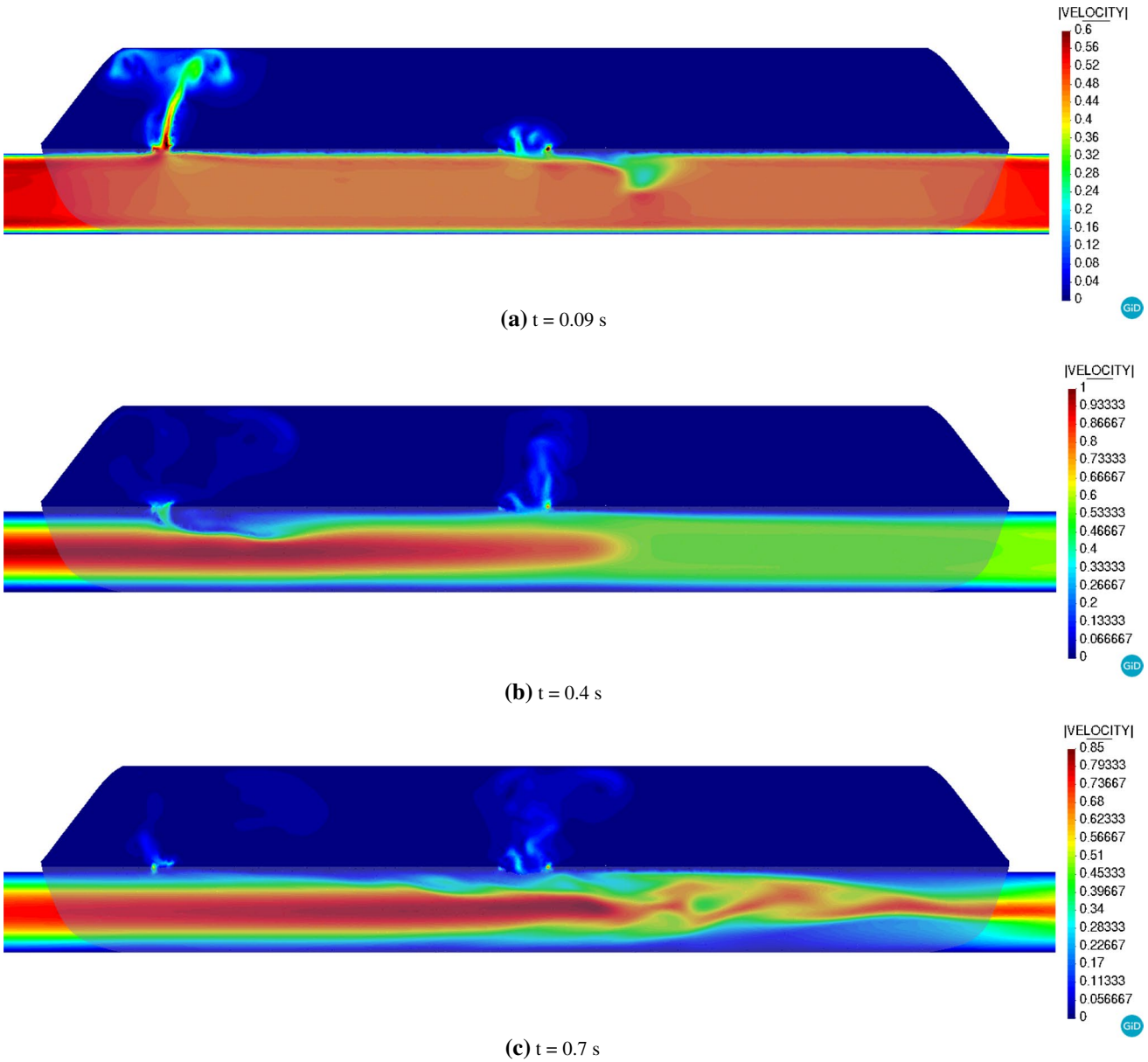


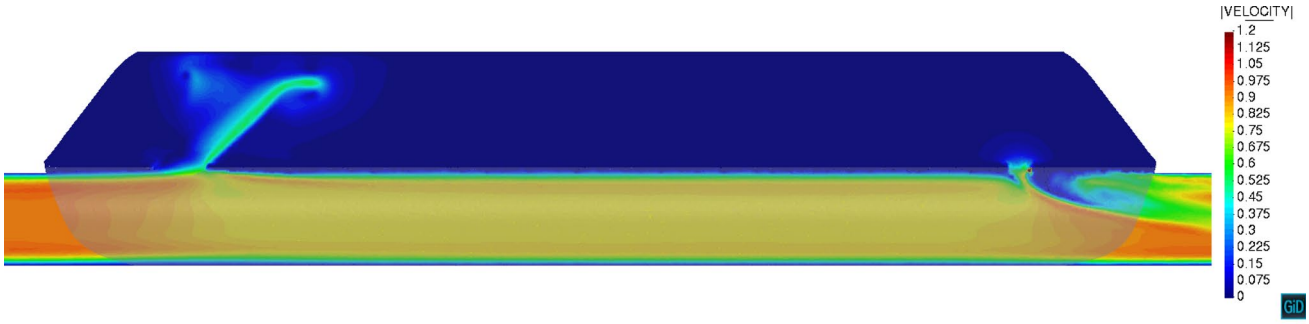
Fig. 13 Case A_F. Velocity cross section at different time steps. IF level set representation shadowed over the cross section

Table 4 Pressure difference between TL and FL control points. Positive values means higher pressure in TL, whereas a negative value means higher pressure in FL

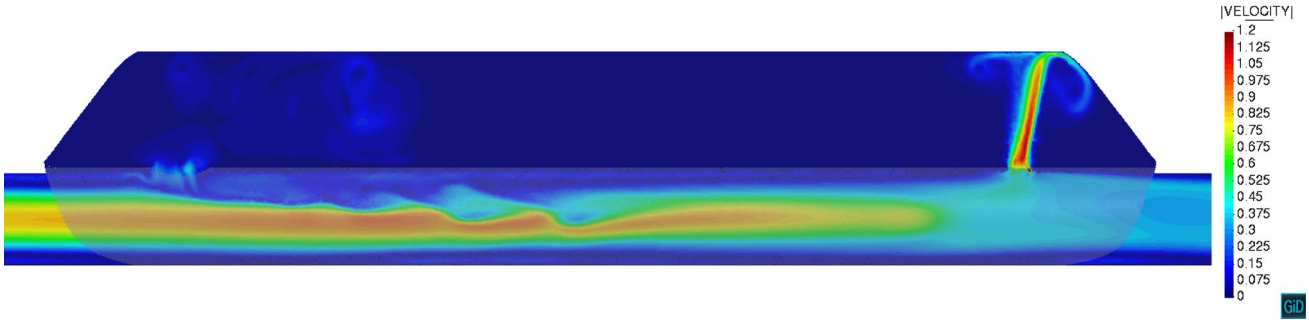
Case	Pressure Difference (P_{TL}-P_{FL})		
	Proximal (Pa)	Mid (Pa)	Distal (Pa)
Case 4–0	41.246	152.021	− 332.684
Case 4–10–0	167.937	23.774	− 209.312

the AD volume can be described by a level set function to study the new flow pattern distribution.

Finally, we would like to remark that this work is conceived as a preliminary validation and proof of concept of a new research line that aims to exploit the capabilities of fixed mesh methods in the framework of biomedical applications. In the short term, we plan to apply our methodology to study patient-specific data coming from 4D-PC-MRI and to extend it to consider the movement of the IF.



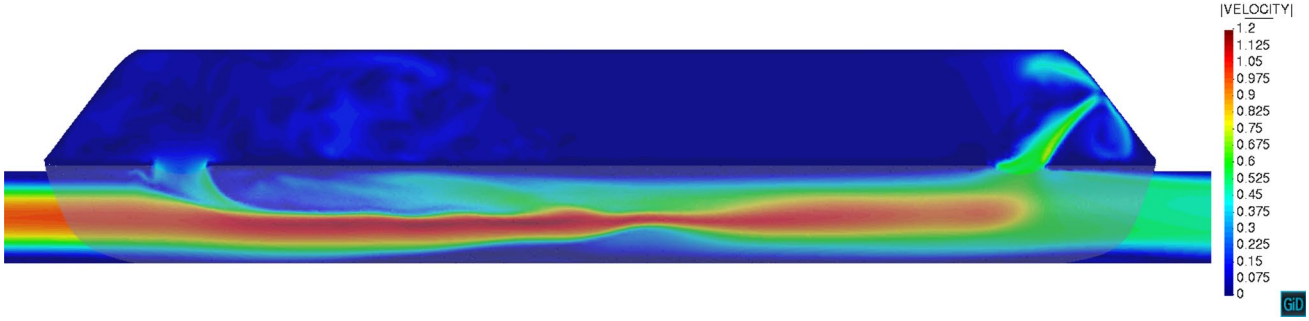
(a) t = 0.15 s



(b) t = 0.4 s

Fig. 14 Case C. Velocity cross section at different time steps. IF level set representation shadowed over the cross section

(a) t = 0.09 s



(b) t = 0.4 s

Fig. 15 Case D. Velocity cross section at different time steps. IF level set representation shadowed over the cross section

Acknowledgements This research has been partly supported by the European Commission (EC) through the Project ExaQUte (H2020-FETHPC-2016-2017-800898). Rubén Zorrilla gratefully acknowledges the support of the Spanish Government for his FPU Grant (FPU15/03796). Finally, the authors want to express their gratitude to Dr. Antonio J. Barros for his fruitful discussion about AD, Dr. Paula A. Rudenick and Dr. Bart H. Bijnens for providing the experimental data of the in vitro tests, as well as to the Spanish Research Network for Cardiac Computational Modeling (V-Heart-SN) for the interest expressed.

Author contributions Investigation was done by RZ, ES and RR. Model implementation was carried out by RZ. Geometrical modeling was done by ES. Data processing was carried out by RZ and ES. Validation was made by RZ and ES. Visualization was carried out by RZ and ES. Discussion was carried out by RZ, ES and RR. The paper was written by RZ, ES and RR.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

Alimohammadi M, Sherwood J, Karimpour M, Agu O, Balabani S, Díaz-Zuccarini V (2015) Aortic dissection simulation models for clinical support: fluid-structure interaction vs. rigid wall models. Biomed Eng Online 14(1):34

Ausas R, Sousa F, Buscaglia G (2010) An improved finite element space for discontinuous pressures. Comput Methods Appl Mech Eng 199:1019–1031. https://doi.org/10.1016/j.cma.2009.11.011

Baumgärtner D, Wolf J, Rossi R, Dadvand P, Wüchner R (2018) A robust algorithm for implicit description of immersed geometries within a background mesh. Adv Model Simul Eng Sci 5(1):21. https://doi.org/10.1186/s40323-018-0113-8

Ben Ahmed S, Dillon-Murphy D, Figueroa C (2016) Computational study of anatomical risk factors in idealized models of type B aortic dissection. Eur J Vasc Endovasc Surg 52(6):736–745

Berguer R, Parodi J, Schlicht M, Khanafer K (2015) Experimental and clinical evidence supporting septectomy in the primary treatment of acute type B thoracic aortic dissection. Ann Vasc Surg 29(2):167–173. https://doi.org/10.1016/j.avsg.2014.10.001

Bhardwaj R, Mittal R (2012) Benchmarking a coupled immersed-boundary-finite-element solver for large-scale flow-induced deformation. AIAA J 50(7):1638–1642. https://doi.org/10.2514/1.J051621

Canchi S, Guo X, Phillips M et al (2018) Role of re-entry tears on the dynamics of type B dissection flap. Ann Biomed Eng 46(1):186–196

Chen H, Peelukhana S, Berwick Z et al (2016) Fluid–structure interaction simulations of aortic dissection with Bench validation. J Vasc Surg 64(6):1892

Codina R, Soto O (2002) A numerical model to track two-fluid interfaces based on a stabilized finite element method and the level set technique. Int J Numer Methods Fluids 40:293–301. https://doi.org/10.1002/fld.277

Dadvand P, Rossi R, Gil M et al (2013) Migration of a generic multi-physics framework to HPC environments. Comput Fluids 80:301–309. https://doi.org/10.1016/j.compfluid.2012.02.004

Dadvand P, Rossi R, Oñate E (2010) An object-oriented environment for developing finite element codes for multi-disciplinary applications. Arch Comput Methods Eng 17(3):253–297. https://doi.org/10.1007/s11831-010-9045-2

Dillon-Murphy D, Noorani A, Nordsletten D, Figueroa CA (2016) Multi-modality image-based computational analysis of haemodynamics in aortic dissection. Biomech Model Mechanobiol 15(4):857–876

Donea J, Huerta A, Ponthot JP, Rodríguez-Ferran A (2004) Arbitrary Lagrangian–Eulerian methods, ch. 14. American Cancer Society, New York

Fattori R, Mineo G, Di Eusanio M (2011) Acute type B aortic dissection: current management strategies. Curr Opin Cardiol 26(6):488–493

Faure E, Canaud L, Cathala P, Serres I, C MA, Alric P (2014) Human ex-vivo model of Stanford type B aortic dissection. J Vasc Surg 60(3):767–775. https://doi.org/10.1016/j.jvs.2013.06.083

François CJ, Markl M, Schiebler ML et al (2013) Four-dimensional, flow-sensitive magnetic resonance imaging of blood flow patterns in thoracic aortic dissections. J Thorac Cardiovasc Surg 145(5):1359–1366

Gravouil A, Moës N, Belytschko T (2002) Non-planar 3D crack growth by the extended finite element and level sets—part II: level set update. Int J Numer Methods Eng 53(11):2569–2586. https://doi.org/10.1002/nme.430

Hartnell G, Gates J (2005) Aortic fenestration: a why, when, and how-to guide. Radiographics 25(1):175–189

Hirt C, Amsden A, Cook J (1974) An arbitrary Lagrangian–Eulerian computing method for all flow speeds. J Comput Phys 14(3):227–253. https://doi.org/10.1016/0021-9991(74)90051-5

Iwai F, Sostman HD, Evans AJ et al (1991) Cine phase-contrast magnetic resonance imaging for analysis of flow phenomena in experimental aortic dissection. Invest Radiol 26(12):1071–1078

Liu D, Fan Z, Li Y et al (2018) Quantitative study of abdominal blood flow patterns in patients with aortic dissection by 4-dimensional flow MRI. Sci Rep 8(1):9111. https://doi.org/10.1038/s41598-018-27249-9

Löhner R, Cebral JR, Camelli FE et al (2008) Adaptive embedded and immersed unstructured grid techniques. Comput Methods Appl Mech Eng 197(25–28):2173–2197

Marconi S, Lanzarone E, De Beaufort H, Conti M, Trimarchi S, Auricchio F (2017) A novel insight into the role of entry tears in type B aortic dissection: pressure measurements in an in vitro model. Int J Artif Organs 40(10):563–574

Nauta FJ, Bogerijen VGH, Conti M et al (2017) Impact of thoracic endovascular repair on pulsatile aortic strain in acute type B aortic dissection. Aorta 5(02):42–52

Nauta FJ, Conti M, Kamman AV et al (2015) Biomechanical changes after thoracic endovascular aortic repair in type B dissection: a systematic review. J Endovasc Ther 22(6):918–933

Osher S, Fedkiw R (2003) Level set methods and dynamic implicit surfaces. Applied mathematical sciences, vol 153, 1st edn. Springer, New York

Peelukhana S, Wang Y, Berwick Z et al (2017) Role of pulse pressure and geometry of primary entry tear in acute type B dissection propagation. Ann Biomed Eng 45(3):592–603. https://doi.org/10.1007/s10439-016-1705-4

Qing KX, Chan YC, Lau SF, Yiu WK, Ting ACW, Cheng SWK (2012) Ex-vivo haemodynamic models for the study of stanford type B aortic dissection in isolated porcine aorta. Eur J Vasc Endovasc Surg 44(4):399–405. https://doi.org/10.1016/j.ejvs.2012.06.004

Rossi R, Larese A, Dadvand P, Oñate E (2013) An efficient edge-based level set finite element method for free surface flow problems. Int J Numer Methods Fluids 71(6):687–716. https://doi.org/10.1002/fld.3680

Rudenick P, Bijnens B, García-Dorado D, Evangelista A (2013) An in vitro phantom study on the influence of tear size and configuration on the hemodynamics of the lumina in chronic type B aortic dissections. J Vasc Surg 57(2):464–474

Rudenick P, Bordone M, Bijnens B et al (2010a) A multi-method approach towards understanding the pathophysiology of aortic dissections—the complementary role of in-silico, in-vitro and in-vivo information. Springer, pp 114–123

Rudenick PA, Bordone M, Bijnens BH, Soudah E, Oñate E, Garcia-Dorado D, Evangelista A (2010b) Influence of tear configuration on false and true lumen haemodynamics in type B aortic dissection. In: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, pp 2509–2512. https://doi.org/10.1109/IEMBS.2010.5626689

Ryzhakov P, Soudah E, Dialami N (2019) Computational modeling of the fluid flow and the flexible intimal flap in type B aortic dissection via a monolithic Arbitrary Lagrangian/Eulerian fluid-structure interaction model. Int J Numer Methods Biomed Eng. https://doi.org/10.1002/cnm.3239

Sherrah A, Callaghan F, Puranik R et al (2017) Multi-velocity encoding four-dimensional flow magnetic resonance imaging in the assessment of chronic aortic dissection. AORTA J 5(3):80

Song SW, Yoo KJ, Kim DK, Cho BK, Yi G, Chang BC (2011) Effects of early anticoagulation on the degree of thrombosis after repair of acute DeBakey type I aortic dissection. Ann Thorac Surg 92(4):1367–1375

Soudah E, Rudenick P, Bordone M et al (2015) Validation of numerical flow simulations against in vitro phantom measurements in different type B aortic dissection scenarios. Comput Methods Biomech Biomed Eng 18(8):805–815

Soudah E, Pennecot J, Pérez JS, Bordone M, Oñate E (2011) MedicalGiD: from medical images to simulations, 4D MRI flow analysis. In: Tavares JMRS, Jorge RMN (eds) Computational vision and medical image processing: recent trends. Springer, Dordrecht, pp 145–160. https://doi.org/10.1007/978-94-007-0011-6_8

Sussman M, Smereka P, Osher S (1994) A level set approach for computing solutions to incompressible two-phase flow. J Comput Phys 114(1):146–159. https://doi.org/10.1006/jcph.1994.1155

Tolenaar J, Keulen VJ, Trimarchi S et al (2013) Number of entry tears is associated with aortic growth in type B dissections. Ann Thorac Surg 96(1):39–42. https://doi.org/10.1016/j.athoracsur.2013.03.087

Tsai TT, Evangelista A, Nienaber CA et al (2007) Partial thrombosis of the false lumen in patients with acute type B aortic dissection. N Engl J Med 357(4):349–359

Tsai T, Schlicht M, Khanafer K et al (2008) Tear size and location impacts false lumen pressure in an ex vivo model of chronic type B aortic dissection. J Vasc Surg 47(4):844–851. https://doi.org/10.1016/j.jvs.2007.11.059

Tse KM, Chiu P, Lee HP, Ho P (2011) Investigation of hemodynamics in the development of dissecting aneurysm within patient-specific dissecting aneurismal aortas using computational fluid dynamics (CFD) simulations. J Biomech 44(5):827–836

Wan AB Naim W, Ganesan P, Sun Z, Osman K, Lim E (2014) The impact of the number of tears in patient-specific Stanford type B aortic dissecting aneurysm: CFD simulation. J Mech Med Biol 14(02):1450017

Winter M, Schott B, Massing A, Wall W (2018) A Nitsche cut finite element method for the Oseen problem with general Navier boundary conditions. Comput Methods Appl Mech Eng 330:220–252. https://doi.org/10.1016/j.cma.2017.10.023

Yazdani A, Li H, Bersi MR et al (2018) Data-driven modeling of hemodynamics and its role on thrombus size and shape in aortic dissections. Sci Rep 8(1):2515

Zorrilla R, Larese A, Rossi R (2019) A modified finite element formulation for the imposition of the slip boundary condition over embedded volumeless geometries. Comput Methods Appl Mech Eng 353:123–157. https://doi.org/10.1016/j.cma.2019.05.007

Chapter 7

An embedded Finite Element framework for the resolution of strongly coupled Fluid–Structure Interaction problems. Application to volumetric and membrane-like structures

7.1 Article data

Title: An embedded Finite Element framework for the resolution of strongly coupled Fluid–Structure Interaction problems. Application to volumetric and membrane-like structures
Authors: R. Zorrilla, R. Rossi, R. Wüchner and E. Oñate
Journal: Computer Methods in Applied Mechanics and Engineering 368 (2020) 113179
Received: 28 January 2020 / Accepted: 23 May 2020 / Available online: 2 June 2020
DOI: 10.1016/j.cma.2020.113179

7.2 Scientific contribution

This chapter presents the last paper of the compendium. In short, the main contribution of the paper is the collection of all the previously developed CFD technologies to extend the applicability range of the embedded framework to FSI problems. It is important to highlight the capability of the novel embedded FSI toolbox to efficiently deal with any type of bodies, including membrane-like geometries.

For the sake of versatility, the FSI coupling is done in a black-box fashion using QN convergence accelerators within a Gauss-Seidel subdomain iteration. The paper discusses the particularities of the embedded FSI coupling, including a proposal for the imposition of the FM-ALE BCs. In addition, the article details the object oriented implementation, which is done according to code sustainability criteria in order to straightforwardly use the same implementation, not only in the body fitted case, but also in the resolution of other coupled problems (e.g. Conjugate Heat Transfer).

The embedded FSI solver is validated by solving several examples involving both volumetric and thin-walled bodies. The obtained solutions are compared, whenever it is possible, with reference data from the literature as well as with the results obtained with an alternative body fitted solver. The article proves that both the accuracy and the efficiency, which is measured in terms of the number of FSI subdomain iterations, are barely affected by the EBM method employed in the resolution of the CFD problem.

It is also interesting to highlight that the article presents one test example that entails extremely large rotations which would definitively have required remeshing if a body fitted solver would have been used. This benchmark showcases the enhanced robustness that the embedded solver has when dealing with such challenging scenarios.

Finally, the last example targets the main objective stated at the early beginnings of the thesis: the efficient and robust simulation of extremely lightweight civil engineering structures. Hence, the FSI analysis of a 4-point tent structure during an extreme wind episode is solved using the developed FSI framework. The results prove that the VWT numerical tool is production-ready and thus prepared to be applied to real engineering problems.

Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 368 (2020) 113179
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

An embedded Finite Element framework for the resolution of strongly coupled Fluid–Structure Interaction problems. Application to volumetric and membrane-like structures

R. Zorrilla[a,b,*], R. Rossi[a,b], R. Wüchner[c,b], E. Oñate[a,b]

[a] *Universitat Politècnica de Catalunya (UPC), Departament d'Enginyeria Civil i Ambiental, Spain*
[b] *International Center for Numerical Methods in Engineering (CIMNE), Spain*
[c] *Technische Universität München, Lehrstuhl für Statik, Germany*

Received 28 January 2020; received in revised form 22 May 2020; accepted 23 May 2020
Available online xxxx

Abstract

This work presents a Fluid–Structure Interaction framework for the robust and efficient simulation of strongly coupled problems involving arbitrary large displacements and rotations. We focus on the application of the proposed tool to lightweight membrane-like structures. Nonetheless, all the techniques we present in this work can be applied to both volumetric and volumeless bodies. To achieve this, we rely on the use of embedded mesh methods in the fluid solver to conveniently handle the extremely large deflections and eventual topology changes of the structure. The coupling between the embedded fluid and mechanical solvers is based on an interface residual black-box strategy. We validate our proposal by solving reference benchmarking examples that consider both volumetric and volumeless geometries. Whenever it is possible, we also compare the embedded solution with the one obtained with our reference body fitted solver. Finally we present a real-life application of the presented embedded Fluid–Structure Interaction solver.

MSC: 00-01; 99-00

Keywords: Fluid–Structure Interaction; Embedded Boundary Methods; Level set methods; Coupled problems; Black-box coupling; Volumeless bodies

1. Introduction

1.1. The fluid–structure interaction problem. Numerical approaches

The understanding of the interaction mechanisms that occur between a moving body and a surrounding (or passing through) fluid was one of the most relevant engineering challenges of the past century. Initially, the interest in this phenomenon, which is widely known as Fluid–Structure Interaction (FSI), was limited to the comprehension and quantification of the dynamic effects arising from the interaction with unsteady fluid flows. Some practical

* Corresponding author at: Universitat Politècnica de Catalunya (UPC), Departament d'Enginyeria Civil i Ambiental, Spain.
E-mail addresses: rzorrilla@cimne.upc.edu (R. Zorrilla), rrossi@cimne.upc.edu (R. Rossi), wuechner@tum.de (R. Wüchner), onate@cimne.upc.edu (E. Oñate).

https://doi.org/10.1016/j.cma.2020.113179
0045-7825/

examples are the aeroelastic phenomena (e.g. fluttering) occurring in slender structures such as aerofoils or bridge decks [1], the performance analysis of wind mill blades [2] or the water hammer in fluid ducts.

There also exist lots of natural phenomena, which include some of the human body processes, that involve FSI mechanisms. Specially in recent decades, this has motivated the appearance of new research lines that aim at understanding the mechanics behind them. Some examples are the flapping of insects wings [3], the hydrodynamics of fish fins [3] or the motion of jellyfish [4]. In the biomedical field, we can find some successful applications in [5–7] and [8], which study the blood flow in human vessels. Similarly, FSI numerical methods are applied in [9,10] and [11] to study the movement of the hearth valves or, more recently, in [12] to study the human cell motion.

The main feature common to all FSI problems is the mutual dependency between the Computational Fluid Dynamics (CFD) and the Computational Solid Mechanics (CSM) problems. Such dependency comes from the fact that the structure is deformable under the action of the fluid load. At the same time, the structure deformation implies a modification in the fluid problem geometry. This turns into a change in the load distribution over the structure, meaning that the fluid and the mechanical problem are somehow linked. This makes the FSI to belong to the so called coupled family of problems.

In this context, the tracking of the wet (or coupling) interface between the fluid and structure domains becomes crucial for the proper resolution of the FSI problem. Based on the movement of such coupling interface, we can roughly divide the FSI problems in two main groups.

On the one hand, it may happen that there is barely interaction between the fluid and the solid domains (e.g. massive civil engineering structures). In these cases, which are denoted as *one-way coupled* FSI problems, it is commonly assumed that there is no interaction on the wet interface. This greatly simplifies the problem as the coupling becomes in a simple load mapping from the fluid to the structure.

At the other extreme, we find the *strongly coupled* FSI problems, whose solution involves a high dependency between the fluid and the structure subdomain problems. This implies the need of taking into account arbitrary, and possibly large, movement of the wet interface.

This is not a problem in the solid domain as it is generally solved in a Lagrangian frame of reference. On the contrary, standard Eulerian fluid techniques are no longer valid as they cannot consider the movement of the coupling interface. Traditionally, this has been achieved either by upgrading the Eulerian frame of reference to the Arbitrary Lagrangian Eulerian (ALE) one [13,14] or by using non-conforming approaches such as the Immersed Boundary Method (IBM) [15].

The ALE methods are based on modifying the Eulerian advection velocity such that the solution is still consistent despite the movement of the Eulerian mesh. As the movement of the background mesh is only known in the coupling interface and domain boundaries, it is required to solve an extra mesh motion problem to spread it into the domain. The most common approach to achieve this is the Laplacian smoothing [16]. Alternatively, one can solve a structural similarity problem [17], which is known to be slightly more robust, specially when moderate rotations appear.

Disregarding the extra mesh motion problem overhead, the main disadvantage of ALE methods is that they are prone to yield excessively distorted, or even inverted, elements when large displacements or rotations occur in the boundaries. As a consequence, remeshing is often needed to tackle these undesired situations.

Another option to deal with moving interfaces, is to adopt purely Lagrangian approaches such as the Particle Finite Element Method (PFEM) [18–20]. While the use of a Lagrangian framework provides obvious advantages in following the internal interfaces, such methods imply remeshing at every time step.

Chimera type methods are also a feasible alternative for the problems at hand. These are based on representing the analysed bodies with a set of independent body conforming mesh patches, that can freely move over the background fluid domain. Thanks to this feature the rigid body movements can be easily treated. However, the method might still suffer from local elemental degeneration due to the deformation of the patch meshes.

All the methods that we have presented so far are based on body conforming (also known as body fitted) discretizations. This means that the mesh boundaries match, as closely as possible, the ones of the analysed bodies. Even though this is the most straightforward representation of the geometry, which typically leads to a simplification in the Boundary Conditions (BCs) imposition, it comes at the price of requiring a mesh updating technique (mesh motion or remeshing).

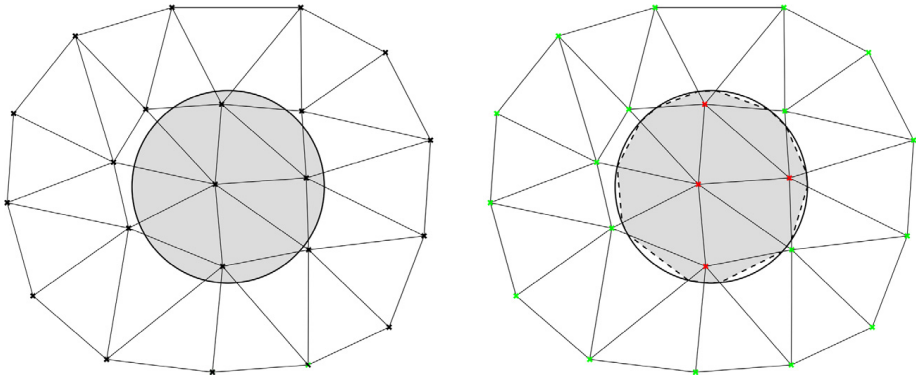


Fig. 1. Continuous distance function. Body with a well defined internal volume (left) and its continuous distance representation (right). Green and red markers denote the positive and negative distance nodes. The dashed lines represent the zero isosurface' skin representation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
Source: [28].

1.2. Embedded mesh methods in fluid–structure interaction problems

1.2.1. Embedded computational fluid dynamics

The need to overcome some of the limitations of body fitted approaches has led to the development of a wide variety of non-conforming mesh methods. Some of these are the IBM [15,21], the Embedded Boundary Method (EBM) (also known as Cut-FEM) [22] and the Shifted Boundary Method (SBM) [23,24].

All these techniques feature a volume mesh discretization, denoted as volume or background mesh, that is completely independent of the analysed bodies. As the interaction boundaries are no longer represented by the mesh, these techniques require doing some extra operations to implicitly represent the FSI boundaries in the background mesh.

Although this introduces another difficulty to the problem, the implicit representation becomes in an advantage when mesh updating issues are experienced since it allows substituting the mesh deformation problem by an update of the objects representation.

We highlight [25] as the pioneering work that applied non-conforming mesh methods to the resolution of an FSI problem. To consider the effect that the immersed body has in the fluid, the authors apply an artificial body force. This same approach is successfully applied also in [15] and [26]. However, it is reported to lack accuracy and (or) stability in other applications.

As an alternative, approaches based on the level set method can be used [27]. These implicitly represent the analysed geometries by using a signed distance function, which is computed by an auxiliary algorithm. Although all these algorithms are built on top of the skin and background mesh intersections calculation, we can classify them into two main families.

On the one hand, there are the algorithms that deal with the concept of inside/outside by using computer graphics techniques such as the ray-casting. These are the most extended level set methods as they always return a smooth continuous level set function (Fig. 1). Their main drawback is that they can only be applied to volumetric bodies. This is a direct consequence of the fact that the distance is computed node-by-node.

On the other hand, we have the discontinuous level set algorithms [29]. As it can be observed in Fig. 2, in this case the level set is not computed node-by-node but element-by-element. This is the key aspect that makes possible to represent not only volumetric bodies, but also membrane-like bodies with no internal volume [28,30]. Nonetheless, this feature comes at the price of having a potentially discontinuous level set, meaning that the same node may have different distance signs and values depending on the element considered.

Among all the level set based non-conforming mesh methods, the IBM is possibly the most straightforward approach to deal with the problems at hand. In short, the IBM consists in using the first layer of nodes laying inside the analysed bodies to impose the fluid essential BCs [31]. This means that the BCs are not imposed on the wet interface but on its closest dry nodes, something that makes the method prone to suffering from mass conservation issues. Nevertheless, it is possible to find different flavours of the IBM that are successfully applied to solve challenging FSI problems [32–34].

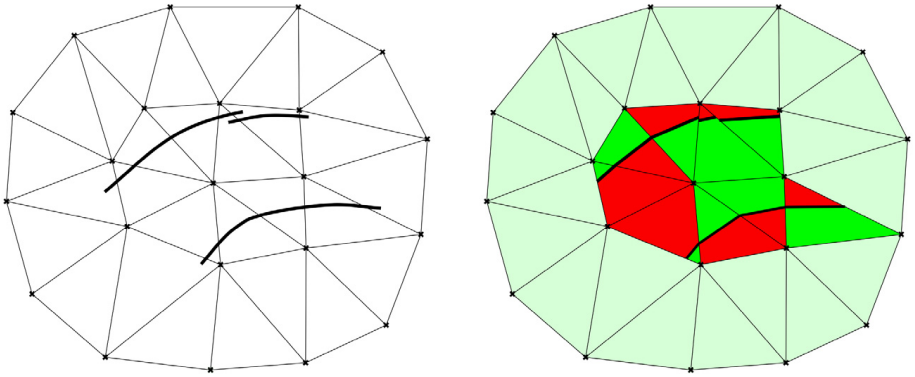


Fig. 2. Discontinuous distance function. Body without internal volume (left) and its discontinuous distance representation (right). Red and green portions of the cut elements indicate the positive and negative discontinuous distance regions. Light green denotes the non-intersected elements. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.) *Source:* [28].

In this context, the SBM [23,24] arises as a novel alternative that aims to overcome the potential mass conservation issues of the IBM. This is achieved by computing a modified boundary value that takes into account the distance between the level set and the dry nodes where the BCs are imposed.

Both the IBM and the SBM rely on the internal (dry) nodes to impose the fluid BCs. This precludes the analysis of thin-walled (volumeless) structures as the inside/outside concept becomes meaningless in this case.

In this regard, the eXtended Finite Element Method (X-FEM) appears a feasible alternative to the previous methods. As the boundary condition is imposed over the level set zero isosurface, the X-FEM can be applied to both CFD [35] and FSI [36,37] problems involving not only volumetric structures but also thin-walled ones. The most representative feature of the X-FEM is that the FE space is enriched in the elements intersected by the embedded skin. Such enhanced FE space makes possible the representation of the velocity and pressure discontinuities arising after the immersion of the analysed bodies. However, this comes at the cost of introducing extra Degrees Of Freedom (DOFs) that correspond to the enrichment unknowns.

The extra DOFs computational overhead becomes crucial when the problem involves moving boundaries as the system matrix graph needs to be reconstructed each time the level set is updated. Besides this, the extension of the method to distributed memory environments could be a challenge due to the presence of blending elements, which might lay in a different partition that the one of the intersected element.

The EBM is another feasible alternative to deal with the problems at hand. Like the X-FEM, the EBM also relies on applying the BCs over the Solid–Fluid interface cuts. Such BCs imposition is, however, done in a weak sense by using techniques such as the penalty method, the Nitsche method [38] or Lagrange multipliers based methods [39]. Despite the weak BCs imposition may introduce some stability issues, the EBM is purely local, avoiding thus the previously commented computational drawbacks of the X-FEM.

Among all the weak BCs imposition techniques we prefer the Nitsche-based approaches, as they are known to be more stable than penalty based ones. Moreover, they avoid the *inf–sup* stability issues and the DOFs set modification (and matrix graph reconstruction) of Lagrange multiplier methods.

It is possible to find in the literature multiple Nitsche based approaches to enforce the no-slip (stick) condition over the embedded fluid interface [40–42]. We highlight the approach presented in [40] due to its simplicity and performance. Concerning the slip condition, the Nitsche method is also used in [43] and [44], which we also highlight because of its stability and wide application range. However, all these techniques can only be applied to bodies that have a well defined internal volume.

In [3] this limitation is overcome by using a ghost cell method that relies on the neighbouring elements to enforce the embedded BCs. In [28] a purely elemental alternative is presented. This approach is based on the substitution of the common FE space within the intersected elements by a discontinuous one proposed in [45].[1] This makes possible to represent the velocity and pressure discontinuities arising from the immersion of any body, regardless of its type (volumetric or volumeless). This formulation is enhanced in [46] by using a Nitsche method to impose a Navier-slip condition, allowing thus the representation of any wall behaviour from the no-slip to the slip limits.

[1] From now on we will denote this modified discontinuous FE space as Ausas FE space after the name of the original author.

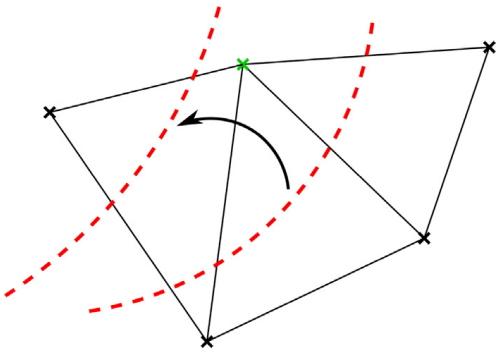


Fig. 3. Embedded nodal initialization example. The node highlighted in green changes its position from one side to the other of the level set (red dashed line), which moves as indicated by the black arrow. This requires to re-initialize the historical values in the green node to consistently approximate the time derivatives. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

1.2.2. Embedded mesh methods and moving bodies

As commented before, one motivation for the use of non-conforming mesh methods in the context of FSI is that they can efficiently handle arbitrary large movements and rotations. This enhanced robustness, however, comes at the price of some particularities that require to be taken care of.

The first one is the small cut instability which manifests when the level set function cuts an element so that there is a very small intersection. Although this issue also happens when solving steady objects CFD problems, its frequency may be even higher when the analysed bodies move across the background mesh. Therefore, it is required to use an auxiliary strategy to ensure the convergence of the problem. Some approaches, such as the ghost penalty method [22] or the ghost cell method [47], prevent the small cut ill-conditioning by using the neighbouring nodes or elements to control the solution.

Although the stability of this family of methods is proven, to the best of our knowledge no purely local approach exists that does not require the neighbouring elements calculation. Taking this into account, our proposal to avoid the small cut instabilities is to use a level set quality check and correction algorithm [28]. The robustness of this approach is evinced in [28], where the distance check and correction algorithm is applied in the resolution of CFD problems involving potentially ill-conditioned distance functions.

The second particularity is the historical data initialization problem. This issue appears when the values required for the temporal derivatives approximation are not consistently initialized in those nodes that move from one side to the other of the level set (Fig. 3). As reported in [28], this leads to an unphysical velocity field in the surroundings of the moving body as the approximation of the inertial terms is not consistent with the updated level set position. We remark that no distinction is done between continuous and discontinuous distance functions, as the historical values need to be consistently updated in both cases.

For the consistent historical data initialization we rely on the Fixed Mesh-Arbitrary Lagrangian Eulerian (FM-ALE) method. The FM-ALE method is described and applied to moving boundaries embedded CFD problems in [48]. This is extended in [49], where it is similarly applied to solid mechanics and FSI problems.

The main idea behind the FM-ALE method is to retrieve the consistent historical values from an auxiliary mesh, which is denoted as virtual mesh by the original authors. Such virtual mesh, is initially set as a copy of the background one. Then it is deformed by solving a mesh moving problem to follow the motion of the level set. After doing this, the historical values, as well as the mesh velocity, are projected back from the virtual mesh to the background origin one where the embedded CFD problem is solved. Finally, the virtual mesh is reset to its original configuration.

We note that the FM-ALE mesh motion problem likely involves small displacements and rotations as the virtual mesh movement is not accumulated but restarted at each time the FM-ALE algorithm is applied. As a consequence, the common problems associated with body conforming ALE solvers (element distortion or inversion) rarely appear when using the FM-ALE method. This is also confirmed in [48,49] and [50] where the FM-ALE method is applied to problems that involve large displacements and rotations of the immersed bodies.

1.3. Algorithms for fluid–structure interaction

1.3.1. Coupling schemes: FSI transmission conditions

The resolution of any coupled problem requires satisfying the so called interface transmission conditions. In the FSI case, this implies complying with the interface force equilibrium and continuity. In order to achieve this, a coupling scheme is required.

Coupling schemes can be classified according to how the transmission conditions are enforced in: Dirichlet–Neumann (DN), Neumann–Neumann (NN) and the Robin type ones such as the Robin–Robin (RR) or the Neumann–Robin (NR). In this work we opt for a DN coupling scheme due to its proved performance and stability. Besides, its implementation is reasonably simple and does not require to modify the subdomain formulations, enabling thus the use of non-intrusive coupling algorithms.

Hence, a Dirichlet BC is enforced over the coupling interface of one subdomain. This is commonly the lower density one, which in the FSI case is most likely the fluid medium. Accordingly, the fluid load is applied as a Neumann BC over the structure interface. The stability and performance of the DN scheme, as well as its interaction with different time discretization schemes, are thoroughly studied in [51].

Nevertheless, we note that DN schemes may lack convergence or have stability issues in some particular applications. In this context, NN and Robin type schemes can be an effective choice to overcome these limitations. As these applications are out of the scope of this work, we refer the reader to [52] and [53] (RR) as well as to [54] and [55] (NR) for a detailed analysis of these alternative coupling schemes.

1.3.2. Coupling strategy: monolithic and partitioned coupling

After defining the transmission conditions that ensure the coupling between the subdomain solutions, it is required to define a coupling strategy to solve the problem. These can be divided in two main families: monolithic and partitioned (also known as staggered) strategies.

Monolithic strategies are considered to be the more robust option to solve a coupled problem. They consist in solving all the subdomain problems within a unique coupled system of equations. This approach is successfully applied to FSI problems in [19] and [56]. More related to our case, in [49] the authors use an embedded monolithic formulation to solve both the fluid and the structure domains.

Despite its better coupling stability, monolithic strategies tend to yield poorly conditioned system matrices that eventually preclude the use of iterative solvers. Moreover their implementation is intrusive and typically requires the development of a new specific solver for the coupling.

On the contrary, partitioned approaches keep a separated solver for each one of the subdomains. This enables the reuse of already existent robust and widely validated subdomain solvers, thus allowing to focus the implementation effort on the coupling algorithm. However, this advantage comes at the price of a convergence rate that is typically not as good as that of monolithic alternatives.

Another feature of partitioned strategies is that the fulfilment of the transmission conditions is ensured by exchanging information at the coupling interfaces. Depending on how many times per time step such information exchange occurs, we can further classify these in loosely-coupled (or explicit) and strongly-coupled partitioned strategies.

In loosely-coupled staggered approaches the information exchange occurs once per time step [57]. Although these is the cheapest approach in terms of computational and implementation effort, they only converge when there is limited interaction between the coupled subdomains. Besides, they are known to suffer a lot from the so called added mass effect instability that appears when solving FSI problems involving incompressible fluids [58,59].

Conversely, in strongly-coupled staggered approaches the information exchange occurs until a certain coupling convergence criterion is reached. Even though the convergence rate is not optimal, implicit staggered approaches can be used in combination with convergence acceleration algorithms to alleviate this when dealing with strongly coupled problems. Concerning the computational effort, as pointed in [60] it is difficult to do a fair comparison with monolithic approaches because this is completely problem dependent.

Besides, according to how the information is exchanged we can further classify implicit strategies in Jacobi and Gauss–Seidel type iterations. In Jacobi type strategies all the information exchanges occur at the same time, allowing thus the parallel resolution of the subdomain problems. Contrariwise, Gauss–Seidel type strategies solve the subdomain problems in a sequential manner. This means that the latest obtained data from one subdomain is

used within the same iteration to solve the next one, as opposite to Jacobi type iterations, which will use the latest information in the next iteration.

Furthermore, there also exist alternative semi-implicit strategies. To this regard, we refer the reader to [55] where a comparative study of multiple alternatives to traditional coupling strategies can be found.

After having reviewed all the possible coupling strategies, we opt to use a Gauss–Seidel strongly-coupled staggered strategy. Disregarding monolithic approaches, this is the most robust technique that allows to reuse our existent CSM and embedded CFD solvers. At the same time, it brings up the concepts of black-box coupling [61,62]. Hence, we also require our coupling algorithm to not interfere with the subdomain solvers more than to get and set information, implying that these are considered as black boxes that return a solution for the provided input data.

Finally, we would like to highlight that the use of black-box coupling techniques not only enables the use of already existent in house codes but also external, even commercial, ones. This new wave of flexible coupling is denoted by the community as co-simulation [63].

1.3.3. Black-box interface residual minimization techniques

The selection of a Gauss–Seidel strongly-coupled strategy requires the definition of an interface residual function to be minimized. By doing this, it is ensured that the coupling transmission conditions are fulfilled. Since the choice of such residual is completely problem dependent, in this subsection we assume that it is already defined. We recover this in the methodology chapter 3 in where we discuss the particularities of the embedded FSI interface residual calculation.

As we are in favour of black-box coupling techniques, we require our residual minimization technique to have restricted access to the subdomain solvers.

The most straightforward approach is a fixed-point iteration with relaxation. Despite convergence can be reached using a constant relaxation parameter, the computational cost is normally prohibitive. In this context, dynamic relaxation schemes appear as a rather simple but effective alternative to reduce the number of coupling iterations. Among all of them, we highlight the second order Aitken relaxation scheme, which is widely used in the FSI community [64,65]. Although the Aiken relaxation scheme has an astonishing performance compared to its implementation effort, the required iterations are still far from being optimal.

A more efficient alternative is to pose the interface residual problem as a non-linear Newton–Raphson (N–R) iterative procedure, leading to the so called Jacobian-based resolution schemes. However, the use of a black-box coupling scheme intentionally precludes the exact calculation of the interface Jacobian that minimizes the problem. Nonetheless, this can be avoided by using Jacobian-free Newton–Krylov (JFNK) [66] or Quasi-Newton (QN) methods [62].

An application of JFNK methods in the FSI context can be found in [67]. Rather than approximating the complete interface Jacobian, JFNK methods are based on approximating its projection onto the iteration update vector by using a finite differences formula. Due to the inherent non-linearity of the FSI problem, such projection needs to be linearized by introducing a small perturbation in the finite differences formula. Although there exist in the literature some formulas to compute the value of such small perturbation [68], its optimal value is completely problem dependent.

The selection of such perturbation constant can be bypassed by using a QN algorithm. In short, the idea behind any QN algorithm is to use the information from previous iterations to calculate an approximation of the interface Jacobian by using a linearized formula. Depending on how such formula is obtained, we end up with one or another QN method. In the following we highlight those ones that have been successfully applied to solve FSI problems.

Although nowadays it is not the most popular method, we acknowledge Broyden's iteration as the pioneer QN algorithm applied to FSI problems [68]. A more recent approach is the Interface Quasi-Newton with Inverse Jacobian from Least Squares model (IQN-ILS) proposed by Degroote et al. in [60]. The IQN-ILS is further studied in [69], where the authors do a thorough performance analysis of the method, showing that the IQN-ILS performance overcomes the Aitken and JFNK ones in all the reported examples. A fairly similar approach to the IQN-ILS is the MultiVector Quasi-Newton (MVQN) algorithm presented by Bogaers et al. in [62], which according to the original authors converges slightly faster than the IQN-ILS. This comes however at the price of requiring to compute a square matrix inversion, whose size equals the interface residual one, each time the interface Jacobian needs to be approximated. Such limitation has been not long ago surpassed in [70].

In this work we decide to use a black-box coupling strategy based on QN algorithms. Despite QN methods can be more computationally expensive than JFNK approaches, as they involve a rather large number of linear algebra

operations, they are proven to be a robust and efficient approach for the FSI residual minimization. Among all the presented QN algorithms, we select the MVQN method as reference convergence accelerator due to its reported well performance. Besides this, we do not foresee that its matrix inversion becomes in a limitation for the problems we aim to solve in this work.

1.4. Objectives

The main target of this work is to build a robust and efficient FSI framework able to solve strongly coupled problems involving structures that undergo large displacements and rotations, with possible topology changes. Taking these requirements into account, we build an implicit black-box coupling strategy on top of a non-conforming embedded CFD solver.

Besides this, we focus on having a unique framework that works with both thin-walled (shells and membranes) and volumetric structures. This is achieved by switching the level set type depending on the nature of the immersed bodies (i.e. continuous for volumetric bodies and discontinuous for membrane-like ones).

The paper is organized as follows. Section 2 briefly describes the embedded fluid and structure formulations. Section 3 details the methodology we follow to achieve the FSI coupling of an embedded CFD solver with a body conforming mechanical one. Section 4 presents our improvement to the FM-ALE method. Section 5 collects the benchmarking of our proposal. A real-life application is also presented in this section. Finally, the conclusions and further work lines are collected in the last section.

All the methods presented in this work are implemented and available to use within the Kratos Multiphysics open source framework [71,72]. All the models are generated, meshed and post-processed using the GiD pre and post-processor [73,74].

2. Formulation

In this section we describe the formulations that are implemented in each one of the subdomain solvers. We firstly describe the governing equations, stabilization technique and BCs imposition of the embedded CFD solver. Secondly, we briefly present the element technologies we use in the structural solver.

2.1. Fluid domain

2.1.1. Governing equations

In this work we only consider viscous incompressible Newtonian fluids. Therefore, the Cauchy stress tensor $\boldsymbol{\sigma}$ is defined as $\boldsymbol{\sigma} = -p\mathbf{I} + \mathbb{C} : \boldsymbol{\nabla}^s\mathbf{v}$, where $\mathbf{v}$ is the fluid velocity, p the fluid pressure, $\boldsymbol{\nabla}^s$ the symmetric gradient operator and $\mathbb{C}$ the Newtonian constitutive tensor that describes the viscous behaviour. Inserting the previous definition of $\boldsymbol{\sigma}$ into the linear momentum equilibrium equation and combining the resulting equation with the incompressible (divergence free) mass balance equation yields the well-known viscous incompressible Navier–Stokes (N–S) equations.

Even though this form of the N–S equations is extensively used to solve almost incompressible flows, we opt to introduce a slight compressibility in the mass conservation equation. This is done to avoid the pressure field to be undefined when isolated fluid domains (i.e. bubbles) appear in the level set function. We note that this undesired situation could be immediately dealt with by imposing a Neumann BC inside the isolated fluid cavities. However, it is not always possible to "a priori" known their location, as they likely appear after the level set calculation, either because it is computed from a poorly defined input geometry or due to the deformation of the immersed geometry. In this context, the extra pseudo-compressibility becomes in a numerical tool to ensure that the pressure field remains always bounded.

Hence, we assume that the density ρ can be related to the pressure by the simplified speed of sound c equation of state $p = \rho c^2 \implies c^2 = \partial p/\partial\rho$, which is valid for almost incompressible fluids. By further considering that the density fluctuations are negligible, meaning that $\boldsymbol{\nabla}\rho \approx \mathbf{0}$, we obtain the final form of the pseudo-compressible N–S equations used in this work

$$\rho\frac{\partial\mathbf{v}}{\partial t} + \rho\mathbf{a}\cdot\boldsymbol{\nabla}\mathbf{v} - \boldsymbol{\nabla}\cdot\left(\mathbb{C} : \boldsymbol{\nabla}^s\mathbf{v}\right) + \boldsymbol{\nabla}p = \rho\mathbf{b} \tag{1a}$$

$$\frac{1}{\rho c^2}\frac{\partial p}{\partial t} + \boldsymbol{\nabla}\cdot\mathbf{v} = 0 \tag{1b}$$

where $\partial \bullet / \partial t$ and ∇ denote the partial time derivative and gradient operators, $\mathbf{b}$ the volume (body) forces and $\mathbf{a}$ the convective velocity, which in an ALE framework equals $\mathbf{v} - \mathbf{v_m}$ being $\mathbf{v_m}$ the mesh velocity. In fixed mesh approaches, like the one presented used this work, it is commonly assumed that $\mathbf{a} = \mathbf{v}$. However, we highlight that it is essential to keep $\mathbf{a} \neq \mathbf{v}$ for the future application of the FM-ALE algorithm.

We also note that the single-fluid fully-incompressible form $\nabla \cdot \mathbf{u} = 0$ is recovered as $c \to \infty$. Taking this into account, the speed of sound is taken as 10^{12}m/s, so that the compressibility is effectively negligible. Nonetheless, the option to employ a different value is left for those cases in which isolated volumes of fluid may appear.

We refer the reader to [28] for a detailed description of the variational form derivation and implementation of the governing equations.

2.1.2. Stabilization

In this work we only use simplicial elements with same velocity and pressure interpolation order. These elements do to not satisfy the *inf–sup condition*, requiring thus the use of a stabilization method [75]. Among all the techniques that have been proven to be effective in similar CFD problems (e.g. the Finite Increment Calculus [76] or the Galerkin Least Squares (GLS) [77]) we select the Variational Multiscales Method (VMS) [78,79].

The VMS relies on separating the solution fields $\mathbf{v}$ and p in two scales as $\mathbf{v} = \mathbf{v_h} + \mathbf{v_s}$ and $p = p_h + p_s$. $\mathbf{v_h}$ and p_h are the FE resolvable scales while $\mathbf{v_s}$ and p_s are the subscales representing the fluctuations that cannot be captured by the FE mesh and thus need to be modelled. The subscale models are commonly built as a projection of the FE residuals. In our case we apply the most simplest approach, which is to use an identity projection operator, leading to the well-known Algebraic Sub-Grid Scales (ASGS) approach [80].

Hence, the ASGS velocity ($\mathbf{u_s}$) and pressure (p_s) subscales are computed from the FE velocity and pressure solutions as

$$\mathbf{u_s} = \tau_1 \mathbf{R^M}(\mathbf{u_h}, p_h) \tag{2a}$$

$$p_s = \tau_2 R^C(\mathbf{u_h}, p_h) \tag{2b}$$

where τ_1 and τ_2 are the stabilization constants, that we take from [80], and $\mathbf{R^M}$ and R^C are the residuals of the momentum and mass conservation equations, which can be found in [28].

2.1.3. Embedded boundary condition imposition

This subsection presents the two embedded methods that we employ to weakly impose the fluid BCs over the level set intersections. On the one hand, we use the modified Nitsche method presented in [40] for those bodies that have a well defined internal volume, On the other hand, we use the discontinuous formulation described in [46] for the membrane-like bodies that feature no internal volume. In the following, we briefly describe the particularities of each one of these approaches.

Volumetric bodies: Modified nitsche method no-slip imposition. The modified Nitsche method proposed by Codina and Baiges requires to firstly identify the set of nodes that are in the interior side of the level set, which is to say, the ones that have a negative distance value. Then, the N–S momentum contribution is substituted in these by an L_2 minimization of the velocity constraint error. We refer the reader to [40] for the implementation details and convergence analysis of the method.

We highlight that, despite its simplicity, the method features a remarkably good performance. We also observe that the stability of the method is only compromised when elements with a small portion of fluid appear. This is due to the fact that the interior nodes that enforce the velocity constraint are in this case far from the embedded skin. Therefore, their shape function values are close to zero in the level set intersection, inducing thus the appearance of spurious peak values. As we mention in the introduction of the paper, we tackle these issues with our level set quality check and correction algorithm.

Volumeless bodies: Ausas FE space and Nitsche Navier-slip imposition. Following the ideas presented in [28], we propose to use the Ausas discontinuous FE space (Fig. 5) to represent the elemental velocity and pressure jumps arising from the immersion of a thin-walled body. This capability comes however at the price of featuring worse interpolation properties than the standard linear FE space as only piecewise constant functions can be exactly represented.

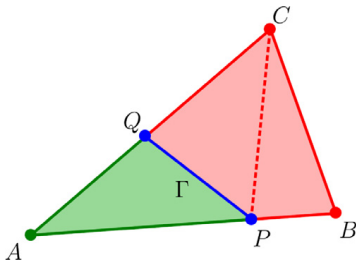
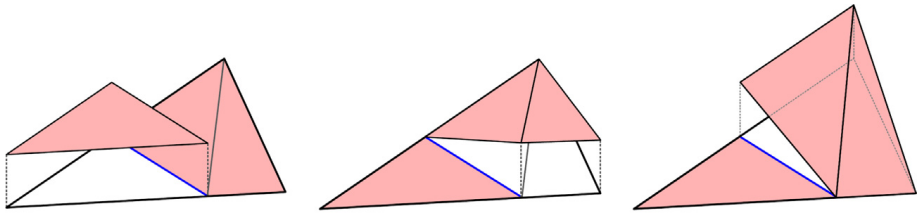


Fig. 4. Partition of a triangular finite element ABC into subelements following the interface PQ. *Source:* [45].



(a) N_A (b) N_B (c) N_C

Fig. 5. Triangle shape functions for the Ausas finite element space and the splitting pattern in 4. *Source:* [45].

On top of that, we also follow the ideas presented in [44] to do a Nitsche-based imposition of the Navier-slip condition over the discontinuous level set intersections.

The Navier-slip condition can be understood as a wall law whose behaviour is regulated by the so called slip length parameter ε. Hence, it approaches the no-slip limit as $\varepsilon \to 0$. Contrariwise, it tends to the slip limit as $\varepsilon \to \infty$. Having said this, it is important to remark the capability of the Nitsche method presented [44] to deal with both the slip and the no slip limits without losing its stability.

From a numerical perspective, the Navier-slip condition is indeed a no penetration condition in the normal direction combined with a shear force imposition in the tangential one. This allows to split and express these two contributions as

$$\mathbf{P}^n (\mathbf{v} - \mathbf{g}) = \mathbf{0} \tag{3a}$$

$$\mathbf{P}^t \left(\varepsilon \left(\left[\mathbb{C} : \left(\boldsymbol{\nabla}^s \mathbf{v} \right) \right] \cdot \mathbf{n} - \mathbf{h} \right) + \mu \left(\mathbf{v} - \mathbf{g} \right) \right) = \mathbf{0} \tag{3b}$$

where $\mathbf{g}$ and $\mathbf{h}$ denote the velocity and the tangential traction to be imposed over the level set intersection. $\mathbf{P}^n$ and $\mathbf{P}^t$ denote the normal and tangential projection operators, which are computed from the unit normal $\mathbf{n}$ as $\mathbf{P}^n = \mathbf{n} \otimes \mathbf{n}$ and $\mathbf{P}^t = \mathbf{I} - \mathbf{n} \otimes \mathbf{n}$, being $\mathbf{I}$ the second order identity tensor.

We note that in the Nitsche imposition of Eq. (3) needs to be applied in both sides of the level set intersections after the ability of the Ausas FE space to disconnect the solutions in each element subdivisions.

More details on the implementation, convergence and accuracy analysis of the discontinuous Nitsche Navier-slip imposition can be found in [46]. This approach is also validated with data from biomedical *in vitro* experiments in [8].

2.2. Structure domain

Bearing in mind that our goal is to solve FSI problems that involve large displacements and (or) rotations, we only consider non-linear finite displacements formulations.

Therefore, volumetric bodies are discretized by using Total Lagrangian linear quadrilateral elements. With regard to Total Lagrangian solid FE, we refer the reader to [81].

Regarding the thin-walled structures, shells are modelled by an enhanced Discrete Kirchhoff Triangular (DKT) based element. This formulation is developed in [82] by combining the DKT plate bending element with the OPtimal Triangle (OPT) membrane element, yielding the so called DKT-OPT element. Such element implements a corotational non-linear formulation capable to efficiently deal with small strain large displacement and rotation problems.

As the benchmarking experiments we reproduce do not require advanced constitutive laws, we can assume throughout the paper that the material response always lays in the elastic regime. Thus, we use the neo-Hookean and Kirchhoff–Saint-Venant material models, which are compatible with the chosen finite displacements formulations.

3. Embedded fluid–structure interaction coupling

In this section we detail the particularities of the embedded FSI coupling. First of all, the embedded DN interface residual is defined together with the non-linear strategy we implement for its minimization. Then, we briefly describe the implementation of such strategy, which is conceived to ease the integration in an existent FSI solver. From now on, we denote the fluid and structure domains as Ω_f and Ω_s while their corresponding FSI interfaces are denoted as Γ_f and Γ_s. Similarly, k superscripts stand for the FSI non-linear iterations counter.

3.1. Dirichlet–Neumann interface residual minimization

Prior to the definition of any interface residual, we recall that the use of an embedded CFD solver entails that Γ_f is implicitly represented by the level set function. As a consequence, the FSI interface residual can only be defined in terms of the structure interface Γ_s.

In this context, the FSI interface residual vector $\mathbf{r}_{\Gamma_s}^{\mathbf{k+1}}$ to be minimized can be computed from the current and previous iteration displacement results $\tilde{\mathbf{u}}_{\Gamma_s}^{\mathbf{k+1}}$ and $\mathbf{u}_{\Gamma_s}^{\mathbf{k}}$ as

$$\mathbf{r}_{\Gamma_s}^{\mathbf{k+1}} = \tilde{\mathbf{u}}_{\Gamma_s}^{\mathbf{k+1}} - \mathbf{u}_{\Gamma_s}^{\mathbf{k}} \tag{4}$$

Here the vector $\tilde{\mathbf{u}}_{\Gamma_s}^{\mathbf{k+1}}$ is the unrelaxed displacement solution obtained after applying the iteration $k+1$ fluid load. Conversely, $\mathbf{u}_{\Gamma_s}^{\mathbf{k+1}}$ is the relaxed displacement obtained after the addition of the solution update $\Delta\mathbf{u}_{\Gamma_s}^{k+1}$ as

$$\mathbf{u}_{\Gamma_s}^{\mathbf{k+1}} = \mathbf{u}_{\Gamma_s}^{\mathbf{k}} + \Delta\mathbf{u}_{\Gamma_s}^{k+1} \tag{5}$$

According to our black-box FSI coupling choice, we require the interface residual (Eq. (4)) minimization to be done with a non-intrusive method (e.g. relaxation, JFNK or Quasi-Newton). To that purpose we opt for the MVQN method [62], which allows to obtain the correction $\Delta\mathbf{u}_{\Gamma_s}^{k+1}$ from the current iteration interface residual $\mathbf{r}_{\Gamma_s}^{\mathbf{k+1}}$ and the previous iteration solution $\mathbf{u}_{\Gamma_s}^{\mathbf{k}}$.

At this point it is important to recall that we tackle the non-linearity of the FSI problem by using a staggered iterative strategy. Hence, we also define an absolute convergence criterion to stop the subdomain iteration when the next residual norm condition is met

$$\frac{\left\|\mathbf{r}_{\Gamma_s}^{\mathbf{k+1}}\right\|}{\sqrt{n_{\Gamma_s}}} < tol_{FSI} \tag{6}$$

where n_{Γ_s} denotes the number of interface unknowns, whose value equals the structure interface residual $\mathbf{r}_{\Gamma_s}$ size.

All these ingredients are combined in Algorithm 1, which describes the steps of our embedded FSI iterative strategy. In this algorithm ϕ_f stands for the level set function while the operators $\mathbf{F}$, $\mathbf{P}^{f\to s}$, $\mathbf{S}$ and $\mathbf{CA}$ represent the fluid solver, the interpolator from the fluid background mesh to the structure skin one, the solid solver and the convergence accelerator. We note that all these operators are considered and implemented as independent objects that, from a given input data, give back an output result.

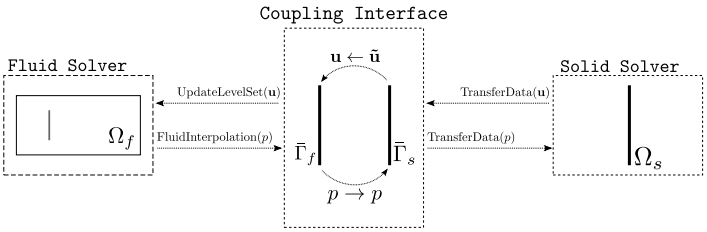


Fig. 6. Coupling objects and methods schematic representation. The dashed lines represent the fluid and solid solvers as well as the FSI coupling interface classes. The dotted lines represent the information exchanges.

With regard to the pressure interpolation $\mathbf{P}^{f\rightarrow s}$, we point out that the same FE space used in the resolution of the CFD problem is employed. Hence, when dealing with volumetric bodies we use the standard linear FE space. On the contrary, we use the Ausas FE space when volumeless bodies are analysed.

Algorithm 1: Embedded FSI partitioned strategy for DN scheme.

$\mathtt{k} = 0$;
$\mathtt{max\_it} = n$;
while $\mathtt{k} \leq \mathtt{max\_it}$ **do**
  $\mathtt{k} \mathrel{+}= 1$;
  1) update level set: $\mathbf{u}^{\mathbf{k}}_{\Gamma_\mathbf{s}} \rightarrow \phi_f^{k+1}$;
  2) do FM-ALE operations: $\mathbf{FMALE}(\phi_f^{k+1})$;
  3) solve embedded fluid: $\mathbf{F}(\phi_f^{k+1}) \rightarrow (\mathbf{v}_\mathbf{f}^{\mathbf{k+1}}, p_f^{k+1})$;
  4) interpolate pressure to structure skin: $\mathbf{P}^{f\rightarrow s}(p_f^{k+1}) \rightarrow p_{\Gamma_s}^{k+1}$;
  5) solve structure: $\mathbf{S}(p_{\Gamma_s}^{k+1}) \rightarrow \tilde{\mathbf{u}}_\mathbf{s}^{\mathbf{k+1}}$;
  6) compute residual in Γ_s: $\mathbf{r}_{\Gamma_\mathbf{s}}^{\mathbf{k+1}} = \tilde{\mathbf{u}}_{\Gamma_\mathbf{s}}^{\mathbf{k+1}} - \mathbf{u}_{\Gamma_\mathbf{s}}^{\mathbf{k}}$;
  7) update solution in Γ_s: $\mathbf{CA}(\mathbf{r}_{\Gamma_\mathbf{s}}^{\mathbf{k+1}}, \mathbf{u}_{\Gamma_\mathbf{s}}^{\mathbf{k}}) \rightarrow \mathbf{u}_{\Gamma_\mathbf{s}}^{\mathbf{k+1}}$;
  if $\frac{\left\|\mathbf{r}_{\Gamma_\mathbf{s}}^{\mathbf{k+1}}\right\|}{\sqrt{n_{\Gamma_s}}} < \mathtt{tol_{FSI}}$ **then**
    break;
  end
end

3.2. Implementation: the FSI interface object

In this subsection we describe our Object Oriented (OO) implementation proposal for Algorithm 1. Such implementation approach, which is schematically depicted in Fig. 6, is based on two basic requirements.

First of all, it is mandatory that each subdomain solver (`Fluid Solver` and `Structure Solver` in Fig. 6) works as a stand-alone unit (class) to achieve an effective OO implementation. Secondly, it is important that the communication of the subdomain solvers with the rest of the code is done through the already existent Application Programming Interface (API). The combination of these two requests ensures that there is no need to modify the implementation of neither the fluid nor the structure solvers for the specific purpose of coupling.

The design is completed by the introduction of a `Coupling Interface` object (Fig. 6), which encapsulates the tools used in the coupling, such as, mapping, residual calculation and solution update.

Multiple options exist for the implementation of the `Coupling Interface` object. Our choice is to duplicate it, so to make an independent instance in each single domain interface. In other words, one interface object operates on the $\bar{\Gamma}_f$ interface mesh while the other does so on the $\bar{\Gamma}_s$ one.

In a general case, either of these two coupling interface instances can be used to perform the residual minimization and interface solution update. However, in the embedded case the skin mesh corresponding to the $\bar{\Gamma}_f$ interface object needs to be created as an exact copy of the $\bar{\Gamma}_s$ one. This choice is taken since the Γ_f mesh only exists implicitly, as the zero isosurface of the level set function ϕ_f.

The DN coupling scheme described in Algorithm 1 can be implemented according to these guidelines. In the following, we tag those operations that are performed in the fluid coupling interface instance with $\texttt{CI}_\texttt{f}$. Equivalently, we mark those ones belonging to the structure interface with $\texttt{CI}_\texttt{s}$. Hence, Algorithm 1 translates to

1. Ω_f: compute the level set function ϕ_f from the $\bar{\Gamma}_f$ interface mesh
2. Ω_f: solve the embedded CFD problem.
3. $\texttt{CI}_\texttt{f}$: interpolate the pressure from Ω_f to $\bar{\Gamma}_f$.
4. $\texttt{CI}_\texttt{f} \rightarrow \texttt{CI}_\texttt{s}$: transfer the pressure from $\bar{\Gamma}_f$ to $\bar{\Gamma}_s$.
5. $\texttt{CI}_\texttt{s}$: transfer the pressure load in $\bar{\Gamma}_s$ to Γ_s.
6. Ω_s: solve the mechanical problem.
7. $\texttt{CI}_\texttt{s}$: get the obtained displacement $\tilde{\mathbf{u}}$ from Γ_s, calculate the interface residual (Eq. (4)) and call the convergence accelerator (owned by $\texttt{CI}_\texttt{s}$) to do the solution update $\Delta\mathbf{u}$ (Eq. (5)).
8. $\texttt{CI}_\texttt{s} \rightarrow \texttt{CI}_\texttt{f}$: update the $\bar{\Gamma}_f$ interface mesh accordingly to compute the next iteration level set function if convergence is not reached.

We would like to conclude this section highlighting the potential of this implementation. The crucial advantage of this approach is to make the FSI implementation essentially the same for the body-fitted and embedded cases. This can be efficiently achieved by simply overriding some of the auxiliary dotted lines operations in Fig. 6. Furthermore, the `Coupling Interface` can be used out of the box for coupled problems other than the FSI as its API allows customizing the variables the data is retrieved from and sent to.

4. Lagrangian interface FM-ALE algorithm

In this section we target the embedded moving bodies historical data initialization issue described in the introductory section. With that aim, we firstly describe the main features of the FM-ALE algorithm proposed by Codina et al. in [48]. Secondly, we present our proposal to compute the FM-ALE mesh moving problem BCs from the motion of the embedded body.

As we have already mentioned, the purpose of the FM-ALE algorithm is to perform a consistent advection of information in the vicinities of the embedded bodies so that their historical values are consistently initialized as their boundaries move across the background mesh.

To this end, a copy of the fixed background mesh $\mathcal{M}^0$ is created at the beginning of the simulation. Such copy is the so called virtual mesh, which we denote as $\mathcal{M}^v$. The idea of the method is to deform $\mathcal{M}^v$, by solving a standard ALE mesh motion problem, to follow the embedded boundaries motion. Then, the historical data, as well as the mesh velocity, are projected from the deformed $\mathcal{M}^v$ to $\mathcal{M}^0$ where the CFD problem is to be solved. Hence, the **FMALE** call in Algorithm 1 can be implemented as described in Algorithm 2. We refer the reader to [48] for a detailed discussion about the implementation and validation of the FM-ALE method.

Algorithm 2: FM-ALE algorithm.

if `virt_mesh` == *false* **then**
  create virtual mesh: $\mathcal{M}^v$ = **CopyMesh**($\mathcal{M}^0$);
  create mesh motion solver: **M**($\mathcal{M}^v$);
  `virt_mesh` = true;
end
1) mesh motion problem;
  1.1) set mesh motion BCs: **M**($\mathcal{M}^v$).**SetBCs**();
  1.2) solve mesh problem: **M**($\mathcal{M}^v$).**Solve**() ;
  1.3) compute mesh velocity: **M**($\mathcal{M}^v$).**ComputeMeshVel**() $\rightarrow$ $\mathbf{v_m}$;
2) data interpolation from $\mathcal{M}^v$ to $\mathcal{M}^0$;
forall the $\texttt{node}_\texttt{i} \in \mathcal{M}^0$.Nodes() **do**
  2.1) find $\texttt{node}_\texttt{i}$ in $\mathcal{M}^v$.Elements() $\rightarrow$ $\texttt{elem}^\texttt{v}$;
  2.2) interpolate $\mathbf{v_i^n}$, $\mathbf{v_i^{n-1}}$, $\mathbf{v}_{\mathbf{m_i}}$, p_i^n, p_i^{n-1} within $\texttt{elem}^\texttt{v}$;
end
3) revert $\mathcal{M}^v$ movement;

At this point, it is interesting to discuss which are most suitable BCs for the mesh motion problem in the step 1.1 of Algorithm 2. Taking the body fitted case as reference, it is required that the FSI interface has a Lagrangian motion to ensure the mass conservation. This turns into enforcing $\mathbf{u}_m^{\Gamma} = \mathbf{u}_{\mathbf{bs}}^{\Gamma}$ and $\mathbf{v}_{\mathbf{m}}^{\Gamma} = \mathbf{v}_{\mathbf{s}}^{\Gamma}$. However, this cannot be imposed in a strong sense when using an embedded discretization due to the implicit representation of Γ_s.

With this regard, we present a slight improvement to the FM-ALE algorithm to approximatively fulfil such Lagrangian motion of the interface. The proposal is based on modifying the step 1.1 in Algorithm 2 to impose a set of $\mathbf{u_m}$ nodal values in the split elements such that their interpolation in the level set intersections yield $\mathbf{u}_{\mathbf{bs}}^{\Gamma}$. This ensures that the mesh displacement field obtained after the deformation of $\mathcal{M}^v$ (step 1.2 in Algorithm 2) matches the movement of Γ_s in an approximate Lagrangian sense.

In Fig. 7(a) we present a simple toy example to explain our proposal. In this figure, the background mesh is represented with solid black lines while the intersecting structure skin is represented with dashed red lines. To ease the discussion we assume that the variable to extrapolate, which we denote as a^{Γ}, is scalar. Nonetheless, the same procedure can be identically applied to a vector variable by extending what is explained in here to each one of the vector components. We note that in the FM-ALE case, the variable of interest is the structure displacement increment $\Delta\mathbf{u_s} = \mathbf{u_s^{n+1}} - \mathbf{u_s^n}$.

The proposed technique consists in solving an extrapolation problem on an auxiliary mesh made up with the background mesh intersected edges. This problem is implemented within the step 1.1 of the FM-ALE algorithm 2 by adding the following sub-steps

1. find those elements that are intersected by the level set. For each intersected element, isolate their intersected edges (Fig. 7(b)).
2. for each intersected edge e_{ij}, obtain the point $\mathbf{x^{int}}$ (light green square dots in Fig. 7(b)) in where the edge intersects the structure skin. Save the distance $d_e = \left\| \mathbf{x^{int}} - \mathbf{x^0} \right\| / l_e$, where $\mathbf{x^0}$ and l_e are the i-node coordinates and length of e_{ij}.
3. interpolate and save the value of the variable of interest a^{Γ} from the structure skin to the intersection points $\mathbf{x^{int}}$.
4. create one extrapolation line element from each intersected edge
5. solve the problem in the green light nodes auxiliary mesh (Fig. 7(c)).

After the resolution of the edge mesh extrapolation problem, we obtain a set of nodal values $\mathbf{a_{ij}}$ whose interpolation in the level set intersections (light green square dots in Fig. 7(c)) equals the a_{Γ} values previously obtained from the structure skin Γ_s. Besides the extrapolation contribution, we add an extra term that penalizes the gradient of the solution values $\mathbf{a_{ij}}$. This prevents the appearance of spurious values in those situations in which the solution is not unique (e.g. isolated edges).

Hence, for an intersected edge e_{ij} the problem takes the form

$$\mathbf{LHS_{ij}}\mathbf{a_{ij}} = \mathbf{RHS_{ij}} \tag{7}$$

where the left and right hand sides of the problem are defined as

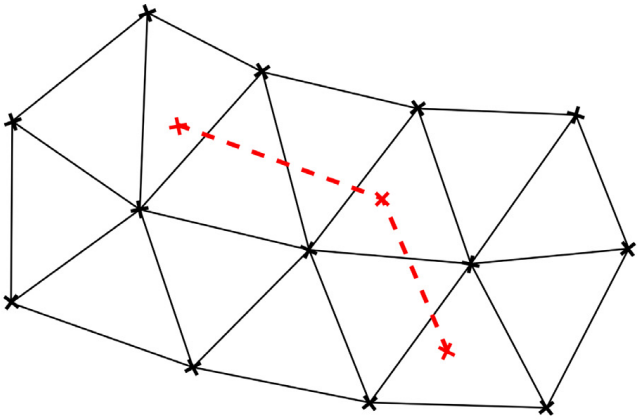
$$\mathbf{LHS_{ij}} = \begin{pmatrix} N_i N_i & N_i N_j \\ N_j N_i & N_j N_j \end{pmatrix} + \begin{pmatrix} \kappa^G \kappa^G & -\kappa^G \kappa^G \\ -\kappa^G \kappa^G & \kappa^G \kappa^G \end{pmatrix} \tag{8}$$

and

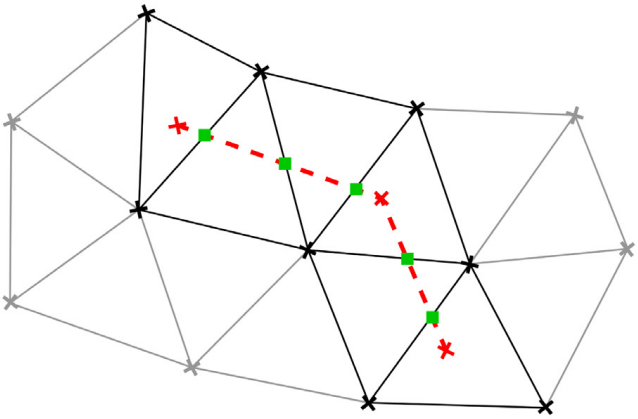
$$\mathbf{RHS_{ij}} = \begin{pmatrix} N_i a^{\Gamma} \\ N_j a^{\Gamma} \end{pmatrix} \tag{9}$$

N_i and N_j are computed from the edge intersection distance d_e as $N_i = 1 - d_e$ and $N_j = d_e$ while κ^G is the solution gradient penalty constant. According to our experience, small values of κ^G (of the order of 10^{-3}–10^{-4}) are sufficient to drive the problem towards a smooth solution.

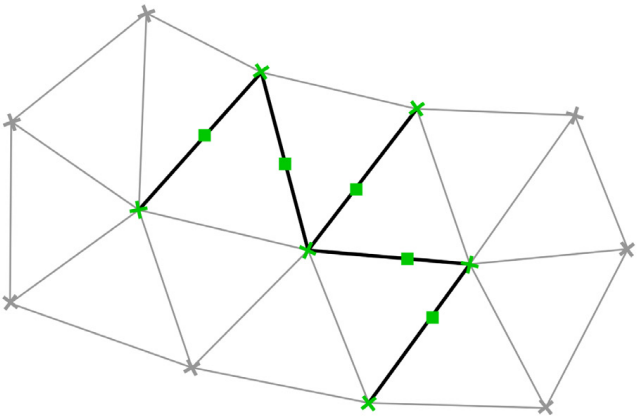
We would like to conclude this section saying that taking the extrapolation problem values as mesh motion BCs guarantees that the movement is large enough to properly initialize the historical values but sufficiently small to avoid the common mesh motion problems (i.e. element distortion and inversion). Last but not least, we also highlight that the extra computational effort of the extrapolation problem is negligible compared to the mesh motion one as it is exclusively solved in those nodes belonging to the intersected edges.



(a) Toy example representation. Black solid lines represent the background mesh. Red dashed lines represent the embedded structure skin mesh.



(b) Intersected elements and edges search (black solid lines). Edge intersection points highlighted with light green square dots.



(c) Edge elements mesh to solve the extrapolation problem (black solid lines). The nodes of interest where the extrapolation is solved are highlighted in light green.

Fig. 7. Embedded nodal variable extrapolation from skin to background mesh. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

5. Validation

In this section we present the validation of the proposed embedded FSI framework. We base the correctness assessment on the comparison with reference data available in the literature. Whenever it is possible, we also compare the embedded solution with the ones obtained with our reference body fitted FSI solver.

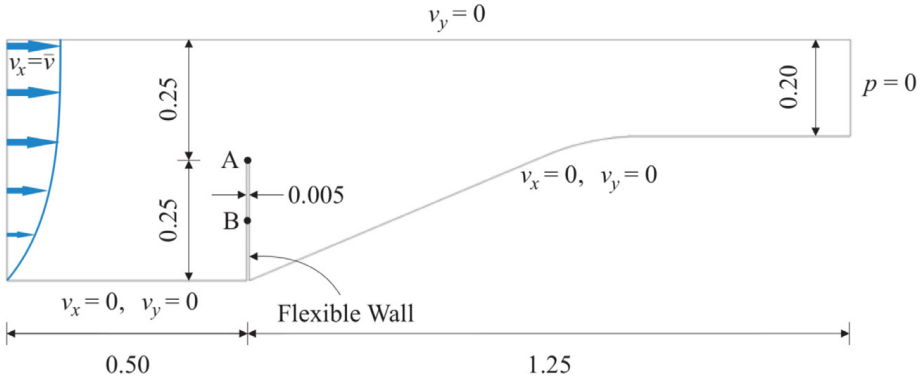


Fig. 8. 2D Mok benchmark. Geometry and boundary conditions.
Source: [83].

Bearing in mind that one of the main motivations for the use of an embedded FSI solver is to get rid of the ALE mesh motion problems, we only consider examples that could be challenging from the mesh motion perspective. Therefore, all the problems presented in this section involve structures that undergo large displacements and (or) rotations.

Besides this, embedded mesh methods can be also exploited to efficiently simulate membrane-like structures, allowing to circumvent all the meshing complexities (i.e. interface entities duplication) associated to the volume meshing of structures without internal volume. In this regard, we also present cases that imply the analysis of thin-walled bodies.

Taking this into consideration, on the one hand we solve three examples involving volumetric bodies. The first two are the well-known Mok and the Turek & Hron benchmarks. The third one is proposed by us to prove the capabilities of embedded mesh methods when dealing with arbitrary large rotations. It consists in a two-dimensional mixer with flexible blades. On the other hand, we also present two three-dimensional cases that involve thin-walled structures. The first one is a simple lid-driven cavity problem with a flexible membrane in the bottom. The second one is a potential industrial application of the presented FSI framework that is the FSI analysis of a four-point tent built as a system of beams and cables supporting a prestressed membrane.

5.1. 2D Mok benchmark

5.1.1. Problem description

The Mok benchmark was firstly presented in [65]. It consists in a convergent channel with an internal flexible wall. The main challenge of the problem is that the fluid and structure densities have a similar order of magnitude, leading to a strongly coupled problem with large interaction between the two fields.

The geometry and boundary conditions are depicted in Fig. 8. As it can be observed, the top edge is set as a slip boundary while the inferior one is no-slip. The pressure is imposed to zero in the right edge. A parabolic inlet defined as

$$v_x(y,t) = 4\bar{v}y(1-y)\ [m/s] \tag{10}$$

is applied to the left edge. $\bar{v}$ is a time dependent reference velocity such that

$$\bar{v} = \begin{cases} \frac{0.06067}{2}\left(1-\cos\frac{\pi t}{10}\right) & if\ t \leq 10 \\ 0.06067 & otherwise \end{cases} \tag{11}$$

The fluid and structure material properties are collected in Table 1. We highlight that the structural material response is modelled using a unit thickness plane stress linear elastic constitutive law. The simulation runs from 0s to 25 s with a fixed time step of 0.1s. The FSI coupling tolerance is set to 10^{-8}.

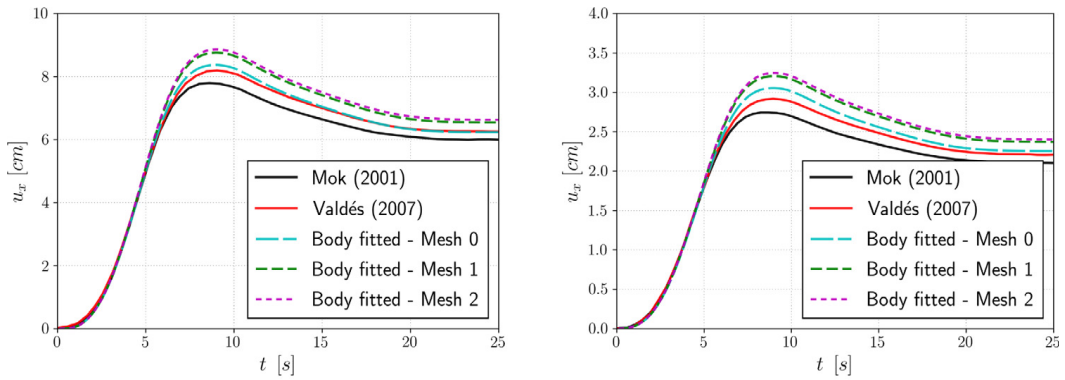
As it is done in the original work by Mok [65], as well as in [83] by Valdés, we take the structure control points A and B u_x displacement evolution as comparison magnitude. These two reference nodes are located at the top and centre points of the windward edge of the wall (Fig. 8).

Table 1
2D Mok benchmark. Material properties.

Structure		Fluid	
ρ_s	1.5e3 kg/m^3	ρ_f	956.0 kg/m^3
E	2.3e6 N/m^2	μ	0.145 Pa s
ν	0.45		

Table 2
2D Mok benchmark. Body fitted convergence study.

	Mok [65]	Valdes [83]	Body fitted		
			Mesh 0	Mesh 1	Mesh 2
$n_{el}^{\Omega_s}$	6 (quadratic)	6 (quadratic)	0.4k	1.6k	6.4k
$n_{el}^{\Omega_f}$	1.9k	6k	6.6k	18.8k	65.5k
$max(u_x^A)$ [cm]	7.789	8.190	8.363	8.756	8.856
$max(u_x^B)$ [cm]	2.743	2.917	3.052	3.204	3.244



(a) Point A. (b) Point B.

Fig. 9. 2D Mok benchmark. Control points u_x [cm] convergence study.

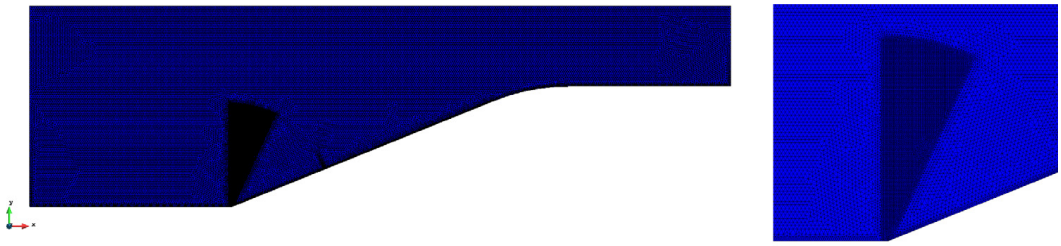
Some discrepancy can be observed with respect to the u_x results reported in [65] and [83]. As the mesh used in the current examples is however much finer than the one employed in the two references, we decided to perform a preliminary body-fitted mesh convergence study to obtain a comparable reference solution. Therefore, we solve the Mok benchmark problem with three different body fitted meshes: mesh 0, mesh 1 and mesh 2. Mesh 0 is generated as similar as possible to the one described in [83]. Mesh 1 is created by halving the element sizes of mesh 0. Similarly, mesh 2 is created by doing so with the element sizes of mesh 1.

Table 2 collects the number of structure and fluid elements ($n_{el}^{\Omega_s}$ and $n_{el}^{\Omega_f}$) employed in each case together with the maximum u_x obtained in both control points A and B. Furthermore, Fig. 9 compares our u_x results with the ones in [65] and [83]. We observe that the solution converges to larger values, specially when the peak displacement occurs, than the ones reported in [65] and [83].

Accordingly, from now on we consider as reference solution the one obtained with our body fitted solver and the finest mesh discretization level (mesh 2).

With these results in mind, and for the sake of a fair comparison, we set our embedded mesh as similar as possible to the finest body fitted one (mesh 2). Hence, the background fluid mesh (Fig. 10) is formed by 82k linear triangular elements (Q1P1)[2].

[2] Henceforth, we follow the same element nomenclature as in [14]. Therefore, Q1P1 states for a linear velocity and pressure interpolation element.



(a) Complete view. (b) Zoom view.

Fig. 10. 2D Mok benchmark.Fluid background mesh snapshots.

As it is done in the body fitted case, the mesh is refined in the region close to the structure in order to properly represent the embedded coupling interface (Fig. 10(b)). However, we note that in the embedded case it is not possible to "a priori" know the location of the structure interface. As a consequence, the mesh refinement needs to be done in the region in where the embedded structure is expected to move across, rather than only refining the geometry edges of interest. This yields thus a slightly larger number of elements for the same refinement level.

Finally, we use the same structure mesh employed in the finest body fitted case (mesh 2). Hence, the embedded structure mesh is also meshed with a total of 6.4k linear quadrilateral Total Lagrangian elements.

5.1.2. Results assessment

Figs. 11 and 12 present some snapshots of the v_x and p solution fields together with the corresponding zero isosurface of the distance function. As expected, the structure undergoes a large displacement, whose value is maximum at the end of the first transition phase. Once the inlet velocity becomes constant in time, the solution reaches a steady state, whose maximum deflection value is around 7 cm.

Taking into account that this example involves a low Re number flow, we decide to solve the reference body fitted case (mesh 2) with and without considering the wall shear component in the structure load. The obtained solutions are compared in Figs. 13 and 14. As it can be observed, the embedded u_x time evolution in both control points is in very good agreement with the body fitted reference ones (Figs. 13(a) and 14(a)). Nonetheless, we note minor differences in the peak displacement values (Figs. 13(b) and 14(b)). We also notice that such differences are almost imperceptible when the embedded solution is compared to the body fitted one without the wall shear component (denoted as "Body fitted (mesh 2) - p" in Figs. 13 and 14).

On top of the wall shear stress contribution, we believe that the small deviations in the results are possibly due to the intrinsic particularities of the embedded CFD formulation, which we recall inherently smooths sharp corners as these cannot be represented in terms of a continuous distance function. Considering that the control point A coincides with one vertex of the structure, minor discrepancies in the values can be expected.

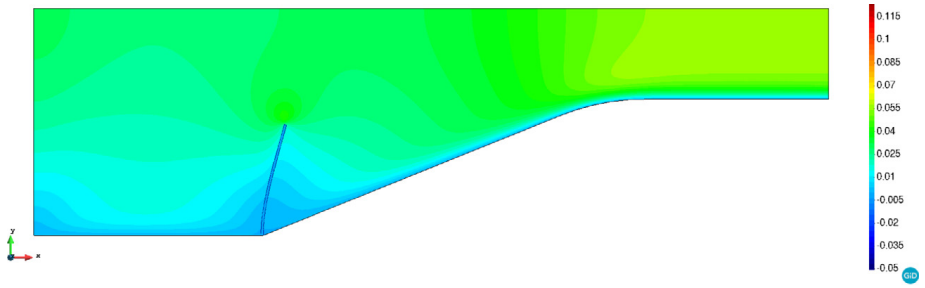
With regard to the computational efficiency, we also compare the number of required FSI coupling iterations. The results depicted in Fig. 15 show that the embedded solver is slightly more efficient than the body fitted one. We think that these results cannot be taken as a general rule. Instead, what we can asseverate is that the use of an embedded framework in the fluid domain does not deteriorate the convergence of the FSI coupling.

Finally, we would like to point out that this problem can be alternatively, and possibly more efficiently, solved by modelling the flexible wall as a non-linear beam and using discontinuous level set based formulation in the fluid.

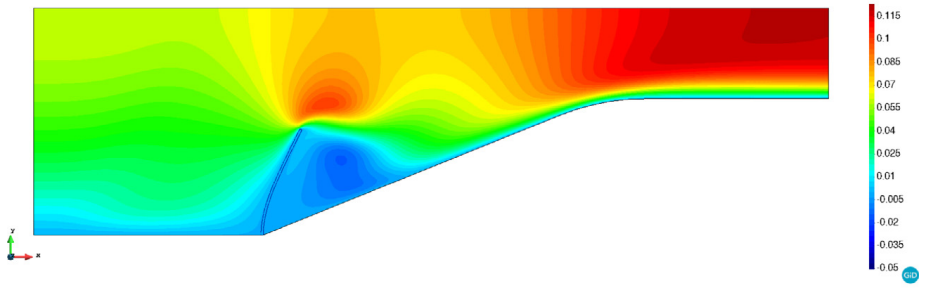
5.2. 2D Turek & Hron benchmark — FSI2

5.2.1. Problem description

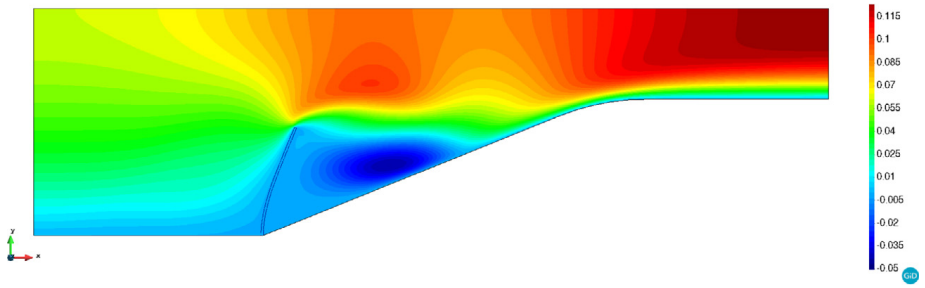
The second example we present is the well-known Turek & Hron benchmark (also known as Turek benchmark). In their original work [84], Turek and Hron present three FSI variants, being the second and the third one the most challenging ones. Since we want to exploit the capabilities of embedded formulations to deal with large boundary displacements, we choose the second variant, denoted by the original authors as FSI2. In this case the problem settings are selected in such a way that the structure undergoes extremely large deflections, being thus a tough example from the mesh motion perspective.



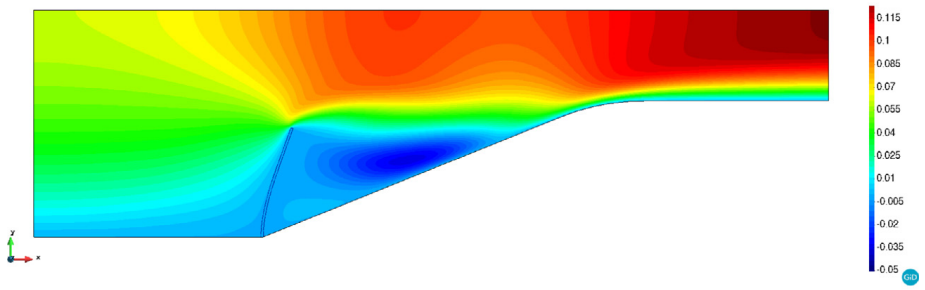
(a) t = 5 s.



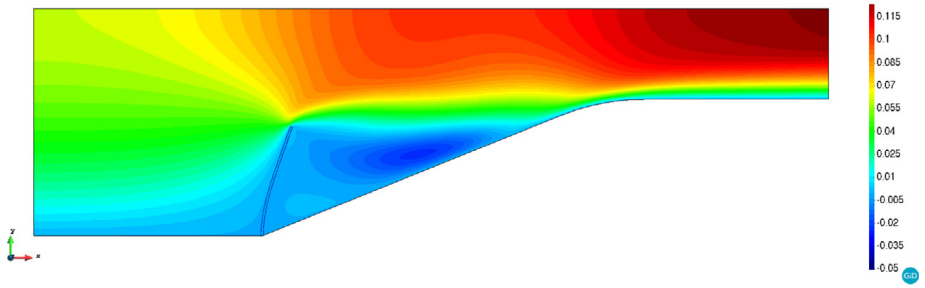
(b) t = 10 s.



(c) t = 15 s.

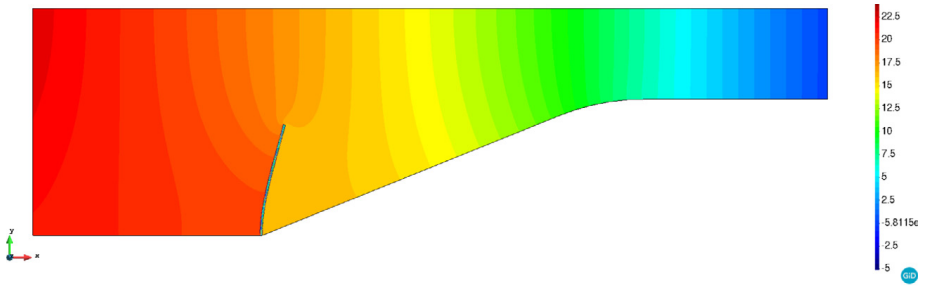


(d) t = 20 s.

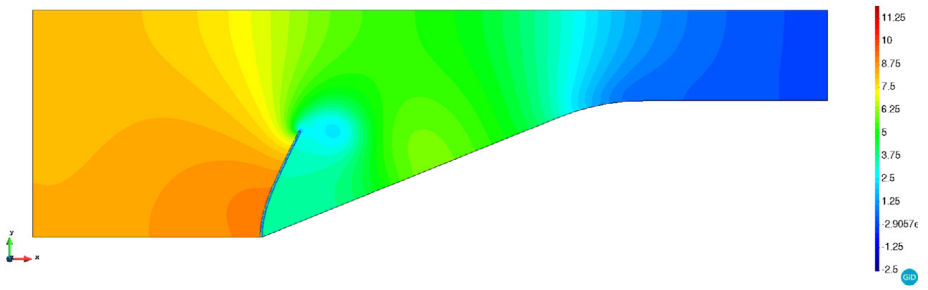


(e) t = 25 s.

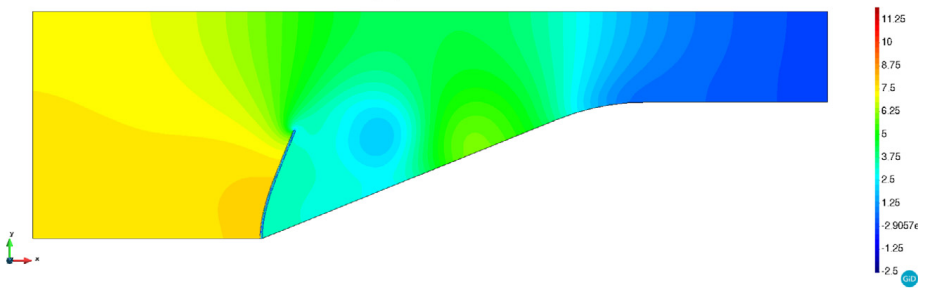
Fig. 11. 2D Mok benchmark. Level set zero isosurface and v_x velocity field [m/s] at different time steps.



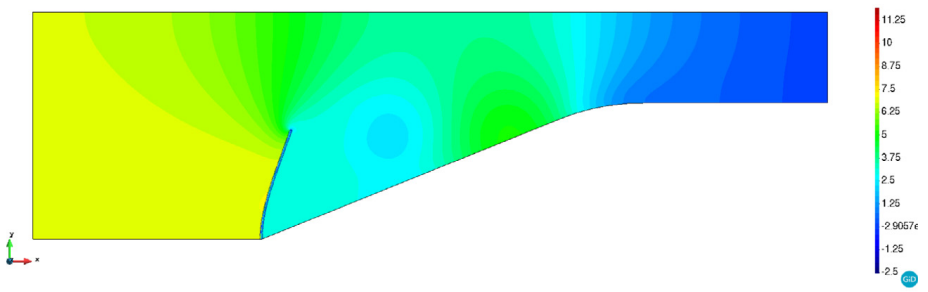
(a) t = 5 s.



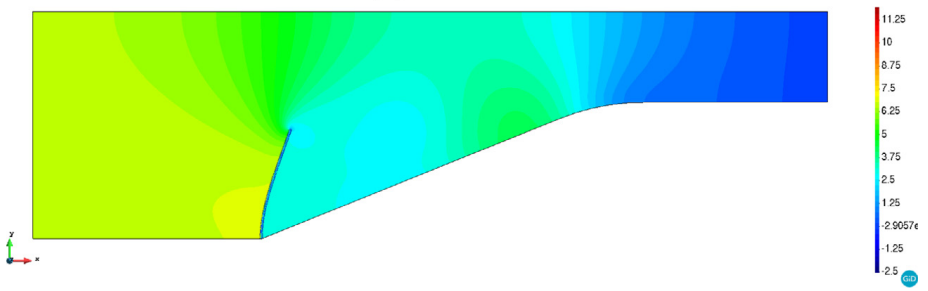
(b) t = 10 s.



(c) t = 15 s.

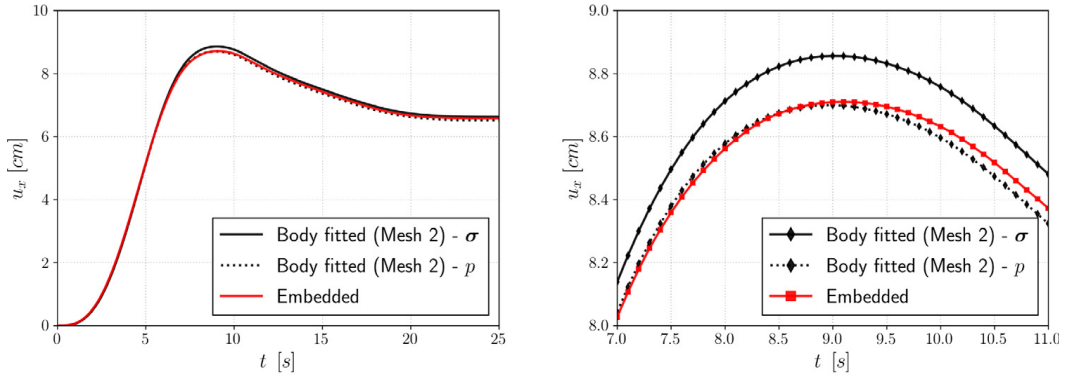


(d) t = 20 s.



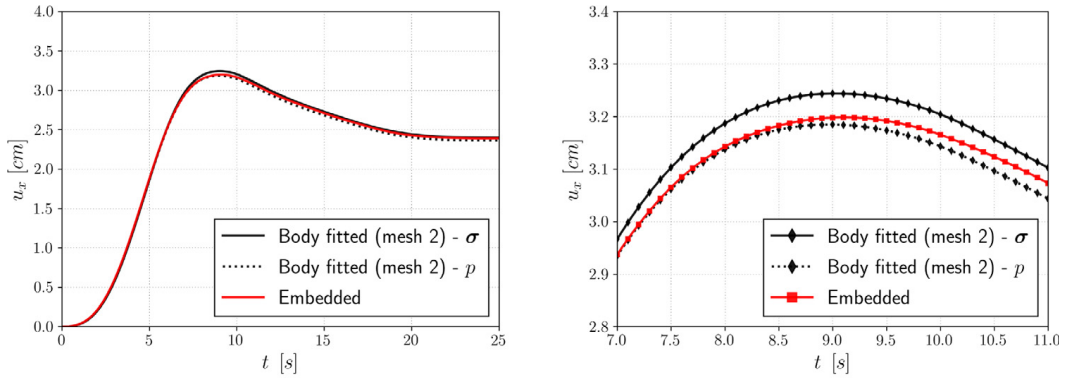
(e) t = 25 s.

Fig. 12. 2D Mok benchmark. Level set zero isosurface and pressure field [Pa] at different time steps.



(a) Complete time history. (b) Zoom from 7 to 11s.

Fig. 13. 2D Mok benchmark. Control point A horizontal displacement u_x [cm].



(a) Complete time history. (b) Zoom from 7 to 11s.

Fig. 14. 2D Mok benchmark. Control point B horizontal displacement u_x [cm].

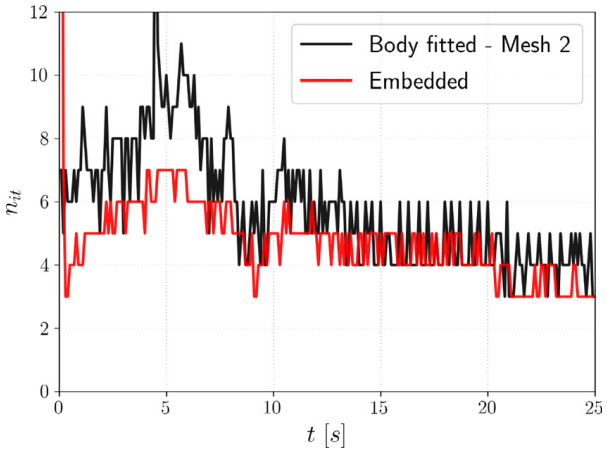
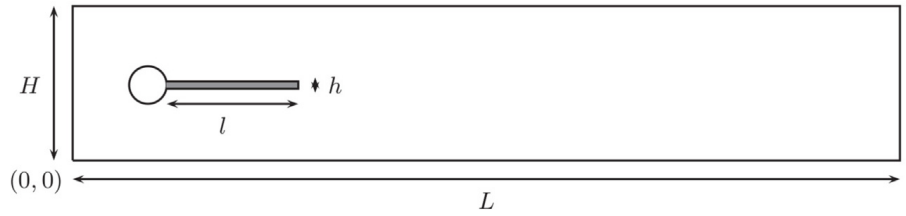
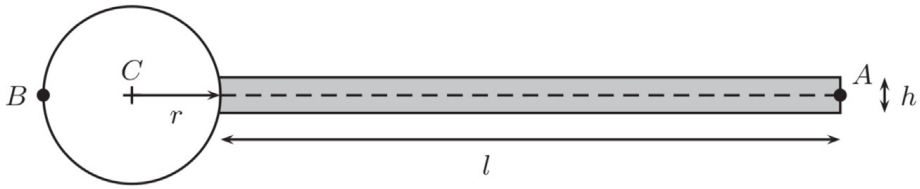


Fig. 15. 2D Mok benchmark. FSI non-linear iterations history.

The problem geometry consists in a rigid cylinder inside a straight channel (Fig. 16(a)). Attached to the cylinder, there is a flexible cantilever beam (Fig. 16(b)). The problem geometry is depicted in Fig. 16 and its dimensions are



(a) Fluid domain.



(b) Structure domain.

Fig. 16. 2D Turek & Hron benchmark — FSI2. Problem geometry.
Source: [84].

Table 3
2D Turek & Hron benchmark — FSI2. Material properties.

Structure		Fluid	
ρ_s	10*e*3 kg/m^3	ρ_f	1000 kg/m^3
μ_s	0.5*e*6 kg/ms^2	μ_f	1 Pa s
ν	0.4		

H 0.41 m, L 2.5 m, h 0.02 m, l 0.35 m and r 0.05 m. The centre of the cylinder is placed at (0.2,0.2)m while the coordinates of control points A (initial position) and B are (0.6,0.2)m and (0.15,0.2)m.

Concerning the boundary conditions, the pressure is fixed to zero along the right edge of the channel. The velocity is fixed to zero in both top and bottom edges as well as in the cylinder boundary. In the left edge, the parabolic inlet function

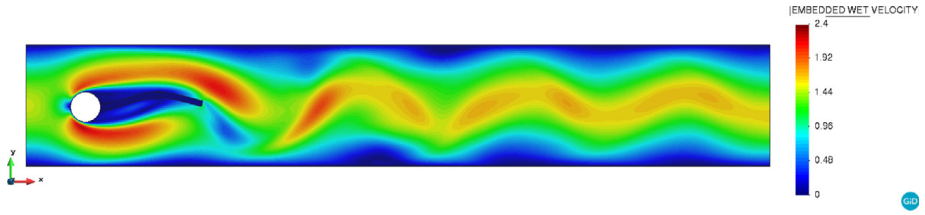
$$v_x(y) = 1.5\bar{v}\frac{y(H-y)}{\left(\frac{H}{2}\right)^2} = 1.5\bar{v}\frac{4}{0.1681}y(0.41-y)\ [m/s] \tag{12}$$

is applied. $\bar{v}$ is a reference velocity that equals 1m/s in the FSI2 case. The previous inlet function is multiplied by an initial ramp-up function to obtain the final velocity profile

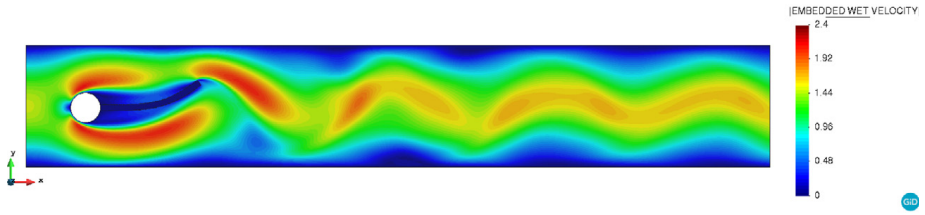
$$v_x(y,t) = \begin{cases} v_x(y)\frac{1-cos\left(\frac{\pi}{2}t\right)}{2} & if\ t \leq 2 \\ v_x(y) & otherwise \end{cases} \tag{13}$$

As is done all throughout the paper, the fluid material response is modelled using a Newtonian constitutive law. Conversely, a Kirchhoff–Saint-Venant material model is used in the structure. The material properties of both the fluid and the structure are collected in Table 3. The problem is run for 20 s using a time step of 2e-3s, which equals the one used in [84]. The non-linear FSI tolerance is set to 10^{-8}.

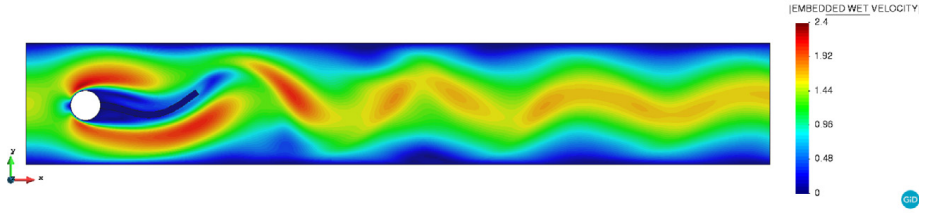
Finally, the computational domain is meshed using linear triangular elements (Q1P1) in the fluid domain and linear quadrilateral Total Lagrangian elements in structural one. As in this case we aim to compare the embedded solution to the one obtained with a body fitted alternative solver, the background fluid mesh is set as similar as possible to the body fitted one. Therefore, the number of elements of the embedded background and body fitted fluid meshes is 63.5k and 28k, which we consider reasonably close taking into account that elements that fall within the embedded structural domain are deactivated. Besides this, we also note that, as in the previous example, the differences in the total number of elements are due to the local refinement we do in the leeward side of the cylinder.



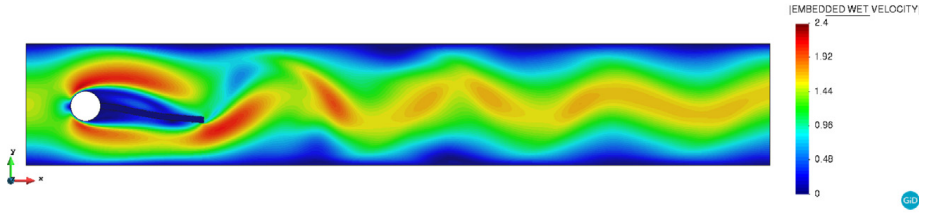
(a) $t = t_0$ s.



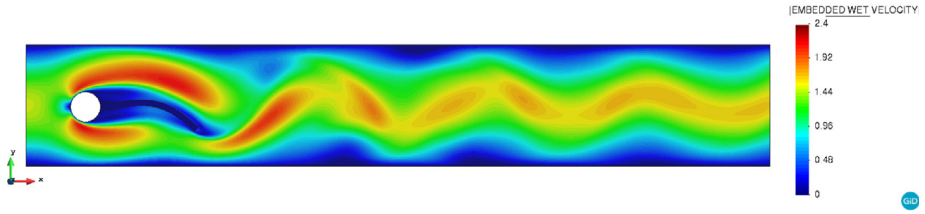
(b) $t = t_0 + 0.2T$ s.



(c) $t = t_0 + 0.4T$ s.



(d) $t = t_0 + 0.6T$ s.



(e) $t = t_0 + 0.8T$ s.

Fig. 17. 2D Turek & Hron benchmark — FSI2. Level set zero isosurface and velocity norm $\|\mathbf{v}\|$ [m/s] for a period of oscillation T.

By doing this we ensure that the level set function is computed with an accuracy level somewhat comparable to the body fitted discretization. With regard to the structure domain, the same structured mesh made up with 2.6k elements is used in both the embedded and the body fitted simulations.

5.2.2. Results assessment

Figs. 17 and 18 collect snapshots of the obtained embedded velocity and pressure fields together with the level set zero isosurface representing the structure skin. Because the solution is periodic in this case, we present these magnitudes for a complete period of oscillation (T) starting at a reference time t_0. It can be observed that the

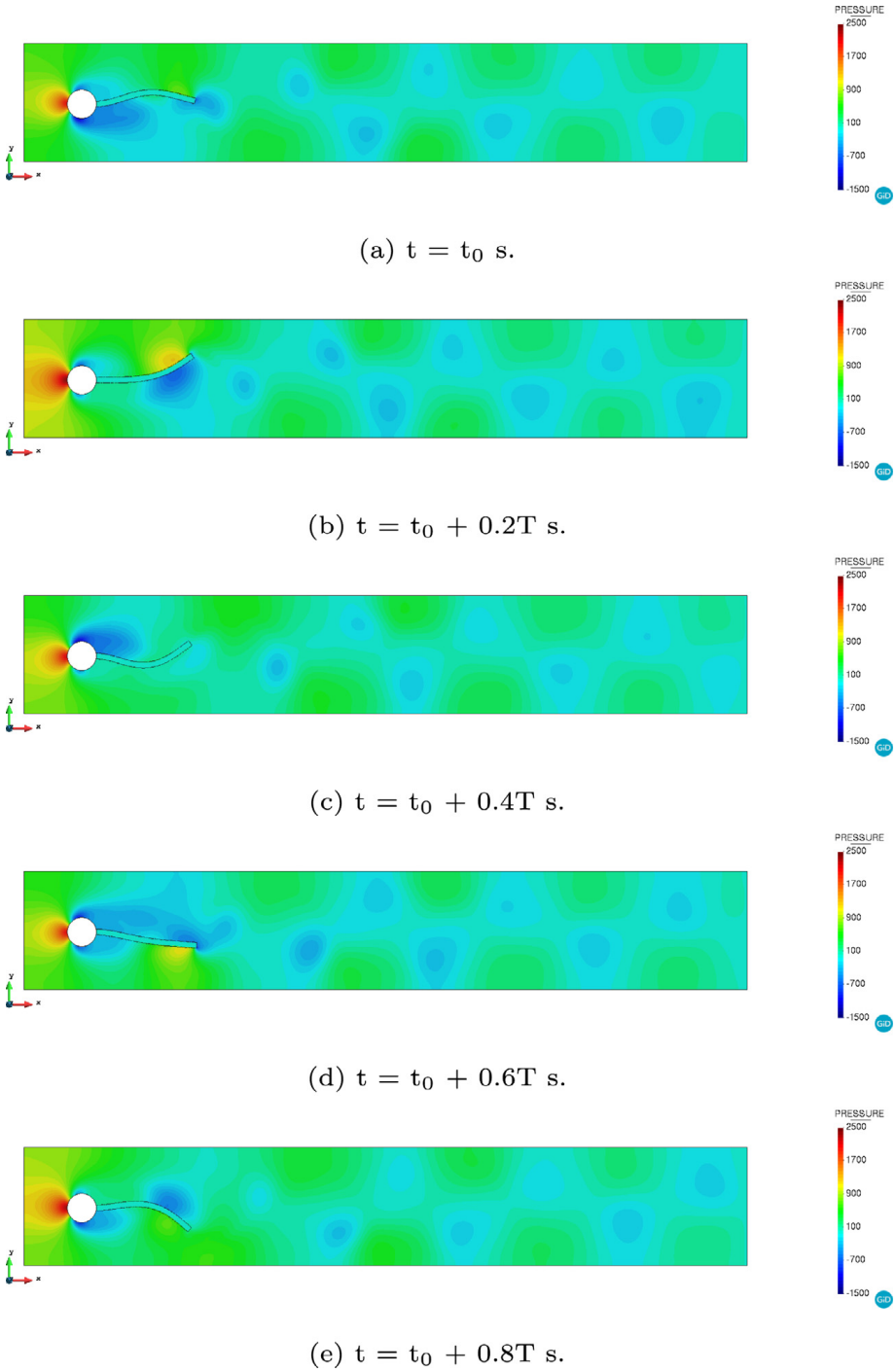
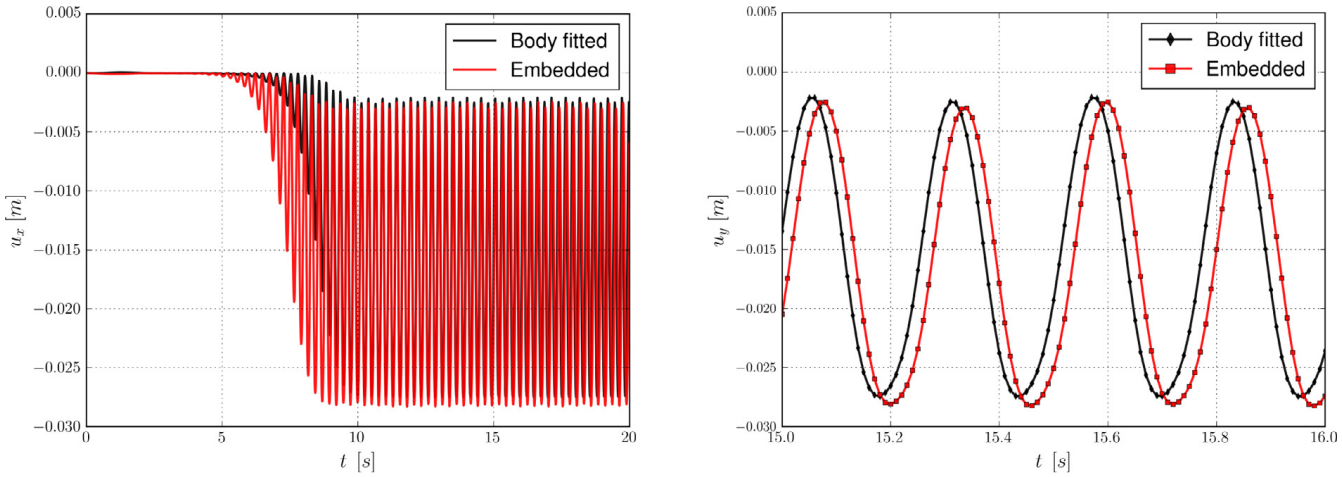


Fig. 18. 2D Turek & Hron benchmark — FSI2. Level set zero isosurface and pressure p [Pa] for a period of oscillation T.

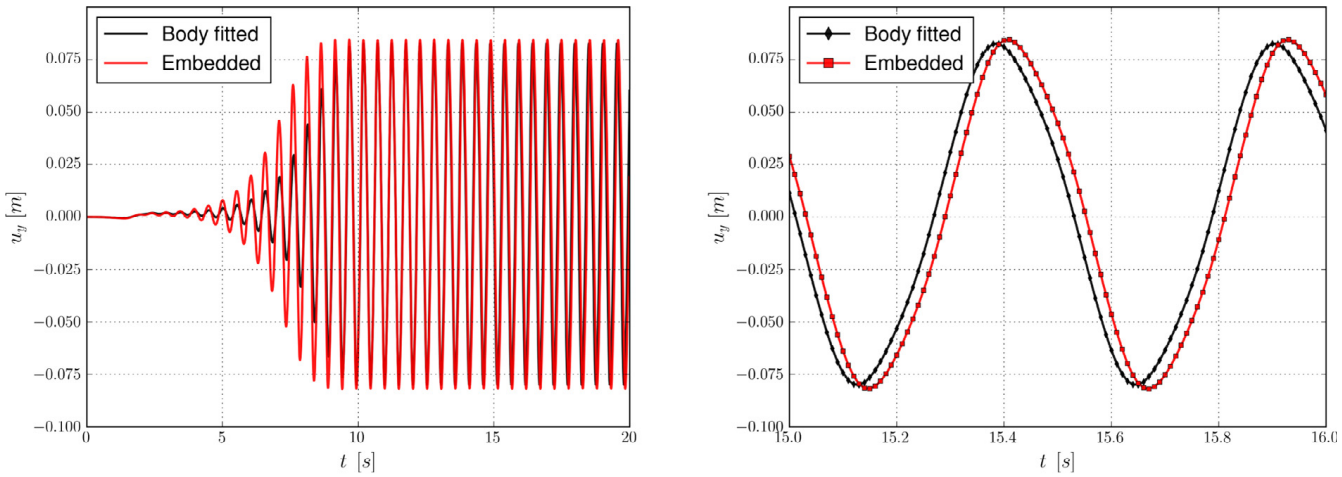
periodic vortex shedding generates a pressure pattern that yields the expected large deflections in the cantilever beam.

The structure deflection evolution is compared with the body fitted solver one in Figs. 19 and 20. Even though the transition phase is shorter in the embedded case, barely differences between both solutions can be observed once the periodic regime is reached. Complementary, Figs. 19(b) and 20(b) zoom in on the embedded and body fitted solutions from time 15 s to 16 s, confirming that the peak values and oscillation periods are almost identical in both solutions.



(a) Complete time history. (b) Zoom from 15 to 16s.

Fig. 19. 2D Turek & Hron benchmark — FSI2. Control point A horizontal displacement u_x [m].



(a) Complete time history. (b) Zoom from 15 to 16s.

Fig. 20. 2D Turek & Hron benchmark — FSI2. Control point A vertical displacement u_y [m].

Table 4 collects the mean and amplitude values of the displacement for the last period of oscillation. These values are in good agreement with the reference ones reported in [84], being the embedded solution values slightly larger.

Finally, we would like to add that for this test case the embedded solver, is remarkably robust. On the contrary, it tends to be challenging to achieve a similar robustness when using the body fitted approach, mainly because of the mesh deformation solver, which tends to yield excessively distorted (or even inverted) elements due to the large deformations and rotations of the tail. Conversely, the embedded solver, which worked out of the box, bypasses all these mesh motion issues.

5.3. 2D mixer with flexible blades

5.3.1. Problem description

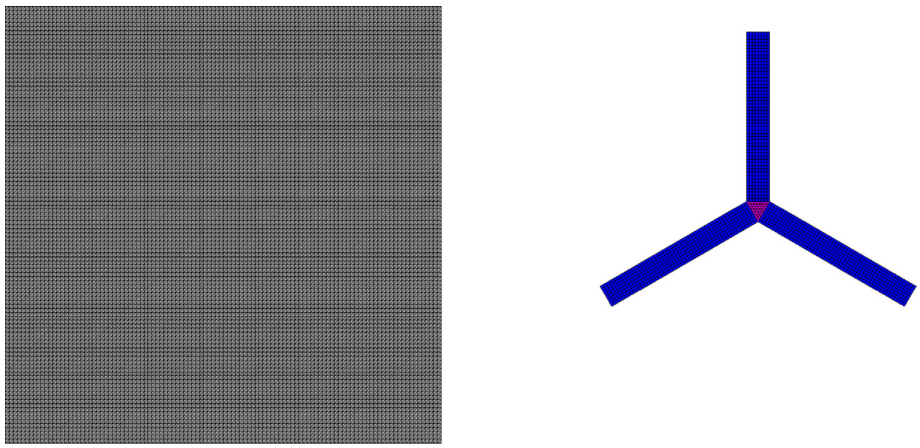
So far we have presented two examples that are considered as reference benchmarking test cases in the FSI community. Although they served us to prove the accuracy of our embedded proposal, they do not highlight the

Table 4
2D Turek & Hron benchmark — FSI2. Point A displacement values [m].

	$u_x \ [\times 10^{-3}]$	$u_y \ [\times 10^{-3}]$
Body fitted	-14.76 ± 12.64	1.29 ± 81.21
Embedded	-15.29 ± 12.77	1.31 ± 83.15
Turek & Hron [84]	-14.58 ± 12.44	1.23 ± 80.60

Table 5
2D mixer with flexible blades. Material properties.

Structure		Fluid	
ρ_s	1.0 kg/m^3	ρ_f	1.0 kg/m^3
E	1.0e2 N/m^2	μ	1.0$e-2$ Pa s
ν	0.3		



(a) Fluid domain. (b) Structure domain.

Fig. 21. 2D mixer with flexible blades. Problem geometry with mesh.

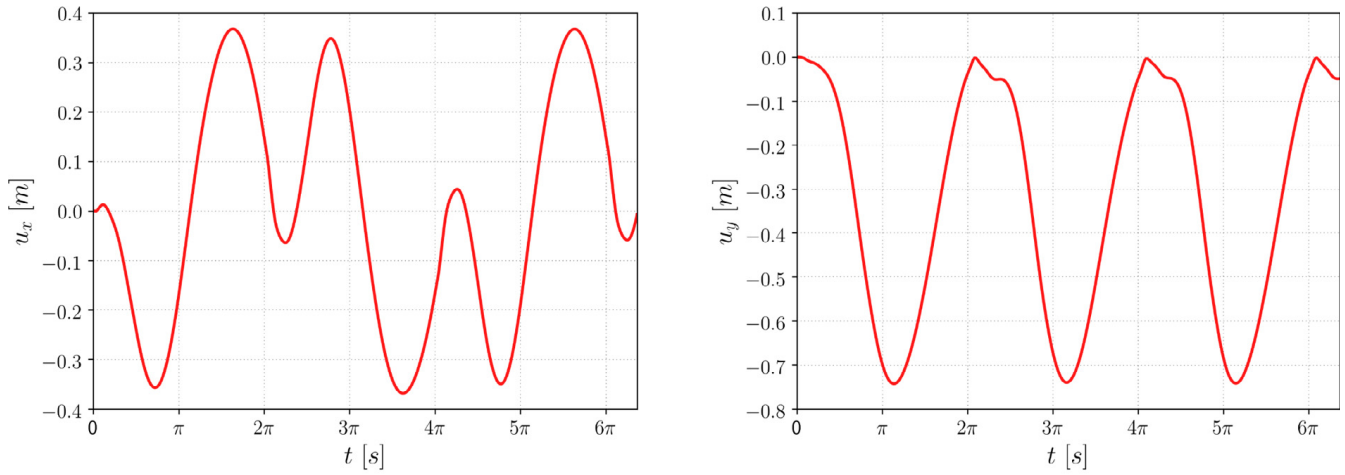
specific capabilities of the new method since they can also be solved by employing ALE approaches. In order to prove the extended scope of application of the proposed technique, we present next an example that involves extremely large rotations, and that would be thus impossible to solve by using body fitted approaches.

The problem is conceived as a 2D idealization of a turbine mixer with clockwise–anticlockwise alternate rotation. Hence, the fluid geometry consists in a unit length square domain whose centre coincides with the coordinates origin. The structure geometry represents the three blades of the mixer. These are also centred in the coordinates origin and aligned such that there is a 120° radial symmetry. The blades are 0.375 m long (measured from the origin) while their thickness is 0.05 m.

With regard to the BCs, a slip BC is enforced in the outer walls while a no-slip behaviour is assumed in the embedded FSI coupling interface. Besides this, we also fix the pressure to zero in the entire wall to obtain a radially symmetric solution. Concerning the structure domain, a rigid body rotation with angular velocity ω equal to ± 1rad/s is imposed in the centre of the blades (pink region in Fig. 21(b)) to emulate the spinning of the mixer, which changes its direction after each lap (2π rad) is completed.

The material properties are collected in Table 5. We note that the stiffness of the blades is intentionally set to an unrealistic value. Although this value might be unfeasible when comparing with a real mixer, it allows to test our solver under the action of not only large rotations but also displacements. On top of that, we also set $\rho_f = \rho_s$ to yield a strongly coupled behaviour. As in the previous example, the material response is modelled using a Newtonian and Kirchhoff–Saint-Venant constitutive models.

The problem is run for 20 s to reproduce three complete rotations (anticlockwise, clockwise and anticlockwise) of the mixer. The time step is 0.01s while the non-linear FSI tolerance is 10^{-7}.



(a) u_x [m]. (b) u_y [m].

Fig. 22. 2D mixer with flexible blades. Control point (0,0.375)m displacement evolution.

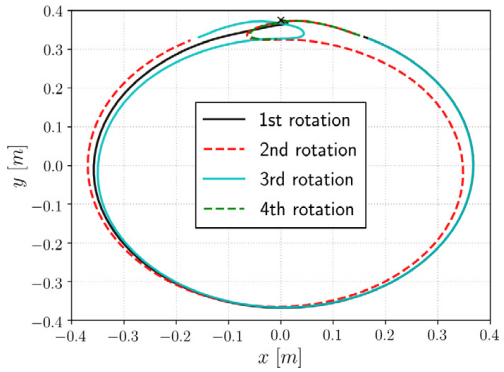


Fig. 23. 2D mixer with flexible blades. Control point tracking. Solid lines represent the anticlockwise rotation cycles. Dashed lines represent the clockwise rotation ones. The black X marker represents the initial position of the control point.

Finally, both the fluid and the structure domains are meshed using structured meshes. On the one hand, we use a structured mesh with 110 perimeter subdivisions in the fluid domain. This amounts a total of 24,2k linear triangular elements (Q1P1). On the other hand, each one of the blades of the rotor is meshed with a 8x39 grid made with linear quadrilateral Total Lagrangian elements. In the rigid body motion region of the rotor we use 64 triangular elements to enforce the rotation.

5.3.2. Results assessment

Prior to any results assessment, we note that what we present in the following is a qualitative analysis of the obtained solution due to the lack of reference data.

Hence, we firstly focus on the displacement evolution of one of the three blades of the mixer. To this purpose, we present in Fig. 22 the time evolution of one control point in the blade that is initially in a vertical position. The initial coordinates of such control point are (0,0.375)m and coincide with the short edge centre point of the blade. We note that as the problem has radial symmetry, there is no need to assess the three blades as they behave exactly the same.

From the results in Fig. 22, we observe that after each n rotation is completed (values at time $n2\pi$) the tip of the blade almost recovers its original position. This can be also observed in Fig. 23 which tracks the evolution of the control point coordinates.

To better comprehend the deformation of the flexible blades, we present in Fig. 24 a collection of snapshots that depict the structure during the second rotation (ω -1rad/s and $2\pi \leq t \leq 4\pi$). In Fig. 24(a) it can be observed that at

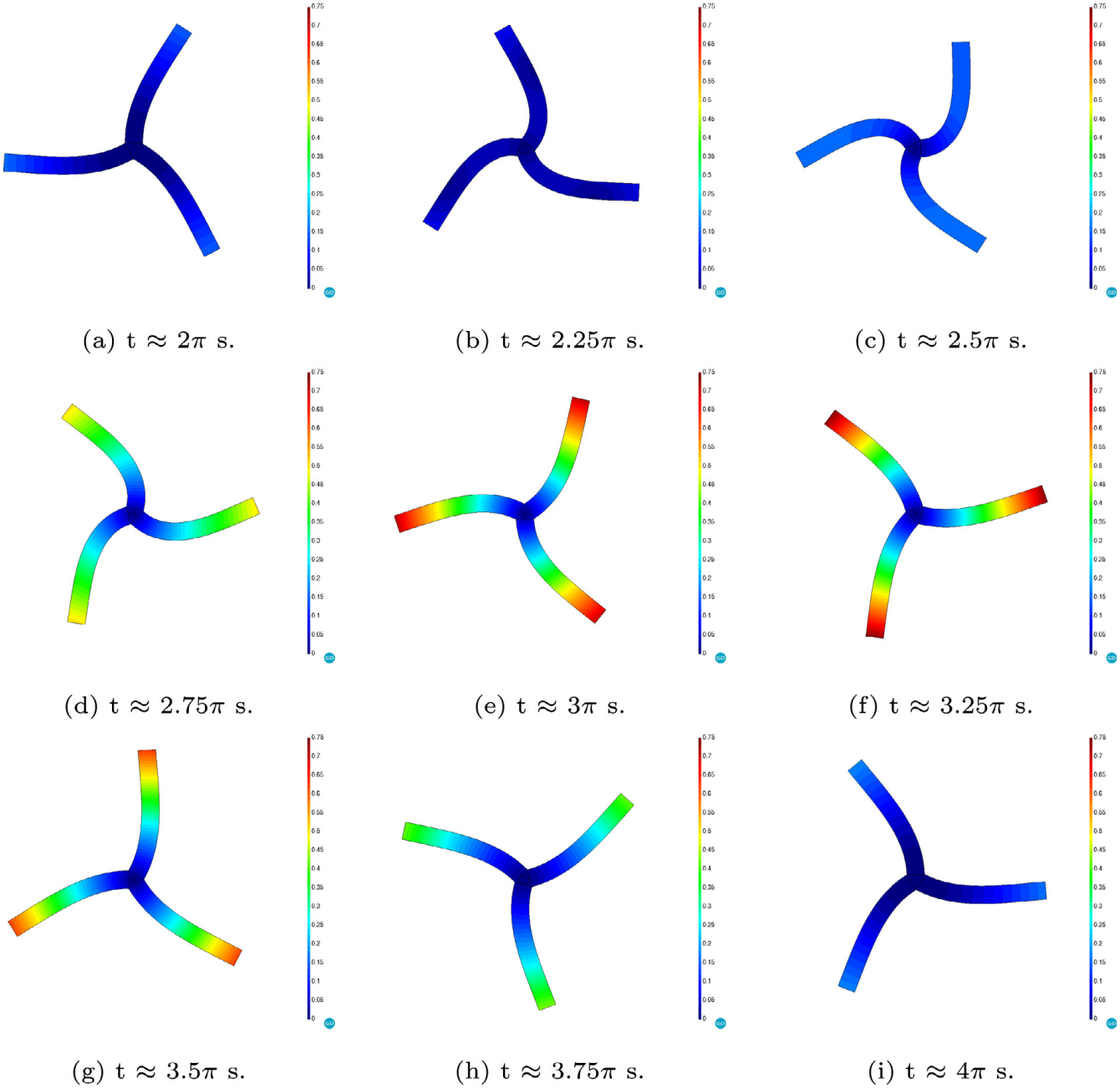


Fig. 24. 2D mixer with flexible blades. Structure displacement norm $\|\mathbf{u}\|$ [m] snapshots during the second (clockwise) rotation ($2\pi \leq t \leq 4\pi$ $[s]$).

the end of the first anticlockwise rotation (t $\approx 2\pi$ s) the structure almost recovers the initial spatial configuration. Then, the rotation direction is suddenly inverted, thus inducing a high acceleration in the structure. As a consequence of its small stiffness, such acceleration yields a large deformation in the blades, (Figs. 24(b) and 24(c)) that is recovered as the rotation evolves.

This transient-to-steady cycle is repeated at each rotation and can be also observed in the fluid domain. As it can be noted in Figs. 25 and 26, at the end of each rotation the flow reaches a steady solution, which presents a radial velocity pattern that grows from zero to a maximum value around 0.4m/s in the ends of the blades (Fig. 25(a)). However, when the rotation changes its direction the flow tends to accordingly do so. Hence, the sudden inversion of the rotation generates three initial pressure (Fig. 26(b)) and velocity (Fig. 25(b)) peaks in the three corresponding ends of the flexible blades.

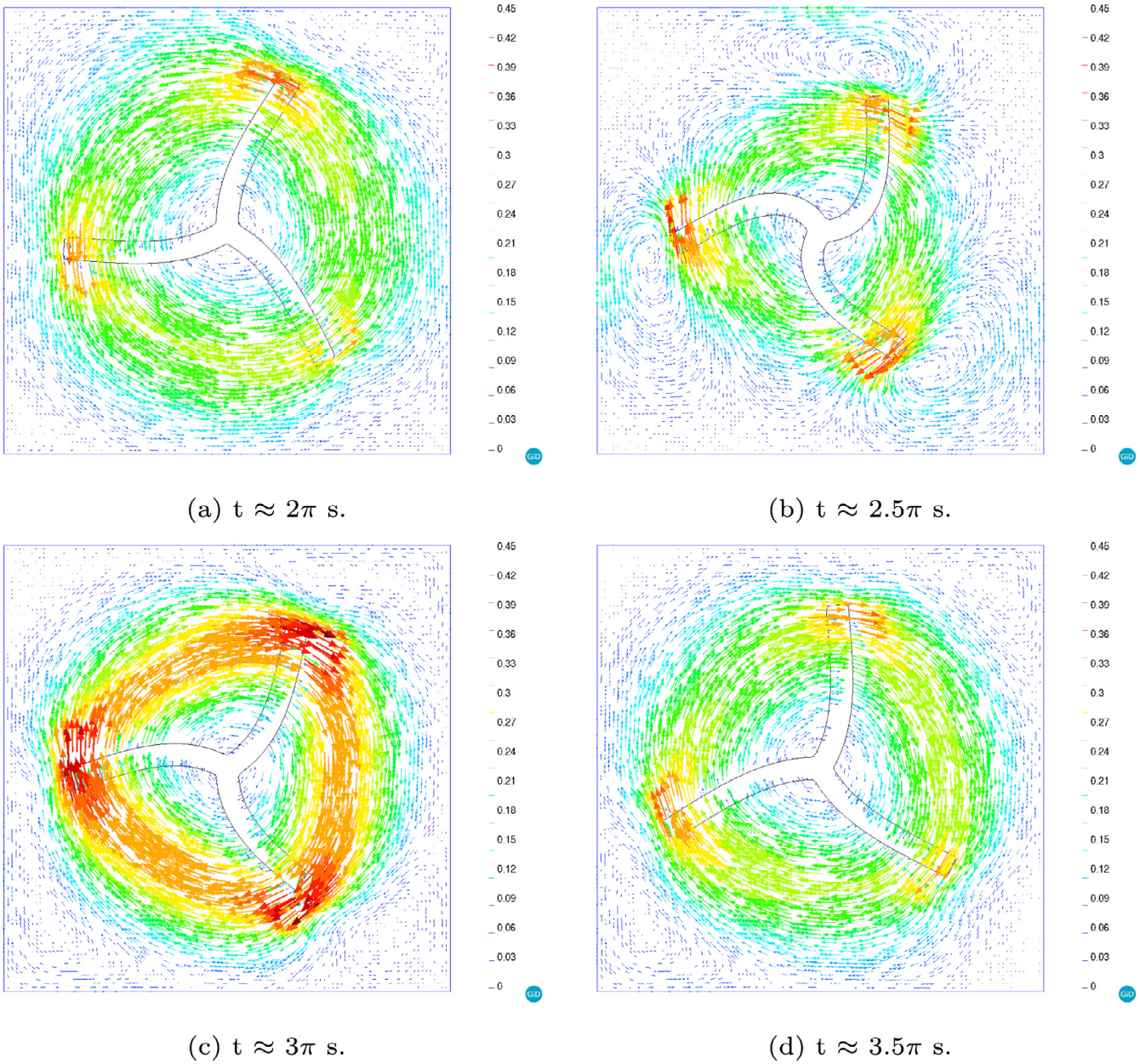


Fig. 25. 2D mixer with flexible blades. Level set zero isosurface and velocity vector field [m/s] snapshots during the second (clockwise) rotation ($2\pi \leq t \leq 4\pi$ $[s]$).

Last but not least, we report the history of FSI non-linear coupling iterations required to reach convergence in Fig. 27. Although the number of FSI iterations varies as the problem evolves, it remains always bounded between 4 and 8 iterations, being in most cases 6 or 7 iterations. This is a remarkable result considering that the densities of the fluid and structure are the same.

Once arrived to this point, we think that the obtained solution feasibly represents the physics of the experimental set up, proving again the correctness of the solver. On top of that, we highlight the capability of our proposal to efficiently solve not only large displacement problems, but also arbitrary extremely large rotation ones. Considering that to solve this problem with a traditional ALE-based solver would definitively require remeshing, we dare say that this is a proof of the enhanced robustness that embedded approaches have to face such extremely large rotation scenarios.

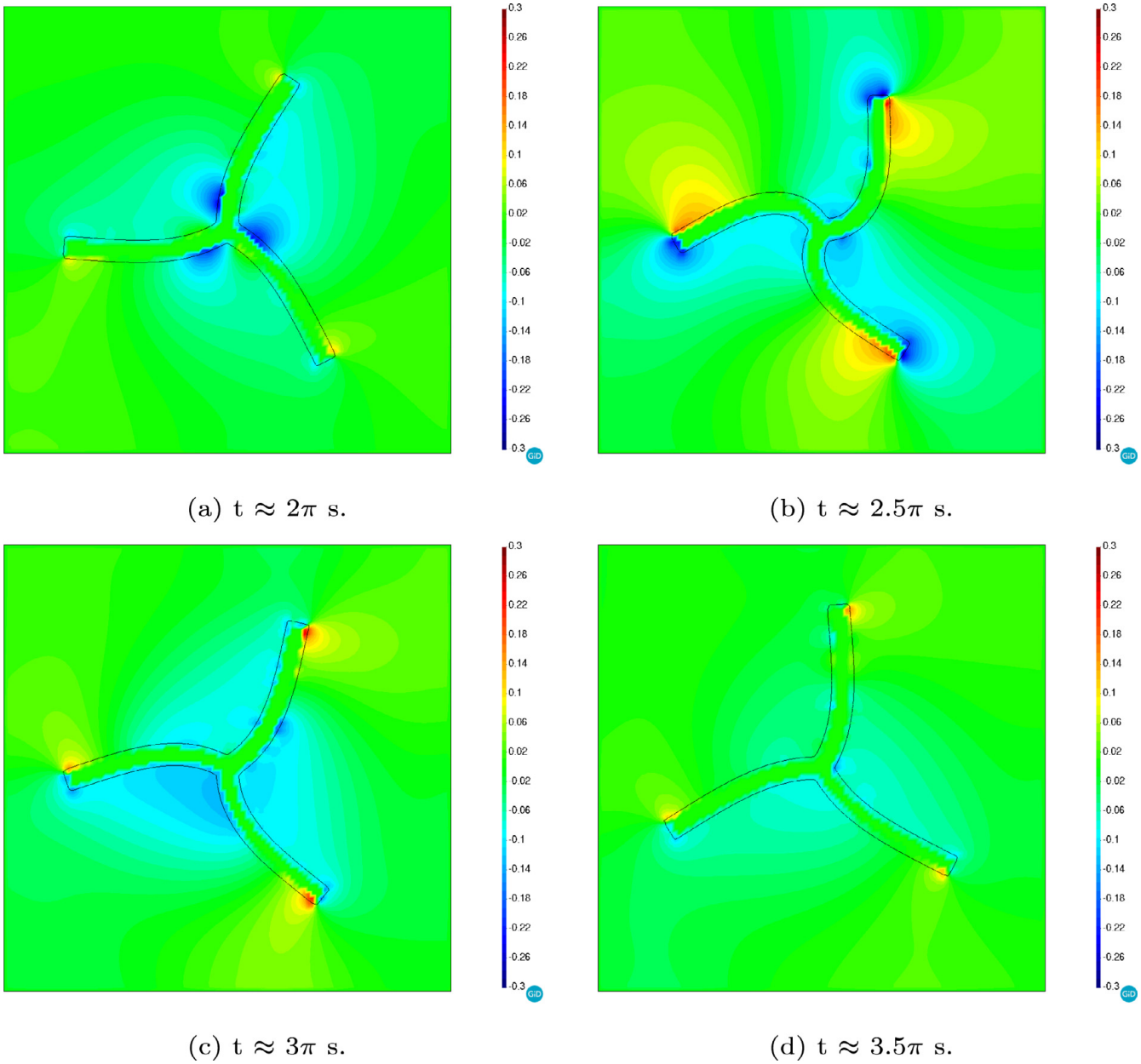


Fig. 26. 2D mixer with flexible blades. Level set zero isosurface and pressure field [Pa] snapshots during the second (clockwise) rotation ($2\pi \leq t \leq 4\pi$ $[s]$).

5.4. 3D lid-driven cavity with flexible shell

5.4.1. Problem description

The most distinguishable feature of the example we present in this section is that it involves a thin-walled structure. Considering that one of the advantages of embedded mesh methods is the capability to efficiently deal with complex volumeless bodies, this example aims to test the performance of the proposed FSI solver when these structural typologies are analysed. On top of that, this example also tests the implementation in a 3D scenario.

The problem geometry is taken from the cavity with a thin bottom shell benchmark presented in [65] and consists in a 1x1x1 m fluid cavity (Fig. 28(a)) whose bottom face is a 2.0e−3 m thickness flexible membrane (Fig. 28(b)).

Assuming that the bottom flexible membrane is placed in the 0 m z-coordinate, the oscillatory velocity profile

$$v_x = \frac{z - 0.875}{0.125} \left(1 - cos\left(0.4\pi t\right)\right) \ [\mathrm{m/s}] \tag{14}$$

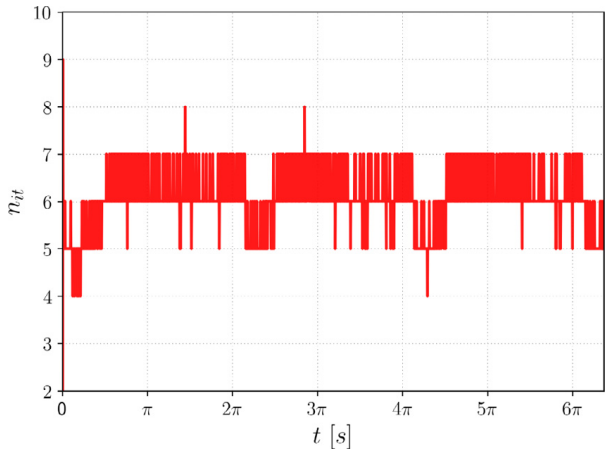


Fig. 27. 2D mixer with flexible blades. FSI non-linear iterations history.

Table 6
3D lid-driven cavity with flexible shell. Material properties.

Structure		Fluid	
ρ_s	500.0 kg/m^3	ρ_f	1.0 kg/m^3
E	250.0 N/m^2	μ	0.01 Pa s
ν	0.0		

is imposed in the surfaces highlighted in Fig. 28(c) (the v_y and v_z components are fixed to zero). The pressure is fixed to zero in the surface depicted in Fig. 28(d). A no-slip BC is set in the remaining lateral faces.

Concerning the constitutive behaviour, a Newtonian model is used in the fluid domain while an elastic one is used in the shell structure, which has no initial prestress. The corresponding material properties are collected in Table 6.

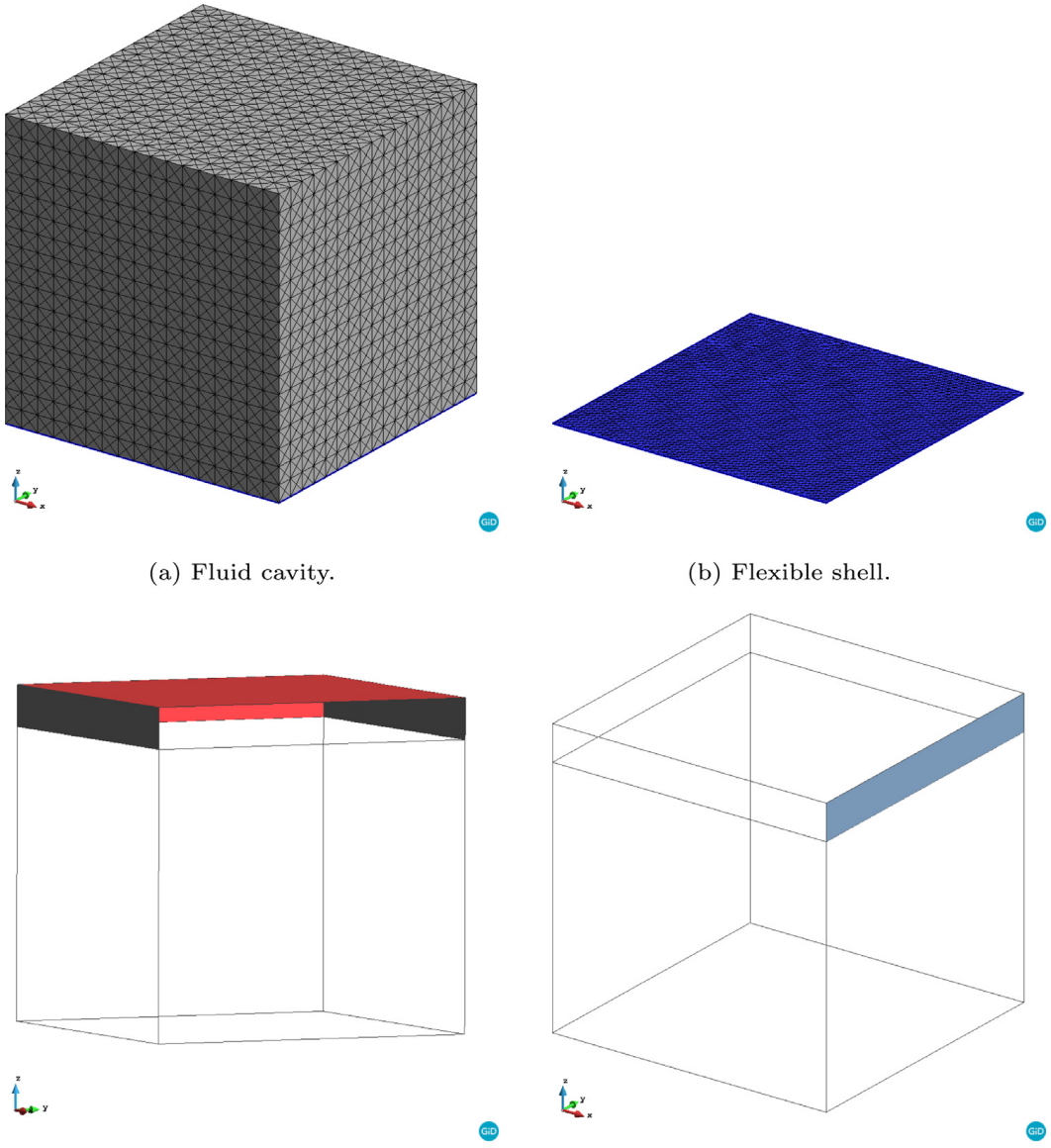
The same fluid and structure meshes are employed in the body fitted and embedded simulation. The fluid volume mesh (Fig. 28(a)) has around 117k linear tetrahedra elements (Q1P1) while the structure one (Fig. 28(b)) is made with 5.5k DKT-OPT shell elements [85]. The problem is run for 60 s with a time step equal to 2.5e−2s. The FSI coupling tolerance is set to 10^{-6}.

5.4.2. Results assessment

Fig. 29 depicts the vertical displacement evolution (u_z) of the shell midpoint. As expected, the deformation becomes periodic after a first transition phase. The obtained solution is compared with the one in [83] and also with the one obtained with the reference body fitted solver. We observe that both solutions are similar, in amplitude and period of oscillation, to the one reported by Valdés in [83]. However, there are noticeable differences in the peak oscillation values. Taking into account the good agreement between the embedded and body fitted solutions, we believe that such differences come from the mesh discretization, which is much finer in our case.

Furthermore, Table 7 compares the values that characterize the oscillatory regime. We highlight the remarkably good correlation of the oscillation period (T) and minimum u_z values between the embedded and body fitted solutions. Nonetheless, there is a noticeable difference in the maximum u_z value. This is explained in Fig. 30, which compares the body fitted and embedded pressure fields when the maximum structure deflection occurs. As it can be observed, the mass below the bottom membrane is considered in the embedded problem (Fig. 30(b)). On the contrary, this is not taken into account in the body fitted case (Fig. 30(a)). The movement of such extra mass turns into a vertical overpressure, whose maximum value appears when the bottom shell undergoes the maximum deflection, that explains the differences in the maximum u_z values.

Finally, we present two snapshots of the embedded shell displacement norm $\|\mathbf{u}\|$ during the periodic oscillation regime (Fig. 31). The first one (Fig. 31(a)) depicts the deformed geometry of the shell for the minimum u_z periodic displacement. Conversely, the second one (Fig. 31(b)) does so for the maximum u_z periodic displacement.



(a) Fluid cavity. (b) Flexible shell.

(c) Imposed velocity surfaces. (d) Fixed pressure surface.

Fig. 28. 3D lid-driven cavity with flexible shell. Problem geometry.

Table 7
3D lid-driven cavity with flexible shell. u_z oscillation magnitudes.

	min(u_z) [m]	max(u_z) [m]	A [m]	T [s]
Valdés [83]	0.163	0.251	0.088	5.1
Body fitted	0.192	0.276	0.084	4.8
Embedded	0.189	0.287	0.098	4.75

5.5. 3D four-point tent

5.5.1. Industrial motivation

This last example presents a feasible industrial application of the proposed embedded FSI tool. The problem at hand is the FSI analysis of a four-point tent under the action of an intense wind load.

Since their appearance in the mid past century, there has been an always growing interest on this kind of highly flexible structures. Although they were initially conceived as more appealing alternatives to traditional structural

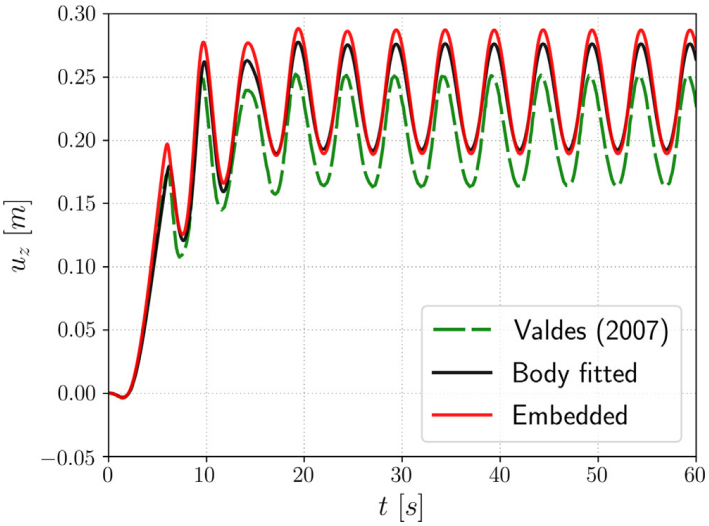
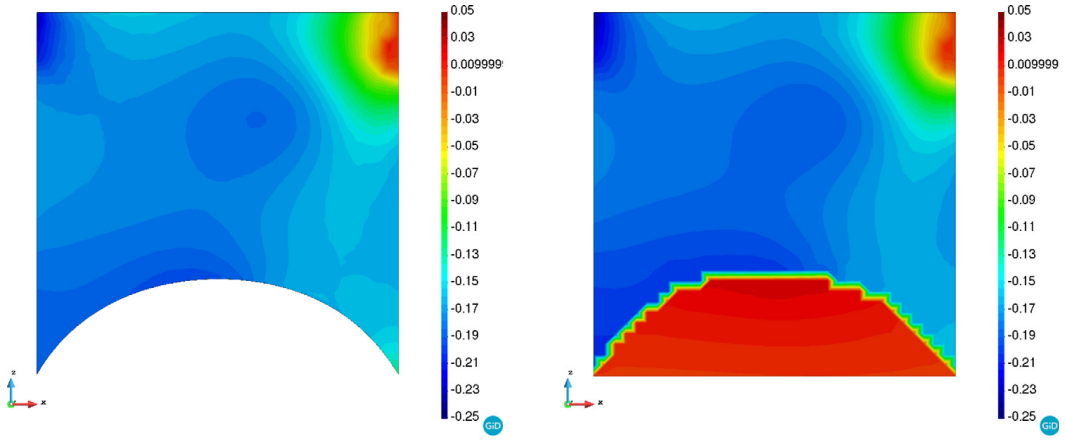
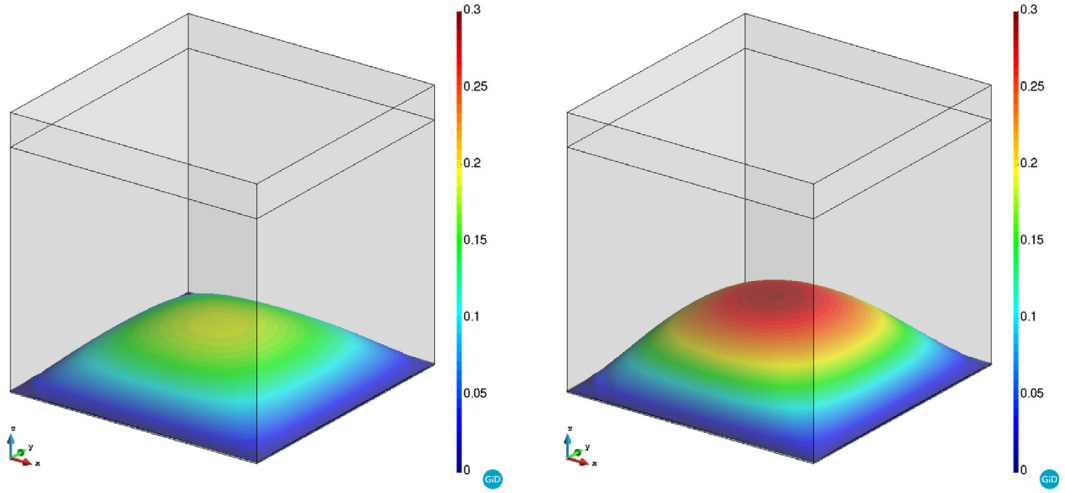


Fig. 29. 3D lid-driven cavity with flexible shell. Midpoint (0.5,0.5)m vertical displacement u_z evolution [m].



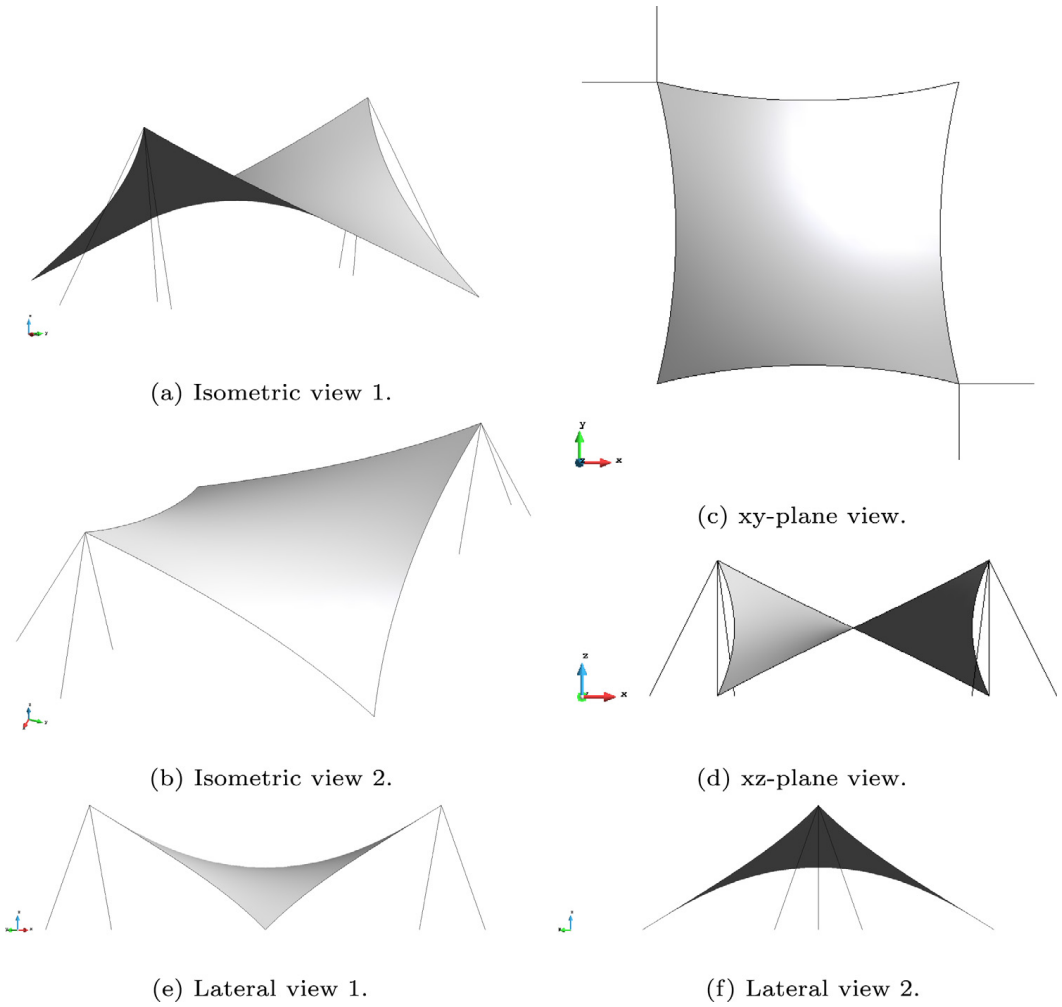
(a) Body fitted. (b) Embedded.

Fig. 30. 3D lid-driven cavity with flexible shell. Pressure field (Pa) xz-plane cross section at maximum vertical displacement u_z.



(a) t = 42.0s (min. u_z). (b) t = 44.4s (max. u_z).

Fig. 31. 3D lid-driven cavity with flexible shell. Displacement norm $\|\mathbf{u}\|$ snapshots during periodic regime [m].



(a) Isometric view 1.

(b) Isometric view 2.

(c) xy-plane view.

(d) xz-plane view.

(e) Lateral view 1.

(f) Lateral view 2.

Fig. 32. 3D four-point tent. Structure geometry.

typologies, their versatility and efficient manufacturing encouraged designers to use them as reference structural solutions in some specific applications [86]. Some examples are the design of extremely lightweight roof covers, such as the Olympiastadion München one by Frei Otto, or the design of sustainable and versatile temporary structures (e.g. inflatable hangars or rapid deployment emergency shelters) [87].

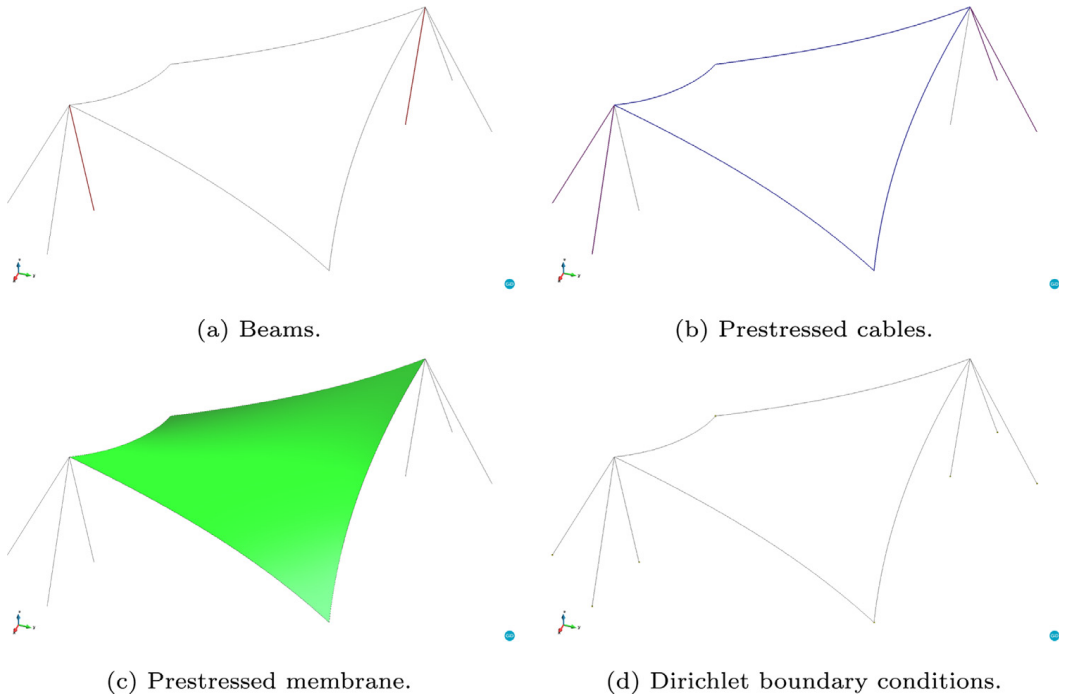
In this context, the use of the embedded approach circumvents all the preprocessing issues that appear in the membrane boundaries during the fluid volume mesh generation. On top of that, these structures are known to suffer from large displacements and rotations, which might even turn into topology changes if self-contact and wrinkling appear. In this regard, the embedded approach is a robust alternative to efficiently handle this situation.

5.5.2. Problem description

The geometry of the problem consists in a four-point tent inside a straight wind channel. Fig. 32(b) collects multiple views of the structure geometry. As it can be observed, the four-point tent is made as a system of beams and prestressed cables supporting a prestressed membrane, whose edges are prestressed cables too (Fig. 33). The membrane is a 1e-3 m thickness hyperbolic paraboloid, whose plan size and elevation are 8x8 m and 4 m.

The wind channel geometry is a 100 $\times$ 40x20 m rectangular prism (Fig. 34), which yields a blockage coefficient of the 4%. The membrane centre is located in the point (4,4,0)m of the channel, which is 24 m far from the inlet yz-plane.

The fluid BCs are rather simple in this case. A constant x-velocity of 50 m/s is imposed in the inlet region (Fig. 34(a)) while the other two components are set to zero. The pressure is fixed to zero in the outlet region



(a) Beams. (b) Prestressed cables.

(c) Prestressed membrane. (d) Dirichlet boundary conditions.

Fig. 33. 3D four-point tent. Element types and boundary conditions. The beams are highlighted in red. The supporting cables are highlighted in purple while the membrane edges ones are highlighted in blue. The prestressed membrane is highlighted in light green. The light brown square dots represent the fixed displacement points. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(Fig. 34(b)). Moreover, we enforce a slip (symmetry) condition in rest of the channel walls (Fig. 34(c)). In the embedded FSI interface, we set the slip length ε to 10^{-3}.

We remark that the slip BC is also applied to the ground wall. Although the slip BC is only well defined for inviscid fluids, it is a good approximation in high Reynolds number (Re) scenarios [88]. Bear in mind that imposing a no-slip condition in the ground would have required the use of an overkill boundary layer mesh to properly capture the wall viscous effects.

Concerning the structural problem BCs, the displacements are constrained in the ground, that is to say in the bottom ends of the supporting beams and cables as well as in two of the membrane corners (Fig. 33(d)).

The fluid and structure material properties are collected in Table 8. For the cables (Fig. 33(b)) we present the density ρ_c, the cross section area A_c, the Young modulus E_c and the prestress σ_c^0. The same information is provided for the supporting beams (Fig. 33(a)) in ρ_b, A_b, E_b and σ_b^0. Equivalently, ρ_m, ν_m, E_m and $\boldsymbol{\sigma}_{\boldsymbol{m}}^{\mathbf{0}}$ stand for the prestressed membrane density, Poisson ratio, Young modulus and x,y and z-directions prestress. ρ_f and μ_f are the fluid density and dynamic viscosity. An elastic material response is used in all the elements that conform the structure while a Newtonian one is used in the fluid.

Both the supporting beams and cables are meshed with a single element. An average element size of 0.15 m is used in the prestressed membrane as well as in its edge cables. These mesh settings yield a 21k elements mesh. For the beams we use standard non-linear Total Lagrangian elements while we use standard non-linear Total Lagrangian truss ones for the cables [81]. The prestressed membrane element formulation is taken from [89].

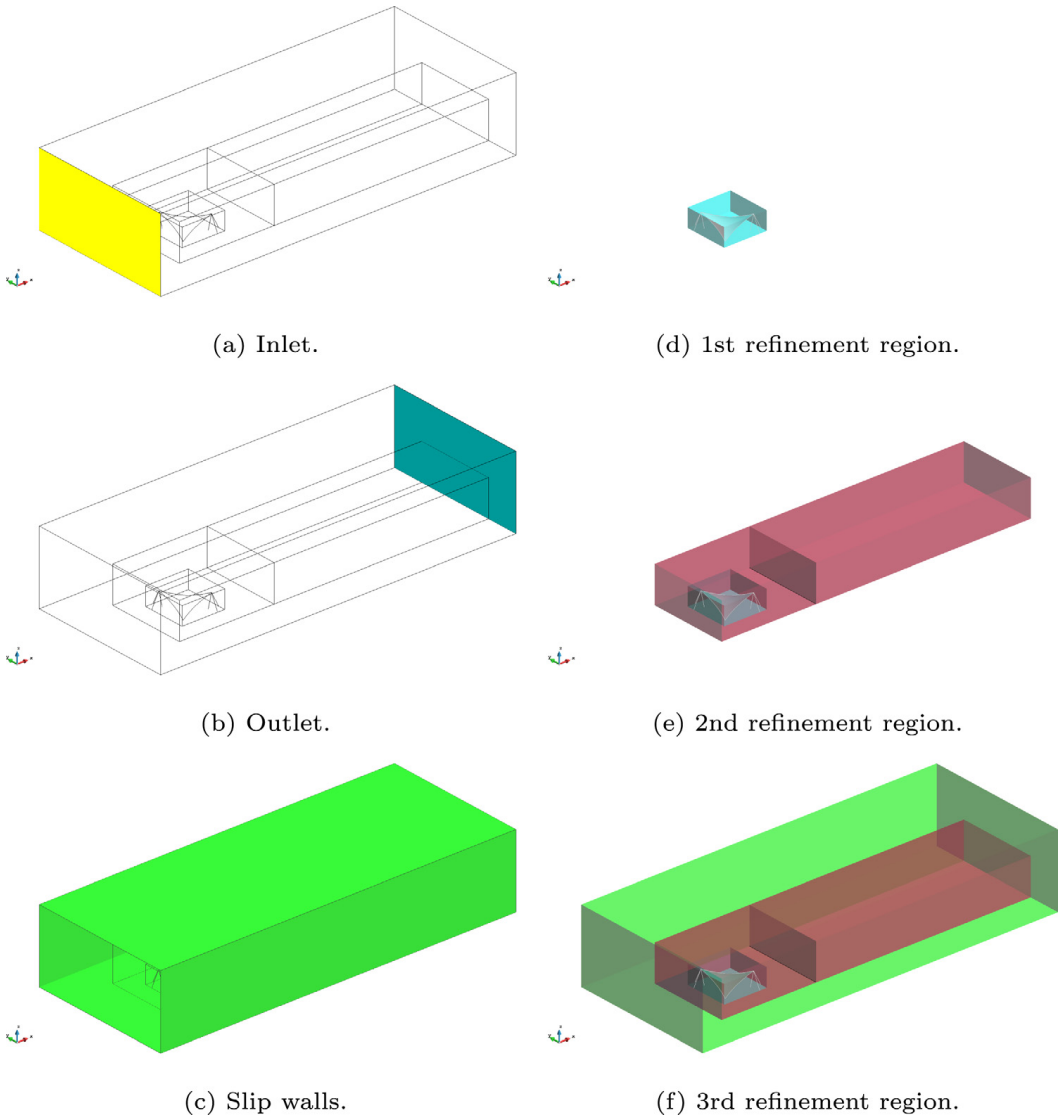
As it can be observed in Fig. 34 the fluid volume mesh is divided in three refinement regions. The 1st region (Fig. 34(d)) is the one surrounding the four-point tent and has an average element size of 0.15 m. The 2nd one (Fig. 34(e)) has 0.5 m (in the vicinities of the 1st region) and 1 m as average element sizes. The 3rd region (the outer one) has an average element size of 2.5 m. Altogether, the three refinement levels amount a total of 3M linear tetrahedra Q1P1 elements.

5.5.3. Results assessment

Owing to the lack of experimental results, what we present in the following is therefore a description and qualitative assessment of the obtained solution.

Table 8
3D four-point tent. Material properties.

Structure						Fluid	
Cables		Beams		Membrane			
ρ_c	8300.0 kg/m^3	ρ_b	8300.0 kg/m^3	ρ_m	800.0 kg/m^3	ρ_f	1.225 kg/m^3
A_c	$1.5e-4$ m^2	A_b	$7.06858e-2$ m^2	ν_m	0.2	μ_f	$1.846e-5$ Pa s
E_c	200.0e9 N/m^2	E_b	200.0e9 N/m^2	E_m	300.0e3 N/m^2		
σ_c^0	1333.3e6 N/m^2	σ_b^0	0.0 N/m^2	$\boldsymbol{\sigma_m^0}$	[5.0e6, 5.0e6, 0] N/m^2		



(a) Inlet. (d) 1st refinement region.

(b) Outlet. (e) 2nd refinement region.

(c) Slip walls. (f) 3rd refinement region.

Fig. 34. 3D four-point tent. Fluid geometry, boundary conditions and refinement regions. The structure geometry is superimposed to the background mesh geometry for the sake of visualization.

First of all, we show in Fig. 35 the time evolution of the three displacement components of a control node whose initial xy-coordinates are (5.25,3.25)m. We select this point because it is located in the region where the structure undergoes the maximum displacements. As it can be observed, all the displacement components show an almost periodic behaviour after an initial transition regime. We also note that the displacement in the wind direction (x-component) is around 0.2 m while the vertical one (z-component) is around 1.0 m.

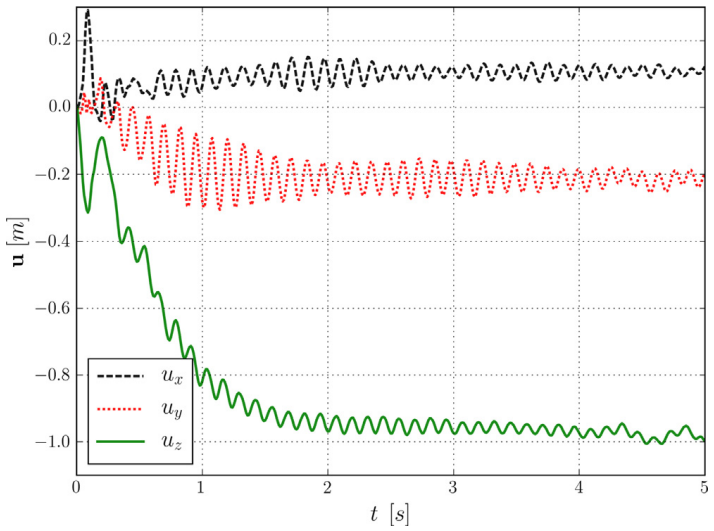


Fig. 35. 3D four-point tent. Point (5.25,3.25) vertical displacement u_z evolution [m].

Furthermore, Fig. 36 presents some snapshots of the deformed geometry together with the corresponding displacement norm field $\|\mathbf{u}\|$. This figure also compares the deformed geometry with the initial configuration (Fig. 36(c)).

Besides that, Fig. 37 collects a set of cross sections of the fluid pressure field p. All these sections are parallel to the xz-plane, which is aligned with the wind direction. Likewise, Fig. 38 depicts the corresponding pressure interpolation on both sides of the membrane structure.

As it is expected, a flow separation occurs in the front region of the tent. This generates an upwards overpressure (Fig. 38) that induces the positive displacements in the vertical direction (Fig. 36(c)). On the contrary, a downwards overpressure appears in the rear part of the tent. Such pressure load in the negative direction is indeed much more relevant than the positive one as the maximum deflections are located in this area of the membrane (Figs. 36(b) and 36(c)).

Although we are confident about the accuracy and reliability of the obtained results, these also reveal some of the possible further enhancements. On the one hand, we notice some issues with the pressure interpolation in the vicinities of the membrane boundaries (Fig. 38). More specifically, these can be easily observed by focusing on the two top membrane edges in Fig. 38(a) and on the right one in Fig. 38(b). These phenomena, which are also reported in [8], are due to the fact that we do not consider any specific treatment of the edge intersected elements (i.e. the intersected elements neighbouring with non-intersected ones). Thus, we can say that this becomes in a "variational crime" that propagates along the contour of the membrane yielding such edge pressure oscillations.

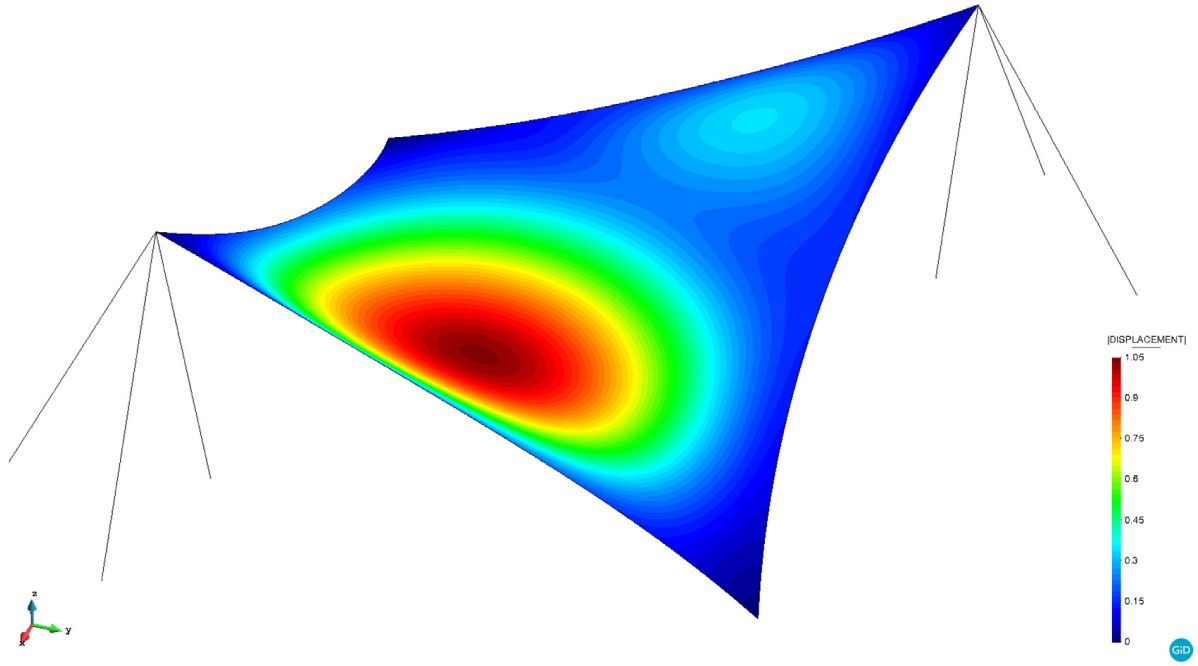
On the other hand, we note that the mesh resolution in the membrane surroundings is not as good as the one that could be achieved by using an equivalent body fitted mesh (Fig. 37). Nonetheless, a reasonably similar resolution can be always achieved upon mesh refining.

Summarizing, the test showcases the capabilities of the method for a realistic application. Aside of the simplifications during the model generation, which allow to bypass the volume meshing of a thin-walled structure, the embedded solver is proved to efficiently solve the problem out of the box. We also note that this example could be a tough case for a traditional ALE mesh motion based solver. Last but not least, we highlight the relevance that these results have for the possible extension of the presented tool to industrial applications.

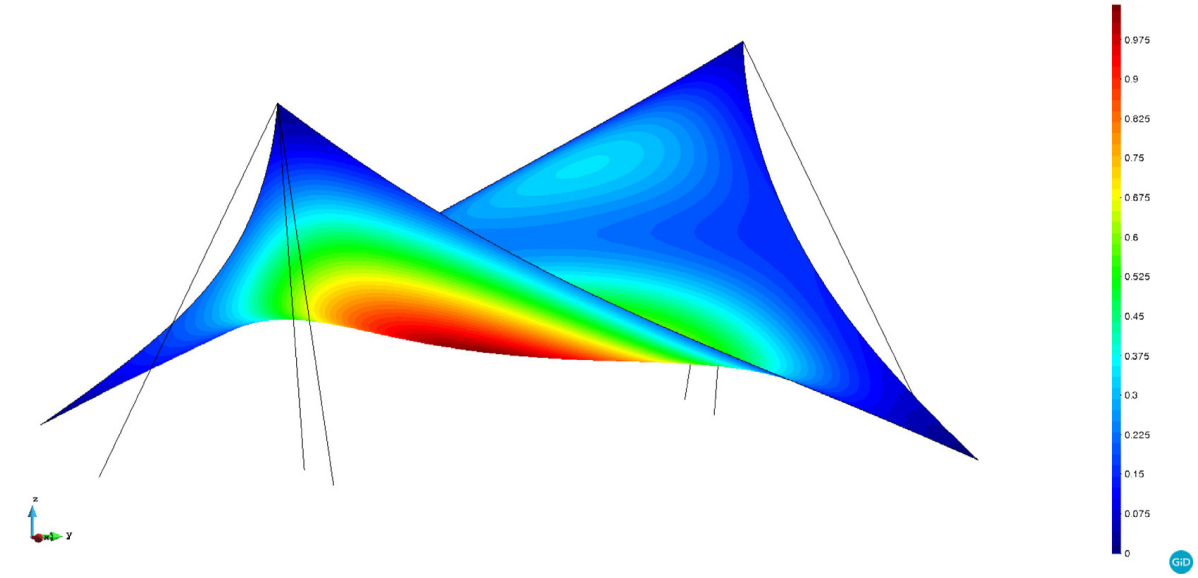
6. Conclusion

In this work we present an embedded framework for the resolution of strongly coupled FSI problems involving structures of any type. On top of the idea that any structure can be classified as volumetric or volumeless, we put special emphasis on creating a single FSI tool to deal with both types of bodies.

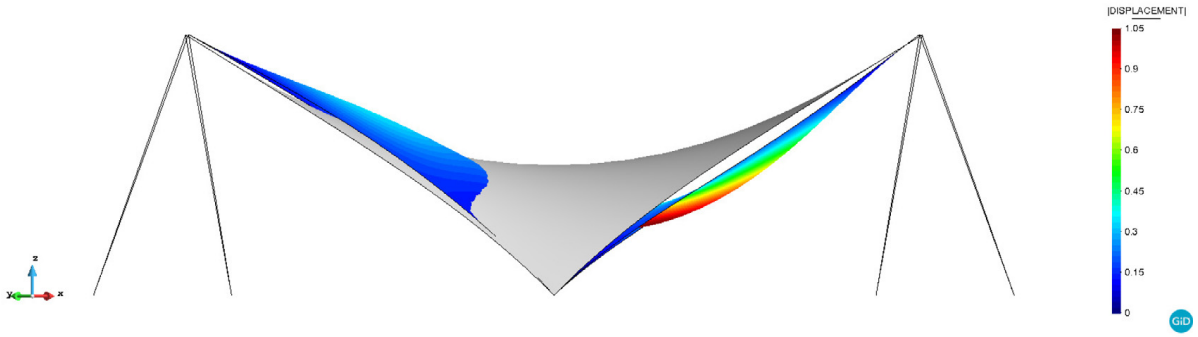
Besides this, the paper also includes some technical advances that could be eventually used in other contexts. These are



(a) Isometric top view.

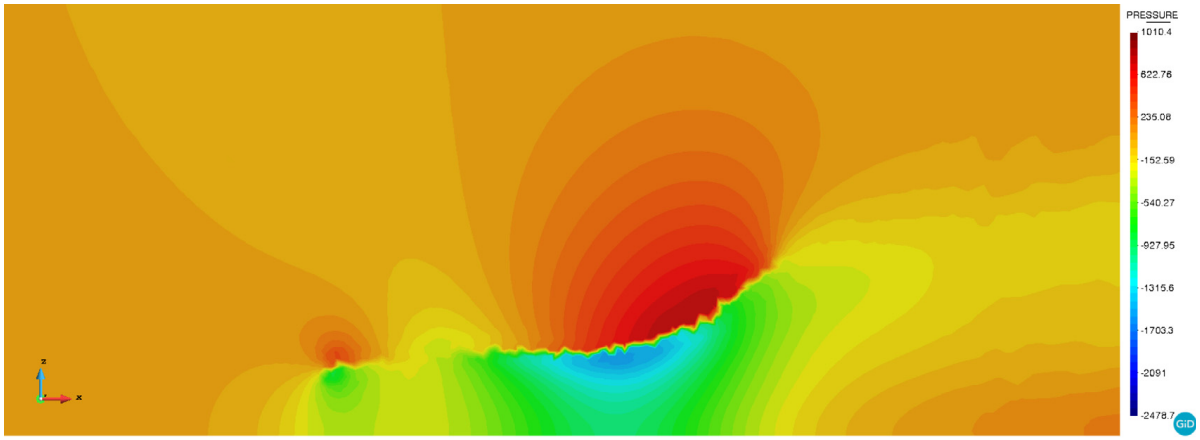


(b) Isometric rear view.

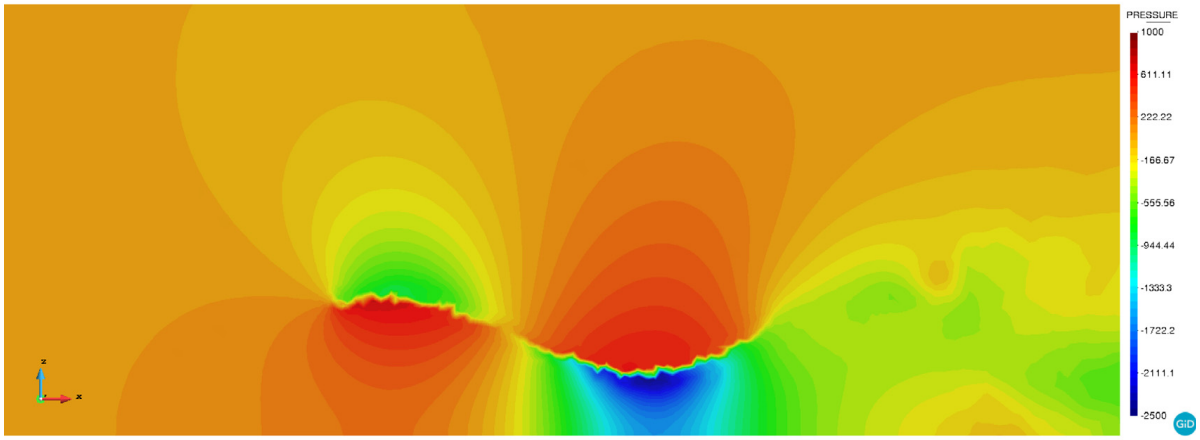


(c) Lateral view with undeformed configuration.

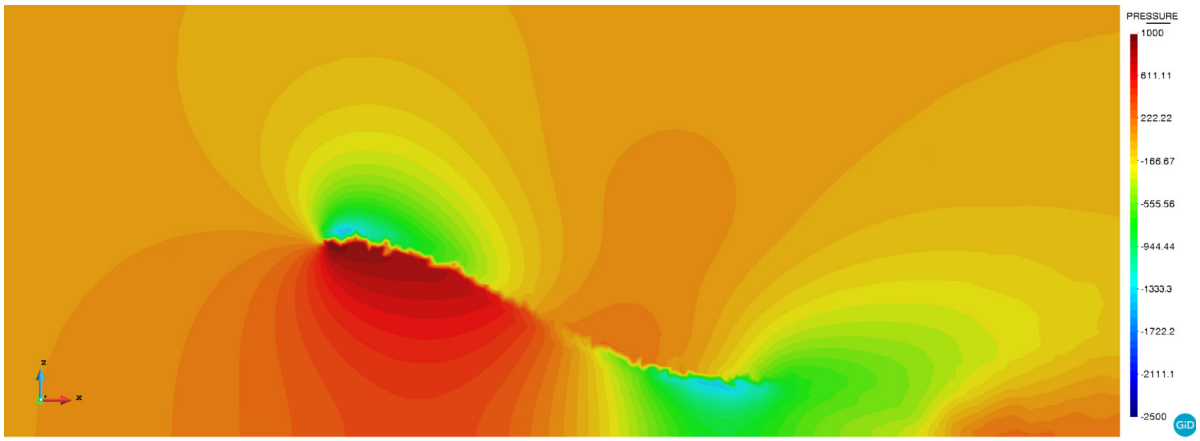
Fig. 36. 3D four-point tent. Structure $\|\mathbf{u}\|$ displacement field at time$t = 5$ s.



(a) y = 2 m.



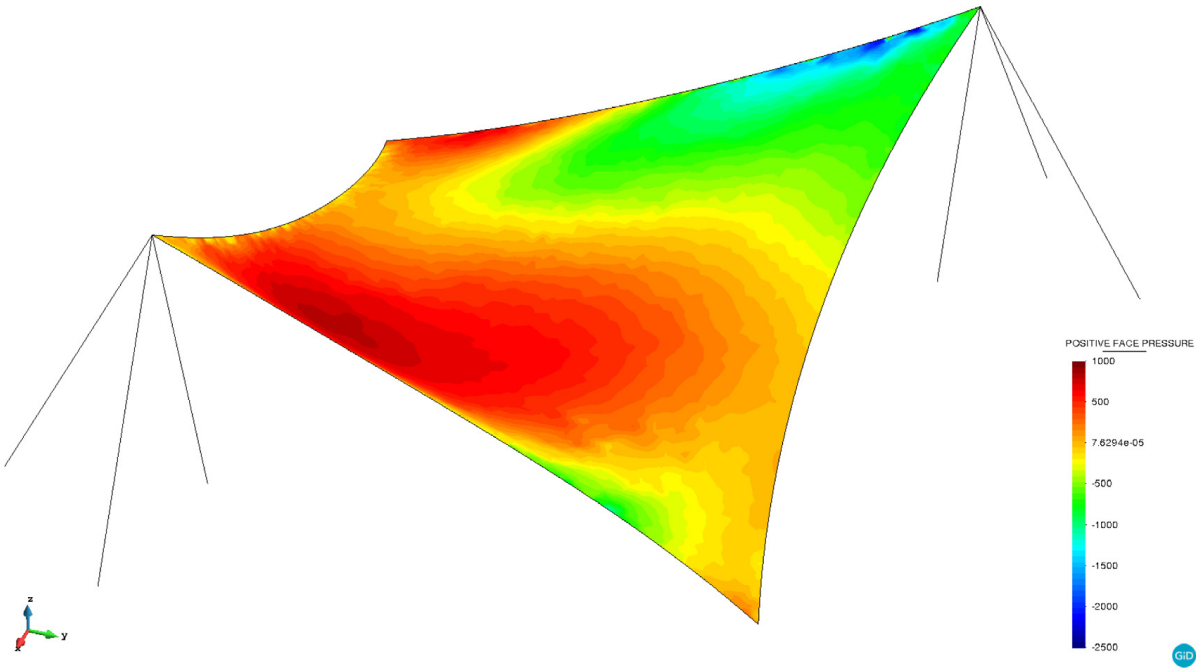
(b) y = 4 m (midplane).



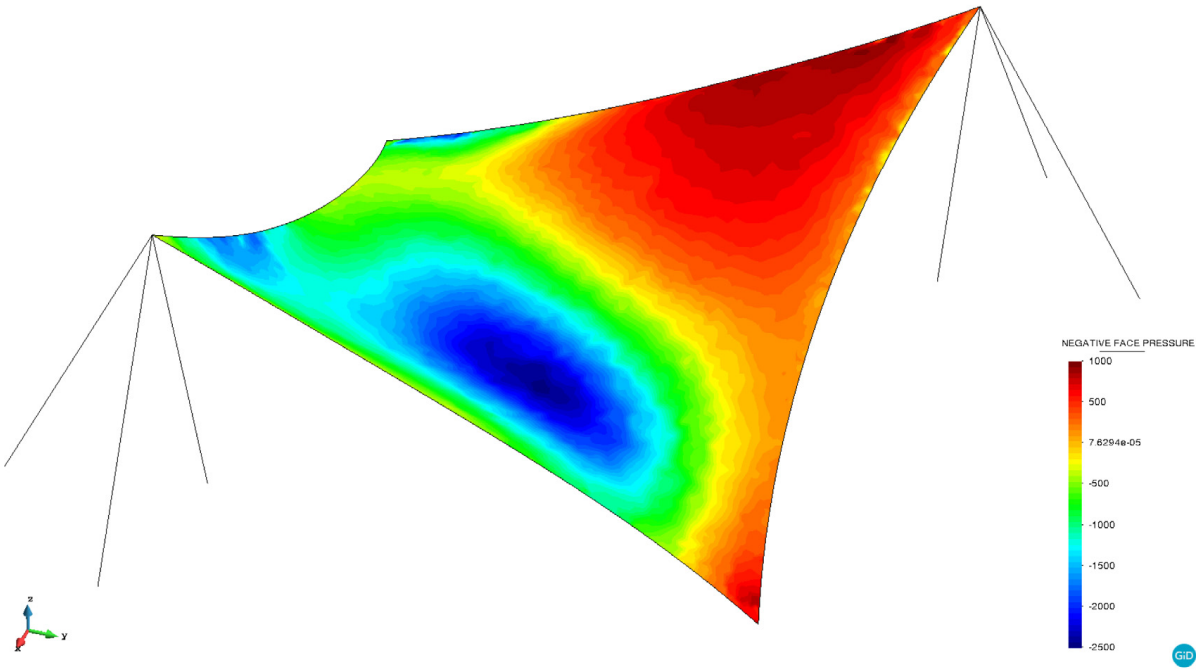
(c) y = 6 m.

Fig. 37. 3D four-point tent. Pressure field xz-plane cross sections at time $t = 5$ s.

- the use of a discontinuous FE embedded approach for the CFD analysis of volumeless structures. This approach is based on the combination of the Ausas discontinuous FE space with a Nitsche imposition of the Navier-slip wall condition.
- the improvement of the FM-ALE algorithm to consider a Lagrangian motion of the embedded interface. Our proposal is to solve a small extrapolation problem in the intersected edges to find the BCs of the FM-ALE



(a) Positive (top) membrane side.



(b) Negative (bottom) membrane side.

Fig. 38. 3D four-point tent. Pressure over the prestressed membrane at time$t = 5$ s.

mesh motion problem. Such extrapolation problem is built so that the interpolation of its results over the embedded skin equals (in a variational sense) the structure mesh displacement.

- the black-box FSI OO implementation. We put special interest on the implementation of our strongly coupled staggered strategy. For the sake of modularity and code reusability we use a black-box coupling approach with a Quasi-Newton interface residual minimization. We highlight the flexibility of the OO implementation

we follow, which makes possible to solve not only FSI problems but also alternative multiphysics ones (e.g. Conjugate Heat Transfer) or to easily integrate it in already existent codes.

We validate our proposal by solving several examples involving both volumetric and membrane-like bodies. The first two examples are the well-known Mok and Turek & Hron benchmarks. The results of the embedded formulation are in really good agreement with the reference ones, as only minor discrepancies, which we believe are associated to the differences in the discretization as well as in the CFD formulation, can be observed. We also highlight the robustness of the embedded approach, which worked out of the box in both cases.

The third example (two-dimensional mixer with flexible blades) is specifically conceived to prove the robustness of the embedded approach to deal with arbitrary and extremely large rotations. Despite the lack of reference results, we observe that the obtained solution is perfectly feasible. We highlight the value of our method to deal with such a challenging scenario, which would have definitively required remeshing if an ALE-based solver was used.

With regard to thin-walled structures, we present two examples involving such structural typologies. The first one is a lid driven cavity problem with a flexible shell structure in the bottom. Although we observe that the embedded deflections are slightly larger than the reference ones, the period and amplitude of the oscillation are however almost identical to the reference ones.

The second one is the FSI analysis of a 4-point tent during a extreme wind load episode. We note that, aside of further assessing the performance of the solver when volumeless bodies are considered, this problem represents a feasible real-life application of the proposed technology. Despite the lack of reference results, we can say that the solver accurately solves the problem and is capable of capturing the flow features that induce the large deflection of the membrane. These results make us be confident about the application of the proposed tool to real engineering problems.

Once arrived to this point, we can say that our embedded FSI framework arises as a possible alternative to solve the problems at hand. We would like to stress the achievements concerning thin-walled structures. It is known that these structures are prone to suffer from extremely large displacements that likely turn into topology changes when self-contact and wrinkling appear. In view of the obtained results and according to the nature of the method, we think that our proposal has the potential of being an alternative to efficiently handle such challenging scenario.

As a summary, we collect in the following lines the main features of our proposal. The first one is that it does not need to reconstruct the system matrix graph each time the level set is updated. Besides that, it is purely local for both volume and volumeless bodies. Hence, any computational overhead coming from the neighbours calculation is completely avoided. Another feature that distinguishes our approach from other non-conforming alternatives is that we do the FSI interface BC imposition (both for volume and volumeless bodies) over the zero isosurface of the level set function, which is the best implicit representation of the real interface. Complementary, we would like to mention that, up to our knowledge, there is no other work in the literature that presents a strongly coupled embedded FSI framework conceived in an object oriented black-box fashion.

Nonetheless, we also note some disadvantages after comparing our technique to the standard body fitted approach. The main drawback is of course the lack of resolution in the vicinities of the immersed bodies. With this regard, we observe that ALE body fitted meshes normally yield a more accurate solution in the FSI interface than the one obtained with a similar embedded discretization. As this is a direct consequence of the nature of the method, which relies on solving the problem using an approximation of the discrete geometry, details can only be recovered upon mesh refinement. In short, we notice that for a similar resolution level, embedded methods normally require a finer mesh in the structure surroundings, which turns into a computational overhead.

We conclude the paper by briefly describing the further investigations arising from this work. The first task that is left to be done is the distributed memory implementation of the presented tool. Although both the embedded CFD and the CSM solvers are already compatible with MPI parallelism, we still lack the FSI coupling implementation in order to efficiently solve real-life problems. Alternatively, we also plan to explore the possibilities of adaptive mesh refinement to tackle the reported mesh resolution issues. Last but not least, we also want to further work on the discontinuous level set representation of volumeless bodies. As we report in the 4-point tent example, it still has some issues in the embedded skin boundaries. We believe that these could be efficiently located and handled by using a discontinuous edge-based level set function instead of the current elemental one.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research has been partly supported by the International Graduate School of Science and Engineering, Germany (IGSSE) and the European Commission (EC) through the projects ATMOPACE and ExaQUte (H2020-FETHPC-2016-2017-800898). The authors also acknowledge financial support from the Spanish Ministry of Economy and Competitiveness , through the "Severo Ochoa Programme for Centres of Excellence in R&D" (CEX2018-000797-S). Rubén Zorrilla gratefully acknowledges the support of the Spanish ministry of Science, Innovation and Universities for his FPU grant (FPU15/03796).

References

[1] R. Scotta, M. Lazzari, E. Stecca, J. Cotela, R. Rossi, Numerical wind tunnel for aerodynamic and aeroelastic characterization of bridge deck sections, Comput. Struct. 167 (2016) 96–114, http://dx.doi.org/10.1016/j.compstruc.2016.01.012, URL http://www.sciencedirect.com/science/article/pii/S0045794916300098.

[2] M. Saeedi, R. Wüchner, K.-U. Bletzinger, Fluid-structure interaction analysis and performance evaluation of a membrane blade, J. Phys. Conf. Ser. 753 (2016) 102009, http://dx.doi.org/10.1088/1742-6596/753/10/102009.

[3] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. 227 (10) (2008) 4825–4852, http://dx.doi.org/10.1016/j.jcp.2008.01.028.

[4] H. Zhao, J.B. Freund, R.D. Moser, A fixed-mesh method for incompressible flow–structure systems with finite solid deformations, J. Comput. Phys. 227 (6) (2008) 3114–3140, http://dx.doi.org/10.1016/j.jcp.2007.11.019.

[5] C.A. Figueroa, I.E. Vignon-Clementel, K.E. Jansen, T.J. Hughes, C.A. Taylor, A coupled momentum method for modeling blood flow in three-dimensional deformable arteries, in: John H. Argyris Memorial Issue. Part II, Comput. Methods Appl. Mech. Engrg. 195 (41) (2006) 5685–5706, http://dx.doi.org/10.1016/j.cma.2005.11.011, URL http://www.sciencedirect.com/science/article/pii/S004578250500513X.

[6] P. Ryzhakov, E. Soudah, N. Dialami, Computational modeling of the fluid flow and the flexible intimal flap in type B aortic dissection via a monolithic Arbitrary Lagrangian/Eulerian fluid-structure interaction model, Int. J. Numer. Methods Biomed. Eng. n/a (n/a) (2019) e3239, http://dx.doi.org/10.1002/cnm.3239.

[7] L. Taelman, J. Bols, J. Degroote, V. Muthurangu, J. Panzer, J. Vierendeels, P. Segers, Differential impact of local stiffening and narrowing on hemodynamics in repaired aortic coarctation: an FSI study, Med. Biol. Eng. Comput. 54 (2) (2016) 497–510, http://dx.doi.org/10.1007/s11517-015-1336-1.

[8] R. Zorrilla, E. Soudah, R. Rossi, Computational modeling of the fluid flow in type b aortic dissection using a modified finite element embedded formulation, Biomech. Model. Mechanobiol. (2020) http://dx.doi.org/10.1007/s10237-020-01291-x.

[9] S. Dahl, J. Vierendeels, J. Degroote, S. Annerel, L. Hellevik, B. Skallerud, FSI simulation of asymmetric mitral valve dynamics during diastolic filling, Comput. Methods Biomech. Biomed. Eng. 15 (2) (2012) 121–130, http://dx.doi.org/10.1080/10255842.2010.517200.

[10] D. Kamensky, M.-C. Hsu, D. Schillinger, J. Evans, A. Aggarwal, Y. Bazilevs, M. Sacks, T. Hughes, An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves, Comput. Methods Appl. Mech. Engrg. 284 (2015) 1005–1053, http://dx.doi.org/10.1016/j.cma.2014.10.040.

[11] M. de Tullio, G. Pascazio, A moving-least-squares immersed boundary method for simulating the fluid–structure interaction of elastic bodies with arbitrary thickness, J. Comput. Phys. 325 (2016) 201–225, http://dx.doi.org/10.1016/j.jcp.2016.08.020, URL http://www.sciencedirect.com/science/article/pii/S0021999116303692.

[12] H. Casquero, Y.J. Zhang, C. Bona-Casas, L. Dalcin, H. Gomez, Non-body-fitted fluid–structure interaction: Divergence-conforming b-splines, fully-implicit dynamics, and variational formulation, J. Comput. Phys. 374 (2018) 625–653, http://dx.doi.org/10.1016/j.jcp.2018.07.020, URL http://www.sciencedirect.com/science/article/pii/S0021999118304819.

[13] C. Hirt, A. Amsden, J. Cook, An arbitrary Lagrangian-Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253, http://dx.doi.org/10.1016/0021-9991(74)90051-5.

[14] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian–Eulerian methods, in: Encyclopedia of Computational Mechanics, American Cancer Society, 2004, http://dx.doi.org/10.1002/0470091355.ecm009.

[15] C. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (3) (1977) 220–252, http://dx.doi.org/10.1016/0021-9991(77)90100-0.

[16] D.A. Field, Laplacian smoothing and Delaunay triangulations, Commun. Appl. Numer. Methods 4 (6) (1988) 709–712, http://dx.doi.org/10.1002/cnm.1630040603.

[17] K. Stein, T. Tezduyar, R. Benney, Mesh moving techniques for fluid-structure interactions with large displacements, J. Appl. Mech. 70 (1) (2003) 58–63, http://dx.doi.org/10.1115/1.1530635.

[18] E.O. nate, S.R. Idelsohn, M.A. Celigueta, R. Rossi, Advances in the particle finite element method for the analysis of fluid–multibody interaction and bed erosion in free surface flows, in: Computational Methods in Fluid–Structure Interaction, Comput. Methods Appl. Mech. Engrg. 197 (19) (2008) 1777–1800, http://dx.doi.org/10.1016/j.cma.2007.06.005, URL http://www.sciencedirect.com/science/article/pii/S0045782507002368.

[19] P.B. Ryzhakov, R. Rossi, S.R. Idelsohn, E. Oñate, A monolithic Lagrangian approach for fluid–structure interaction problems, Comput. Mech. 46 (6) (2010) 883–899, http://dx.doi.org/10.1007/s00466-010-0522-0.

[20] A. Franci, E.O. nate, J.M. Carbonell, Unified Lagrangian formulation for solid and fluid mechanics and FSI problems, Comput. Methods Appl. Mech. Engrg. 298 (2016) 520–547, http://dx.doi.org/10.1016/j.cma.2015.09.023, URL http://www.sciencedirect.com/science/article/pii/S0045782515003175.

[21] C. Peskin, The immersed boundary method, Acta Numer. 112 (2002) 479–517, http://dx.doi.org/10.1017/S0962492902000077.
[22] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, CutFEM: Discretizing geometry and partial differential equations, Internat. J. Numer. Methods Engrg. 104 (7) (2015) 472–501, http://dx.doi.org/10.1002/nme.4823.
[23] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems, J. Comput. Phys. 372 (2018) 972–995, http://dx.doi.org/10.1016/j.jcp.2017.10.026.
[24] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. Part II: Linear advection–diffusion and incompressible Navier–Stokes equations, J. Comput. Phys. 372 (2018) 996–1026, http://dx.doi.org/10.1016/j.jcp.2018.01.023.
[25] C. Peskin, Flow patterns around heart valves: A numerical method, J. Comput. Phys. 10 (2) (1972) 252–271, http://dx.doi.org/10.1016/0021-9991(72)90065-4.
[26] C. Peskin, D. McQueen, A three-dimensional computational method for blood flow in the heart I. Immersed elastic fibers in a viscous incompressible fluid, J. Comput. Phys. 81 (2) (1989) 372–405, http://dx.doi.org/10.1016/0021-9991(89)90213-1.
[27] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, first ed., in: Applied Mathematical Siences, vol. 153, Springer-Verlag New York, 2003.
[28] R. Zorrilla, A. Larese, R. Rossi, A modified finite element formulation for the imposition of the slip boundary condition over embedded volumeless geometries, Comput. Methods Appl. Mech. Engrg. 353 (2019) 123–157, http://dx.doi.org/10.1016/j.cma.2019.05.007, URL http://www.sciencedirect.com/science/article/pii/S004578251930266X.
[29] D. Baumgärtner, J. Wolf, R. Rossi, P. Dadvand, R. Wüchner, A robust algorithm for implicit description of immersed geometries within a background mesh, Adv. Model. Simul. Eng. Sci. 5 (1) (2018) 21, http://dx.doi.org/10.1186/s40323-018-0113-8.
[30] J. Wolf, D. Baumgärtner, R. Rossi, P. Dadvand, R. Wüchner, Contribution to the Fluid-Structure Interaction Analysis of Ultra-Lightweight Structures using an Embedded Approach, International Center for Numerical Methods in Engineering, 2015, http://dx.doi.org/10.13140/RG.2.1.1079.8561.
[31] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (1) (2001) 132–150, http://dx.doi.org/10.1006/jcph.2001.6778, URL http://www.sciencedirect.com/science/article/pii/S0021999101967786.
[32] L. Zhang, A. Gerstenberger, X. Wang, W.K. Liu, Immersed finite element method, in: Flow Simulation and Modeling, Comput. Methods Appl. Mech. Engrg. 193 (21) (2004) 2051–2067, http://dx.doi.org/10.1016/j.cma.2003.12.044.
[33] Z.-Q. Zhang, G.R. Liu, B.C. Khoo, A three dimensional immersed smoothed finite element method (3D IS-FEM) for fluid–structure interaction problems, Comput. Mech. 51 (2) (2013) 129–150, http://dx.doi.org/10.1007/s00466-012-0710-1.
[34] R. Löhner, J.D. Baum, E. Mestreau, D. Sharov, C. Charman, D. Pelessone, Adaptive embedded unstructured grid methods, Internat. J. Numer. Methods Engrg. 60 (3) (2004) 641–660, http://dx.doi.org/10.1002/nme.978.
[35] L. Foucard, F. Vernerey, An X-FEM-based numerical–asymptotic expansion for simulating a Stokes flow near a sharp corner, Internat. J. Numer. Methods Engrg. 102 (2) (2015) 79–98, http://dx.doi.org/10.1002/nme.4746.
[36] T. Sawada, A. Tezuka, LLM and x-FEM based interface modeling of fluid–thin structure interactions on a non-interface-fitted mesh, Comput. Mech. 48 (3) (2011) 319–332, http://dx.doi.org/10.1007/s00466-011-0600-y.
[37] F. Alauzet, B. Fabrèges, M.A. Fernández, M. Landajuela, Nitsche-XFEM for the coupling of an incompressible fluid with immersed thin-walled structures, Comput. Methods Appl. Mech. Engrg. 301 (2016) 300–335, http://dx.doi.org/10.1016/j.cma.2015.12.015.
[38] J. Nitsche, Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind, Abh. Math. Semin. Univ. Hambg. 36 (1) (1971) 9–15, http://dx.doi.org/10.1007/BF02995904.
[39] J. Baiges, R. Codina, F. Henke, S. Shahmiri, W.A. Wall, A symmetric method for weakly imposing Dirichlet boundary conditions in embedded finite element meshes, Internat. J. Numer. Methods Engrg. 90 (5) (2012) 636–658, http://dx.doi.org/10.1002/nme.3339.
[40] R. Codina, J. Baiges, Approximate imposition of boundary conditions in immersed boundary methods, Internat. J. Numer. Methods Engrg. 80 (2009) 1379–1405, http://dx.doi.org/10.1002/nme.2662.
[41] A. Massing, M. Larson, A. Logg, M. Rognes, A stabilized nitsche fictitious domain method for the Stokes problem, J. Sci. Comput. 61 (3) (2014) 604–628, http://dx.doi.org/10.1007/s10915-014-9838-9.
[42] A. Massing, B. Schott, W. Wall, A stabilized Nitsche cut finite element method for the Oseen problem, Comput. Methods Appl. Mech. Engrg. 328 (2018) 262–300, http://dx.doi.org/10.1016/j.cma.2017.09.003.
[43] J. Urquiza, A. Garon, M.-I. Farinas, Weak imposition of the slip boundary condition on curved boundaries for Stokes flow, J. Comput. Phys. 256 (2014) 748–767, http://dx.doi.org/10.1016/j.jcp.2013.08.045.
[44] M. Winter, B. Schott, A. Massing, W. Wall, A Nitsche cut finite element method for the Oseen problem with general Navier boundary conditions, Comput. Methods Appl. Mech. Engrg. 330 (2018) 220–252, http://dx.doi.org/10.1016/j.cma.2017.10.023.
[45] R. Ausas, F. Sousa, G. Buscaglia, An improved finite element space for discontinuous pressures, Comput. Methods Appl. Mech. Engrg. 199 (2010) 1019–1031, http://dx.doi.org/10.1016/j.cma.2009.11.011.
[46] R. Zorrilla, A. Larese, R. Rossi, A discontinuous nitsche-based finite element formulation for the imposition of the general Navier-slip condition over embedded volumeless geometries, Comput. Methods Appl. Mech. Engrg. (2020) submitted for publication.
[47] S. Badia, F. Verdugo, A.F. Martín, The aggregated unfitted finite element method for elliptic problems, Comput. Methods Appl. Mech. Engrg. 336 (2018) 533–553, http://dx.doi.org/10.1016/j.cma.2018.03.022.
[48] R. Codina, G. Houzeaux, H. Coppola-Owen, J. Baiges, The fixed-mesh ALE approach for the numerical approximation of flows in moving domains, J. Comput. Phys. 228 (5) (2009) 1591–1611, http://dx.doi.org/10.1016/j.jcp.2008.11.004.
[49] J. Baiges, R. Codina, The fixed-mesh ALE approach applied to solid mechanics and fluid–structure interaction problems, Internat. J. Numer. Methods Engrg. 81 (12) (2010) 1529–1557, http://dx.doi.org/10.1002/nme.2740.
[50] J. Baiges, R. Codina, H. Coppola-Owen, The fixed-mesh ALE approach for the numerical simulation of floating solids, Internat. J. Numer. Methods Fluids 67 (8) (2011) 1004–1023, http://dx.doi.org/10.1002/fld.2403.

[51] R. Rossi, E.O. nate, Analysis of some partitioned algorithms for fluid-structure interaction, Eng. Comput. 27 (1) (2010) 20–56, http://dx.doi.org/10.1108/02644401011008513.
[52] S. Badia, F. Nobile, C. Vergara, Fluid–structure partitioned procedures based on Robin transmission conditions, J. Comput. Phys. 227 (14) (2008) 7027–7051, http://dx.doi.org/10.1016/j.jcp.2008.04.006.
[53] L. Gerardo-Giorda, F. Nobile, C. Vergara, Analysis and optimization of Robin–Robin partitioned procedures in fluid–structure interaction problems, SIAM J. Numer. Anal. 48 (6) (2010) 2091–2116, http://dx.doi.org/10.1137/09076605X.
[54] M.A. Fernández, M. Landajuela, M. Vidrascu, Fully decoupled time-marching schemes for incompressible fluid/thin-walled structure interaction, J. Comput. Phys. 297 (2015) 156–181, http://dx.doi.org/10.1016/j.jcp.2015.05.009.
[55] M. Landajuela, M. Vidrascu, D. Chapelle, M.A. Fernández, Coupling schemes for the FSI forward prediction challenge: Comparative study and validation, Int. J. Numer. Methods Biomed. Eng. 33 (4) (2017) e2813, http://dx.doi.org/10.1002/cnm.2813.
[56] R. Rossi, P.B. Ryzhakov, E.O. nate, A monolithic FE formulation for the analysis of membranes in fluids, Int. J. Space Struct. 24 (4) (2009) 205–210, http://dx.doi.org/10.1260/026635109789968263.
[57] C. Farhat, K.G. van der Zee, P. Geuzaine, Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity, in: Fluid-Structure Interaction, Comput. Methods Appl. Mech. Engrg. 195 (17) (2006) 1973–2001, http://dx.doi.org/10.1016/j.cma.2004.11.031.
[58] P. Causin, J. Gerbeau, F. Nobile, Added-mass effect in the design of partitioned algorithms for fluid–structure problems, Comput. Methods Appl. Mech. Engrg. 194 (42) (2005) 4506–4527, http://dx.doi.org/10.1016/j.cma.2004.12.005.
[59] S.R. Idelsohn, F. Del Pin, R. Rossi, E. Oate, Fluid–structure interaction problems with strong added-mass effect, Internat. J. Numer. Methods Engrg. 80 (10) (2009) 1261–1294, http://dx.doi.org/10.1002/nme.2659.
[60] J. Degroote, K.-J. Bathe, J. Vierendeels, Performance of a new partitioned procedure versus a monolithic procedure in fluid–structure interaction, Comput. Struct. 87 (11) (2009) 793–801, http://dx.doi.org/10.1016/j.compstruc.2008.11.013.
[61] J. Degroote, J. Vierendeels, Multi-solver algorithms for the partitioned simulation of fluid–structure interaction, Comput. Methods Appl. Mech. Engrg. 200 (25) (2011) 2195–2210, http://dx.doi.org/10.1016/j.cma.2011.03.015.
[62] A. Bogaers, S. Kok, B. Reddy, T. Franz, Quasi-Newton methods for implicit black-box FSI coupling, Comput. Methods Appl. Mech. Engrg. 279 (2014) 113–132, http://dx.doi.org/10.1016/j.cma.2014.06.033.
[63] S. Sicklinger, V. Belsky, B. Engelmann, H. Elmqvist, H. Olsson, R. Wüchner, K.-U. Bletzinger, Interface Jacobian-based co-simulation, Internat. J. Numer. Methods Engrg. 98 (6) (2014) 418–444, http://dx.doi.org/10.1002/nme.4637.
[64] U. Küttler, W.A. Wall, Fixed-point fluid–structure interaction solvers with dynamic relaxation, Comput. Mech. 43 (1) (2008) 61–72, http://dx.doi.org/10.1007/s00466-008-0255-5.
[65] D. Mok, W. Wall, E. Ramm, Accelerated iterative substructuring schemes for instationary fluid-structure interaction, Comput. Fluid Solid Mech. 2 (2001) 1325–1328.
[66] D. Knoll, D. Keyes, Jacobian-free Newton–Krylov methods: a survey of approaches and applications, J. Comput. Phys. 193 (2) (2004) 357–397, http://dx.doi.org/10.1016/j.jcp.2003.08.010.
[67] U. Küttler, M. Gee, C. Förster, A. Comerford, W.A. Wall, Coupling strategies for biomedical fluid–structure interaction problems, Int. J. Numer. Methods Biomed. Eng. 26 (3–4) (2010) 305–321, http://dx.doi.org/10.1002/cnm.1281.
[68] S. Minami, S. Yoshimura, Performance evaluation of nonlinear algorithms with line-search for partitioned coupling techniques for fluid–structure interactions, Internat. J. Numer. Methods Fluids 64 (10–12) (2010) 1129–1147, http://dx.doi.org/10.1002/fld.2274.
[69] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, J. Vierendeels, Performance of partitioned procedures in fluid–structure interaction, Comput. Struct. 88 (7) (2010) 446–457, http://dx.doi.org/10.1016/j.compstruc.2009.12.006.
[70] T. Spenke, N. Hosters, M. Behr, A multi-vector interface quasi-Newton method with linear complexity for partitioned fluid–structure interaction, Comput. Methods Appl. Mech. Engrg. 361 (2020) 112810, http://dx.doi.org/10.1016/j.cma.2019.112810, URL http://www.sciencedirect.com/science/article/pii/S0045782519307029.
[71] P. Dadvand, R. Rossi, E.O. nate, An object-oriented environment for developing finite element codes for multi-disciplinary applications, Arch. Comput. Methods Eng. 17 (3) (2010) 253–297, http://dx.doi.org/10.1007/s11831-010-9045-2.
[72] P. Dadvand, R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S. Idelsohn, E.O. nate, Migration of a generic multi-physics framework to HPC environments, Comput. & Fluids 80 (2013) 301–309, http://dx.doi.org/10.1016/j.compfluid.2012.02.004.
[73] A. Coll, R. Ribó, M. Pasenau, E. Escolano, J. Perez, A. Melendo, A. Monros, J. Gárate, GiD v.13 User Manual, CIMNE, 2016.
[74] A. Coll, R. Ribó, M. Pasenau, E. Escolano, J. Perez, A. Melendo, A. Monros, J. Gárate, GiD v.13 Reference Manual, CIMNE, 2016.
[75] T. Hughes, L. Franca, M. Balestra, A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: a stable Petrov-Galerkin formulation of the Stokes problem accommodating equal-order interpolations, Comput. Methods Appl. Mech. Engrg. 59 (1) (1986) 85–99, http://dx.doi.org/10.1016/0045-7825(86)90025-3.
[76] J. Cotela, R. Rossi, E. Oñate, A FIC-based stabilized finite element formulation for turbulent flows, Comput. Methods Appl. Mech. Engrg. 315 (2017) 607–631, http://dx.doi.org/10.1016/j.cma.2016.11.020.
[77] T. Tezduyar, S. Mittal, S. Ray, R. Shih, Incompressible flow computations with stabilized bilinear and linear equal-order-interpolation velocity-pressure elements, Comput. Methods Appl. Mech. Engrg. 95 (2) (1992) 221–242, http://dx.doi.org/10.1016/0045-7825(92)90141-6.
[78] T. Hughes, Multiscale phenomena: Green's function, the dirichlet to Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations, Comput. Methods Appl. Mech. Engrg. 127 (1) (1995) 387–401, http://dx.doi.org/10.1016/0045-7825(95)00844-9.
[79] T. Hughes, G. Feijóo, L. Mazzei, J. Quincy, The variational multiscale method—a paradigm for computational mechanics, in: Advances in Stabilized Methods in Computational Mechanics, Comput. Methods Appl. Mech. Engrg. 166 (1) (1998) 3–24, http://dx.doi.org/10.1016/S0045-7825(98)00079-6.

[80] R. Codina, A stabilized finite element method for generalized stationary incompressible flows, Comput. Methods Appl. Mech. Engrg. 190 (20) (2001) 2681–2706, http://dx.doi.org/10.1016/S0045-7825(00)00260-7.
[81] T. Belytschko, W.K. Liu, B. Moran, K. Elkhodary, Nonlinear Finite Elements for Continua and Structures, John Wiley & Sons, 2013.
[82] P. Khosravi, R. Ganesan, R. Sedaghati, Corotational non-linear analysis of thin plates and shells using a new shell element, Internat. J. Numer. Methods Engrg. 69 (4) (2007) 859–885, http://dx.doi.org/10.1002/nme.1791.
[83] G. Valdés, Nonlinear Analysis of Orthotropic Membrane and Shell Structures Including Fluid-Structure Interaction (Ph.D. thesis), Universitat Politècnica de Catalunya, 2007.
[84] S. Turek, J. Hron, Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow, in: H. Bungartz, M. Schäfer (Eds.), Fluid-Structure Interaction, Springer Berlin Heidelberg, Berlin, Heidelberg, 2006, pp. 371–385, http://dx.doi.org/10.1007/3-540-34596-5_15.
[85] C.A. Felippa, A study of optimal membrane triangles with drilling freedoms, Comput. Methods Appl. Mech. Engrg. 192 (16) (2003) 2125–2168, http://dx.doi.org/10.1016/S0045-7825(03)00253-6.
[86] F. Otto, R. Trostel, F.K. Schleyer, Tensile Structures: Design, Structure, and Calculation of Buildings of Cables, Nets, and Membranes, vol. 69, MIT Press London, 1967.
[87] E. Oñate, F.G. Flores, J. Marcipar, Membrane structures formed by low pressure inflatable tubes. New analysis methods and recent constructions, in: E. Oñate, B. Kröplin (Eds.), Textile Composites and Inflatable Structures II, Springer Netherlands, Dordrecht, 2008, pp. 163–196, http://dx.doi.org/10.1007/978-1-4020-6856-0_10.
[88] J. Hoffman, J. Jansson, C. Johnson, New theory of flight, J. Math. Fluid Mech. 18 (2015) http://dx.doi.org/10.1007/s00021-015-0220-y.
[89] J. Kiendl, Isogeometric Analysis and Shape Optimal Design of Shell Structures (Ph.D. thesis), Technische Universität München, 2011.

Chapter 8

Towards predictive territory planning

Predictive territory planning could be defined as all the social, political and technical actions that need to be taken in advance to ensure the sustainable management and evolution of a populated regions. Hence, the predictive territory planning is a holistic concept in which multiple knowledge areas are involved. Among all of them, this chapter focuses on how the computational wind engineering, used in combination with level set methods, can help to achieve such challenging objective.

The chapter is organized as follows. First of all, the advantages of using a level set discretization to deal with dirty input geometries are discussed. Secondly, an application case of such techniques to analyse the wind flow over a city digital model is shown. The last section collects some thoughts about the applicability of the method to real management strategies.

8.1 Introduction

Nowadays, Computer Aided Design (CAD) is a well established tool in the design pipeline of all the engineering areas. Thanks to CAD modelling, it is relatively easy to sketch and modify any conceptual design, regardless the complexity of its geometrical features. This, in combination with numerical analysis has allowed designers and engineers to have a rapid virtual prototyping lab in their own laptops.

Having said this, it is clear that the ideal scenario would be to directly use the CAD model created during the design stage to run the simulation, avoiding thus the need of generating an intermediate simulation model from the CAD-based geometry. This idea

gave rise to the so called IsoGeometric Analysis (IGA) [26, 50] or ImmersoGeometric Analysis methods [58, 112] in which the simulation is done on top of the CAD geometries.

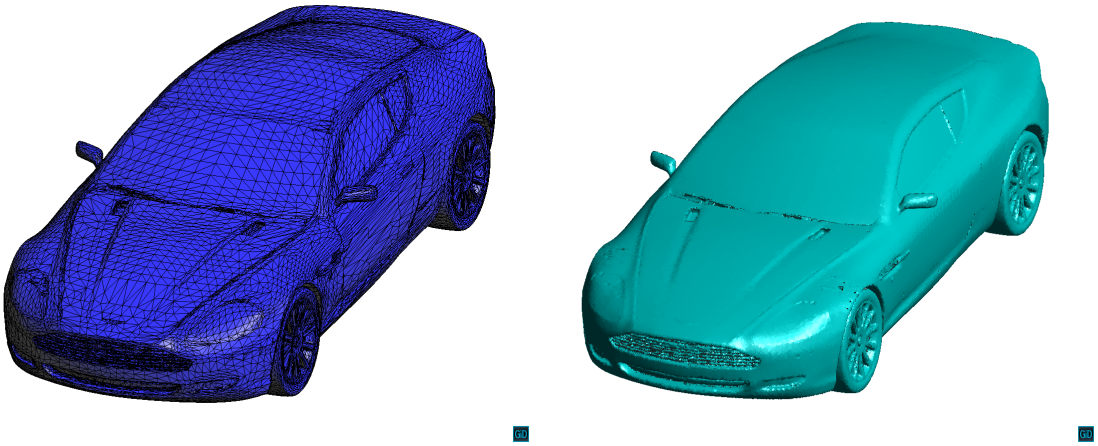
Unfortunately, there are cases in which the CAD model is not suitable for the numerical analysis and thus requires modifications to become simulation-compatible. On the one hand, it might happen that the CAD model has a high level of detail which is neither required nor suitable for the numerical simulation. This situation commonly happens when dealing with CAD models from architecture or civil engineering projects, in which the level of detail required to do the blueprints or the measurements is much beyond that one demanded to do an accurate CFD analysis of the structure. On the other hand, it might also happen that the input geometry is given in a low fidelity format such as the STereoLithography (stl). These "dirty" input geometries can be rarely meshed out of the box since the mesh generation algorithms are rather sensitive to the geometrical imperfections.

Traditionally, these issues have been addressed by manually repairing the input geometries. This implies to clean up and fix all the possible defects, namely duplicated entities, holes and overlapped surfaces, in order to create a watertight meshable geometry. As it can be easily guessed, this task rapidly turns into an extremely time and effort consuming, or even unaffordable, operation, specially when dealing with real complex geometries. Indeed, the geometrical repairing and meshing are reported to take up to the 50% of the total simulation time in real engineering applications [26].

In this context, the body conforming preprocess bottleneck can be avoided by using an implicit description of the bodies. By doing so the defects of the input geometry are automatically filtered by the level set calculation algorithm, which is an always defined operation, meaning that all the human geometrical operations are completely bypassed. Hence, thanks to the robustness of the level set calculation operation the preprocessing of complex geometries is extremely simplified and accelerated as it only requires the generation of an unfitted background computational mesh. Furthermore, this opens the possibility of using of fast Octree-based algorithms [24] to generate the background mesh, which can be upgraded later on by using a distance-based anisotropic refinement [30].

These ideas are presented in Fig. 8.1 for an Aston Martin Vanquish CAD model. As it can be observed in Fig. 8.1a, the stl input geometry features a high level of detail which makes impossible to create a watertight volume mesh out of the box. Instead of repairing the stl file, something that seems a priori to be extremely complicated, the

geometry can be implicitly represented in the background mesh. For a reasonable level of discretization, the distance calculation algorithm yields the level set function depicted in Fig. 8.1b, which features a sufficient level of detail to run a CFD simulation.



(a) Input stl geometry. (b) Level set representation.

Figure 8.1: Aston Martin Vanquish level set calculation test.

8.2 A civil engineering application: city wind flow

In this section the previously described advantages of implicit geometry representations are exploited in the context of predictive territory planning. More specifically, the application at hand consists in the wind CFD analysis of a portion of a real city.

To efficiently achieve this, the computational domain is created as a prismatic vertical extrusion of the region of interest boundaries. Then the bottom face of the prism is removed such that the inferior part of the computational domain matches the ground surface, whose geometry can be obtained from a digital terrain elevation model. Once this operation is completed, the computational domain is ready to be meshed.

Note that so far only the terrain, with no buildings over it, is represented by the volume mesh boundaries. Taking into account that the public databases from which the buildings geometries can be retrieved commonly involve a rather poor quality, repairing the input files representing the buildings is fairly impossible when large urban areas are to be analysed. Instead, the input file describing the building geometries can be literally thrown into the computational domain in order to implicitly represent these in the volume mesh.

This approach is applied to solve the wind flow over a vast extension (2.5x1.5km) of the Montigalà district of the city of Badalona (Spain). Fig. 8.2 depicts a map of the region of interest while Fig. 8.3 does so with the computational model, including the level set representation of the buildings and the computational box wire frame. As it can be observed, the ground is described by the lower surface of the computational box (Fig. 8.3a) while Figs. 8.3b and 8.3c zoom in the buildings implicit representation. The 62M elements background mesh yields a resolution around 1m, which is considered enough taking into account that the objective of this kind of analysis is to have an overall understanding of the flow over a vast region of terrain. A constant air inlet of 10m/s is imposed in the south-west side of the domain (left edge in Fig. 8.2). With regard to the wall behavior, both the computational domain faces and the embedded skin are modeled as slip walls as the boundary layer viscous effects cannot be properly represented with this mesh resolution. Owing to the dimensions of the problem, it is impossible to solve it in a desktop machine and the use of an HPC facility is thus required. In this case the problem is solved in the in-house cluster Acuario using 192 MPI processes.

Fig. 8.4 presents some streamlines from the obtained velocity field. It can be observed that the distance based approach is capable of capturing the main features of the wind effects, also including the urban canyon phenomenon (Fig. 8.4b), which is of particular interest in the urban wind engineering context.

8.3 Conclusion

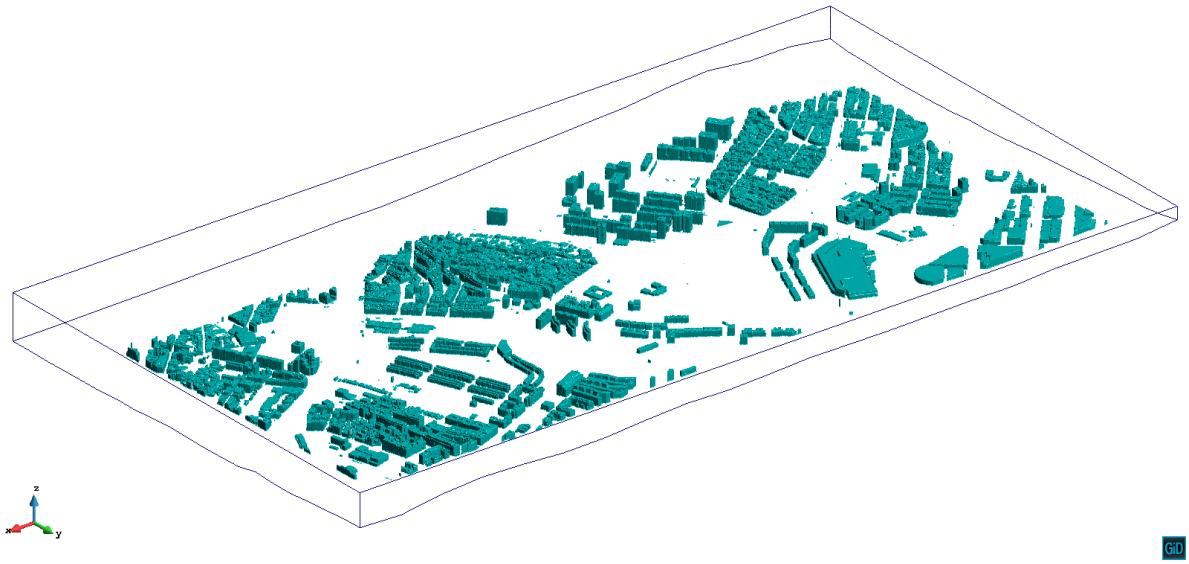
This chapter aims at exploiting the robustness of the implicit level set representation from a completely different perspective. This is to simplify the preprocessing of extremely "dirty" input geometries in the context of civil wind engineering. Hence, all the tedious geometrical repairing operations, which might be unaffordable in complex scenarios as the one presented in the previous section, are automatically bypassed thanks to the inherent capabilities of the distance calculation algorithm. However, such enhanced robustness comes at the price of requiring a way finer volume mesh discretization to achieve a reasonably accurate resolution of the embedded boundaries. In consequence, a scalable MPI implementation, as well as the access to an HPC facility, becomes mandatory to efficiently transfer this technology to real applications.

It is important to remark that the results in this chapter are a proof-of-concept of a future research line that arises from the main developments of this thesis. Hence,

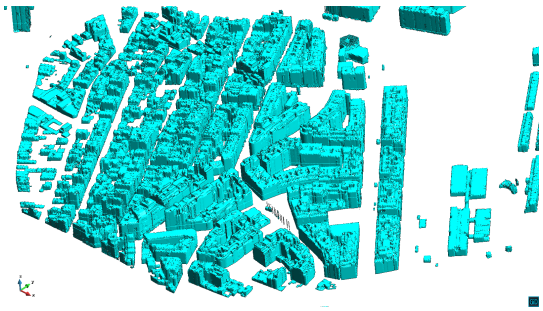
Figure 8.2: Wind flow over cities. Montigalà district (Badalona, Spain) domain of interest.

this chapter only focuses on the modeling simplification, meaning that other important aspects of the wind engineering such as the turbulent wind inlet BC are not considered yet. Despite this, the obtained results are promising and make possible the gradual addition of extra features.

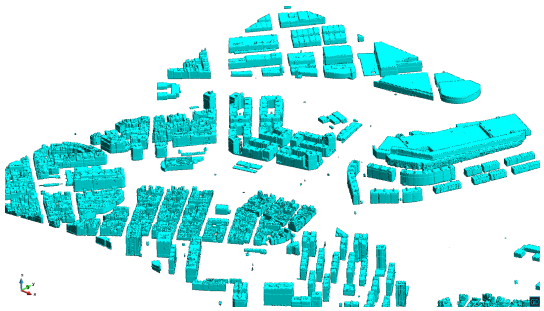
A feasible enhancement that would be of great utility in the civil engineering field is the combination of the embedded CFD solver with a pollutant transport one in order to assess how an accidental or chimney contaminant plume would affect a region. Furthermore, the ideas in this chapter can be perfectly used in combination with a body conforming representation. This opens the possibility to select a building of interest, in which the wind effects require to be accurately captured, with an embedded representation of the surroundings.



(a) 3D view.

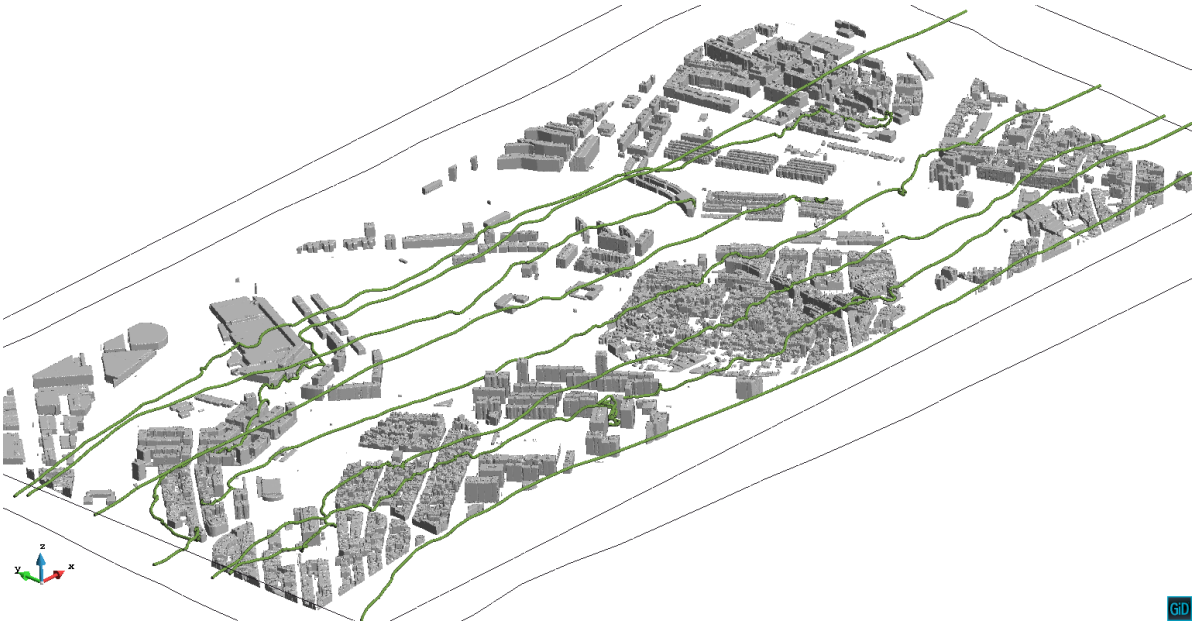


(b) Level set detail 1.

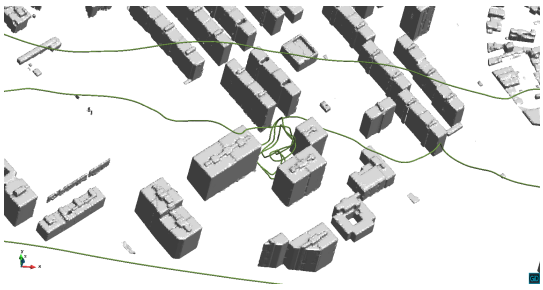


(c) Level set detail 2.

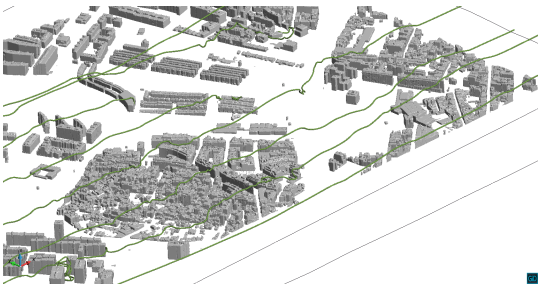
Figure 8.3: Wind flow over cities. Montigalà district (Badalona, Spain). Computational model. The boundaries of the computational domain are represented by the blue wire frame. The level set representation of the buildings is showed in cyan.



(a) Complete view.



(b) Streamlines detail 1.



(c) Streamlines detail 2.

Figure 8.4: Wind flow over cities. Montigalà district (Badalona, Spain). Obtained streamlines.

Chapter 9

Conclusion

This last chapter presents the closure of the work. To that purpose, the achievements of each one of the previous chapters are discussed first. Secondly, the generic conclusions that arise after having finished the thesis are presented as a closure. Finally, the last section outlines the future research lines and improvements.

9.1 Achievements

The first remarkable scientific contribution of this thesis is the discontinuous Navier–Stokes embedded formulation for the imposition of the slip BC presented in Chapter 4. Such formulation, which is intended to be used in combination with a discontinuous level set representation, modifies the standard FE space in the intersected elements by the Ausas discontinuous one. This substitution, in combination with a weak imposition of the no penetration BC, namely slip BC, allows the resolution of CFD problems involving not only volumetric bodies but also thin-walled ones. The experiments, which involve low and high Re cases, show that the $h^{3/2}$ convergence rate of the Ausas FE space is kept. Furthermore, the method is of the same order of accuracy as the body fitted and embedded approaches that the tests compare to.

In Chapter 5 the discontinuous slip formulation is enhanced by including a Nitsche-based imposition of the general Navier-slip condition. Aside of keeping the capability of the previous formulation to deal with membrane-like bodies, this improved version makes possible to model any wall behaviour from the slip to the no-slip limits. The experiments show a convergence rate around $h^{3/2}$ when the wall BC approaches the

slip limit. However, the convergence rate deteriorates to h when the wall effects become relevant, that is to say when the wall BC approaches the no-slip limit. This is associated to the inability of the Ausas FE space to properly capture the gradient in the intersected elements.

Concerning the computational efficiency, none of these formulations have neither blending elements nor extra DOFs. This becomes in an advantage when working in a distributed memory environment since there is no extra communication overhead. Furthermore, the matrix graph calculation is not required to be constructed each time the level set is updated, something that would definitively be a computational bottleneck when dealing with moving boundary problems (i.e. FSI).

Complementary, Chapter 5 proves the utility of the pseudo-compressibility term that is added to the mass conservation equation in Chapter 3. Hence, the ability of such extra term to keep the pressure bounded is evinced by solving a test case involving an isolated fluid cavity with no Neumann BC. It has to be highlighted that this feature becomes crucial for the application of the developed methods to real engineering problems involving ill-conditioned level set functions, which may come either from "dirty" input geometries as it is discussed in Chapter 8, or from moving boundary problems involving complex bodies.

In Chapter 6 the discontinuous Navier-slip formulation is further validated by reproducing an *in vitro* experiment that consists in a phantom model of the aortic dissection cardiovascular disease. The obtained results are in agreement with both the experimental data and the literature numerical simulations. Besides this, the results also prove the utility of the formulation to model the flow around thin biological tissues.

Chapter 7 extends all the previous embedded CFD developments to be applied in the resolution of FSI problems. In other words, it can be said that the thesis objective of creating the VWT framework is ultimately accomplished in this chapter. The FSI coupling is achieved by the use of a black-box embedded interface residual Gauss-Seidel iteration that makes possible the resolution of strongly coupled problems. The robustness and performance of the embedded FSI solver is tested by solving several examples involving both volumetric and thin-walled structures. In addition, the accuracy and efficiency are also assessed by comparing the obtained results with the ones of a body fitted solver. The results prove that using a non-conforming discretization in the fluid domain neither compromises the quality nor the convergence of the solution.

Chapter 7 also includes an improvement for the BC imposition of the FM-ALE algo-

rithm mesh motion problem. This enhancement, whose computational cost is negligible compared with the rest of the FM-ALE operations, allows treating the embedded interfaces in a sort of Lagrangian manner such that the movement of the FM-ALE virtual mesh coincides (in a variational sense) with the embedded skin within the cut elements. This improvement ensures that the movement is large enough to guarantee a proper initialization of the historical variables but also small enough to prevent the common element distortion issues associated to mesh motion problems. Complementary, Chapter 7 details the implementation of the embedded FSI coupling by providing the guidelines to build an easily extensible black-box coupling framework based on object oriented design patterns.

Furthermore, it is important to highlight that the last example in Chapter 7 entails the simulation of a real scale extremely lightweight membrane structure during a strong wind episode. The results confirm the capability of the embedded FSI solver to robustly deal with the target application established at the beginning of the thesis.

Last but not least, Chapter 8 presents an alternative civil engineering application in which level set based approaches can be useful. In this case the robustness of the level set method, which makes possible to efficiently deal with poor quality input geometries, is exploited to easily generate large scale wind flow simulations.

9.2 Closure

After having discussed all the achievements in the previous section, it can be said that fixed mesh methods can play an important role in nowadays computational engineering. Thanks to the level set representation of the bodies, this family of methods can easily deal with arbitrary moving interfaces and thus bypass all the common problems that ALE-based body conforming approaches have. Furthermore, level set based approaches can also help to make the preprocess more efficient, specially when dealing with "dirty" input geometries or when a volume mesh has to be created from a volumeless thin-walled structure.

These advantages come however at the price of requiring some extra complexities, such as the weak BC imposition or the use (and implementation) of an efficient level set calculation algorithm. Besides this, body conforming approaches are always slightly more accurate than embedded ones. Aside of the fact that the weak BC imposition may affect in this regard, the implicit representation of geometries is inherently less

accurate than the body fitted one since some geometrical features, such as sharp corners in volumetric bodies and partial intersections in thin-walled ones, cannot be represented by a level set function. As a consequence, the unique way to minimize such geometrical representation error is to refine the mesh in those regions where the embedded bodies are expected to move across.

Taking into account such disadvantages, the embedded approach should not be considered as a substitute of the standard body conforming ones. Instead, it has to be thought of as an always robust alternative to be applied in those cases in which the standard methods are expected to perform badly.

Recalling the motivation of the thesis in Chapter 1, it can be said that the initially stated objectives have been all accomplished. The embedded framework that has been developed is capable of dealing with both CFD and FSI problems involving any type of bodies, including membrane-like ones, which was one of the main demands. Altogether, these developments are the so called VWT, which can be considered ready to be applied in the resolution of real civil engineering problems.

9.3 Future research lines

This section briefly describes which are the remaining improvements and future work lines arising from this thesis. These have been divided in three categories: implementation, technical advances and other applications.

Regarding the missing implementations, they be summarized in two main tasks. The first and more urgent one is the MPI implementation of some auxiliary utilities, which are mainly related to the FM-ALE algorithm. The second one is the development and validation of a user-friendly Graphical User Interface (GUI). Right now only a mock-up version of the VWT GUI is available. These two requirements are mandatory for the effective application of the VWT toolbox in real industrial projects.

Concerning the technical improvements, these aim at limiting the effect of the implicit representation accuracy issues discussed in the previous section. On the one hand, using an edge-based level set function instead of the current elemental one could help to detect those partial intersections that occur when a thin-walled skin geometry does not completely split an element. Once such partial intersections are detected, the boundary artifacts commented in Chapter 7 could be addressed by including an extra variational term. On the other hand, both the CFD and FSI embedded solvers could benefit

from the use of a level set-based anisotropic mesh refinement strategy. This would substantially help to recover the geometrical details that are lost because of the implicit representation.

Complementary, the MVQN convergence accelerator can be also improved to avoid the inverse Jacobian matrix storage, which might involve a great memory consumption when dealing with large FSI problems. Some experiments have been carried out in this regard using a recursive implementation of the inverse Jacobian approximation. After some preliminary tests, the alternative implementation seems to be valid. Nevertheless, a thorough validation is still pending, specially to assess how the number of recursive steps used in the approximation of the Jacobian (i.e. buffer size) affects the FSI coupling convergence.

Finally, it is important to highlight that the VWT developments can be exploited in other engineering fields. Chapter 6 proves that the embedded CFD solver can be efficiently applied in the resolution of problems that involve fluid flow around biological tissues. The next step is to use the embedded FSI solver to solve problems that require introducing the deformation of the tissues such as the blood flow through the tricuspid or mitral hearth valves. Moreover, embedded methods are understood to be one of the few options that can efficiently deal with topology changes without requiring remeshing. In this regard, it could be also interesting to apply the embedded FSI solver to problems involving contact or membrane wrinkling (e.g. folding/unfolding sails or parachutes). Preliminary fluid-structure-contact interaction experiments combining the embedded FSI solver with an implicit mortar contact formulation showed promising results.

Bibliography

[1] F. Alauzet, B. Fabrèges, M. A. Fernández, and M. Landajuela. Nitsche-XFEM for the coupling of an incompressible fluid with immersed thin-walled structures. *Computer Methods in Applied Mechanics and Engineering*, 301:300 – 335, 2016.

[2] R. Ausas, F. Sousa, and G. Buscaglia. An improved finite element space for discontinuous pressures. *Comput. Methods Appl. Mech. Engrg.*, 199:1019–1031, 2010.

[3] R. Ausas, F. Sousa, and S. Idelsohn. A statically condensable enrichment for pressure discontinuities in two-phase flows. *Mecánica Computacional*, 30(4):175–191, 2011.

[4] S. Badia, F. Nobile, and C. Vergara. Fluid-structure partitioned procedures based on robin transmission conditions. *Journal of Computational Physics*, 227(14):7027 – 7051, 2008.

[5] S. Badia, F. Verdugo, and A. F. Martín. The aggregated unfitted finite element method for elliptic problems. *Computer Methods in Applied Mechanics and Engineering*, 336:533 – 553, 2018.

[6] J. Baiges and R. Codina. The fixed-mesh ALE approach applied to solid mechanics and fluid-structure interaction problems. *International Journal for Numerical Methods in Engineering*, 81(12):1529–1557, 2010.

[7] J. Baiges, R. Codina, and H. Coppola-Owen. The fixed-mesh ALE approach for the numerical simulation of floating solids. *International Journal for Numerical Methods in Fluids*, 67(8):1004–1023, 2011.

[8] J. Baiges, R. Codina, F. Henke, S. Shahmiri, and W. A. Wall. A symmetric method for weakly imposing Dirichlet boundary conditions in embedded finite element meshes. *International Journal for Numerical Methods in Engineering*, 90(5):636–658, 2012.

[9] J. Baiges, R. Codina, A. Pont, and E. Castillo. An adaptive fixed-mesh ALE method for free surface flows. *Computer Methods in Applied Mechanics and Engineering*, 313:159 – 188, 2017.

[10] D. Baumgärtner, J. Wolf, R. Rossi, P. Dadvand, and R. Wüchner. A robust algorithm for implicit description of immersed geometries within a background mesh. *Advanced Modeling and Simulation in Engineering Sciences*, 5(1):21, 2018.

[11] D. Boffi, F. Brezzi, and M. Fortin. *Mixed Finite Element Methods and Applications*. Springer, 2013.

[12] A. Bogaers, S. Kok, B. Reddy, and T. Franz. Quasi-newton methods for implicit black-box FSI coupling. *Computer Methods in Applied Mechanics and Engineering*, 279:113 – 132, 2014.

[13] A. Brooks and T. Hughes. Streamline upwind/Petrov-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible navier-stokes equations. *Comput. Methods Appl. Mech. Engrg.*, 32(1):199 – 259, 1982.

[14] E. Burman, S. Claus, P. Hansbo, M. G. Larson, and A. Massing. CutFEM: Discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501, 2015.

[15] R. Carter. Boat remains and maritime trade in the persian gulf during the sixth and fifth millennia bc. *Antiquity*, 80(307):52–63, 2006.

[16] H. Casquero, Y. J. Zhang, C. Bona-Casas, L. Dalcin, and H. Gomez. Non-body-fitted fluid-structure interaction: Divergence-conforming B-splines, fully-implicit dynamics, and variational formulation. *Journal of Computational Physics*, 374:625–653, 2018.

[17] P. Causin, J. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 194(42):4506 – 4527, 2005.

[18] T. Chung. Finite element analysis in fluid dynamics. *NASA STI/Recon Technical Report A*, 78, 1978.

[19] R. Codina. A stabilized finite element method for generalized stationary incompressible flows. *Comput. Methods Appl. Mech. Engrg.*, 190(20):2681–2706, 2001.

[20] R. Codina. Stabilized finite element approximation of transient incompressible flows using orthogonal subscales. *Computer Methods in Applied Mechanics and Engineering*, 191(39):4295 – 4321, 2002.

[21] R. Codina and J. Baiges. Approximate imposition of boundary conditions in immersed boundary methods. *Int. J. Numer. Meth. Engng.*, 80:1379–1405, 2009.

[22] R. Codina, G. Houzeaux, H. Coppola-Owen, and J. Baiges. The fixed-mesh ALE approach for the numerical approximation of flows in moving domains. *Journal of Computational Physics*, 228(5):1591 – 1611, 2009.

[23] R. Codina, J. Principe, O. Guasch, and S. Badia. Time dependent subscales in the stabilized finite element approximation of incompressible flow problems. *Computer Methods in Applied Mechanics and Engineering*, 196(21):2413 – 2430, 2007.

[24] A. Coll. *Robust volume mesh generation for non-watertigh geometries*. PhD thesis, Universitat Politècnica de Catalunya, 2014.

[25] J. Cotela, R. Rossi, and E. Oñate. A FIC-based stabilized finite element formulation for turbulent flows. *Comput. Methods Appl. Mech. Engrg.*, 315:607 – 631, 2017.

[26] J. Cottrell, T. Hughes, and Y. Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

[27] P. Dadvand, R. Rossi, M. Gil, X. Martorell, J. Cotela, E. Juanpere, S. Idelsohn, and E. Oñate. Migration of a generic multi-physics framework to HPC environments. *Computers & Fluids*, 80:301 – 309, 2013.

[28] P. Dadvand, R. Rossi, and E. Oñate. An object-oriented environment for developing finite element codes for multi-disciplinary applications. *Archives of Computational Methods in Engineering*, 17(3):253–297, 2010.

[29] S. Dahl, J. Vierendeels, J. Degroote, S. Annerel, L. Hellevik, and B. Skallerud. FSI simulation of asymmetric mitral valve dynamics during diastolic filling. *Computer Methods in Biomechanics and Biomedical Engineering*, 15(2):121–130, 2012.

[30] C. Dapogny, C. Dobrzynski, and P. Frey. Three-dimensional adaptive domain remeshing, implicit domain meshing, and applications to free and moving boundary problems. *Journal of Computational Physics*, Apr. 2014.

[31] M. de Tullio and G. Pascazio. A moving-least-squares immersed boundary method for simulating the fluid-structure interaction of elastic bodies with arbitrary thickness. *Journal of Computational Physics*, 325:201 – 225, 2016.

[32] B. M. DeBlois. Linearizing convection terms in the Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 143(3):289 – 297, 1997.

[33] J. Degroote, K.-J. Bathe, and J. Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Computers & Structures*, 87(11):793 – 801, 2009. Fifth MIT Conference on Computational Fluid and Solid Mechanics.

[34] J. Degroote, R. Haelterman, S. Annerel, P. Bruggeman, and J. Vierendeels. Performance of partitioned procedures in fluid-structure interaction. *Computers & Structures*, 88(7):446 – 457, 2010.

[35] J. Degroote and J. Vierendeels. Multi-solver algorithms for the partitioned simulation of fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 200(25):2195 – 2210, 2011.

[36] J. Donea, A. Huerta, J.-P. Ponthot, and A. Rodríguez-Ferran. *Arbitrary Lagrangian-Eulerian Methods*, chapter 14. American Cancer Society, 2004.

[37] C. Ellington, C. van den Berg, A. Willmott, and A. Thomas. Leading-edge vortices in insect flight. *Nature*, 384(6610):626–630, 1996.

[38] C. Farhat, K. G. van der Zee, and P. Geuzaine. Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 195(17):1973 – 2001, 2006. Fluid-Structure Interaction.

[39] C. Felippa. Introduction to finite element methods. *University of Colorado, USA*, 2001.

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

[41] C. A. Figueroa, I. E. Vignon-Clementel, K. E. Jansen, T. J. Hughes, and C. A. Taylor. A coupled momentum method for modeling blood flow in three-dimensional deformable arteries. *Computer Methods in Applied Mechanics and Engineering*, 195(41):5685 – 5706, 2006. John H. Argyris Memorial Issue. Part II.

[42] L. Foucard and F. Vernerey. An X-FEM-based numerical-asymptotic expansion for simulating a Stokes flow near a sharp corner. *International Journal for Numerical Methods in Engineering*, 102(2):79–98, 2015.

[43] A. Franci, E. Oñate, and J. M. Carbonell. Unified lagrangian formulation for solid and fluid mechanics and FSI problems. *Computer Methods in Applied Mechanics and Engineering*, 298:520 – 547, 2016.

[44] T. Gelhard, G. Lube, M. A. Olshanskii, and J.-H. Starcke. Stabilized finite element schemes with LBB-stable elements for incompressible flows. *Journal of Computational and Applied Mathematics*, 177(2):243 – 267, 2005.

[45] L. Gerardo-Giorda, F. Nobile, and C. Vergara. Analysis and optimization of Robin–Robin partitioned procedures in fluid–structure interaction problems. *SIAM Journal on Numerical Analysis*, 48(6):2091–2116, 2010.

[46] R. Glowinski, T. Pan, and J. Periaux. Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. *Computer Methods in Applied Mechanics and Engineering*, 151(1):181 – 194, 1998. Containing papers presented at the Symposium on Advances in Computational Mechanics.

[47] R. Glowinski, T.-W. Pan, and J. Periaux. A Lagrange multiplier/fictitious domain method for the Dirichlet problem — generalization to some flow problems. *Japan Journal of Industrial and Applied Mathematics*, 12(1):87, Feb 1995.

[48] M. Glück, M. Breuer, F. Durst, A. Halfmann, and E. Rank. Computation of fluid-structure interaction on lightweight structures. *Journal of Wind Engineering and*

Industrial Aerodynamics, 89(14):1351 – 1368, 2001. Bluff Body Aerodynamics and Applications.

[49] T. Hughes. Multiscale phenomena: Green's function, the dirichlet to Neumann formulation, subgrid scale models, bubbles and the origins of stabilized formulations. *Comput. Methods Appl. Mech. Engrg.*, 127(1):387–401, 1995.

[50] T. Hughes, J. Cottrell, and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer Methods in Applied Mechanics and Engineering*, 194(39):4135 – 4195, 2005.

[51] T. Hughes, G. Feijóo, L. Mazzei, and J. Quincy. The variational multiscale method-a paradigm for computational mechanics. *Comput. Methods Appl. Mech. Engrg.*, 166(1):3–24, 1998. Advances in Stabilized Methods in Computational Mechanics.

[52] T. Hughes, L. Franca, and M. Balestra. A new finite element formulation for computational fluid dynamics: V. circumventing the babuška-brezzi condition: a stable Petrov-Galerkin formulation of the stokes problem accommodating equal-order interpolations. *Computer Methods in Applied Mechanics and Engineering*, 59(1):85 – 99, 1986.

[53] T. J. Hughes, L. P. Franca, and G. M. Hulbert. A new finite element formulation for computational fluid dynamics: VIII. the galerkin/least-squares method for advective-diffusive equations. *Computer Methods in Applied Mechanics and Engineering*, 73(2):173 – 189, 1989.

[54] S. Idelsohn, J. Gimenez, and N. Nigro. Multifluid flows with weak and strong discontinuous interfaces using an elemental enriched space. *International Journal for Numerical Methods in Fluids*, 86(12):750–769, 2018.

[55] S. R. Idelsohn, F. Del Pin, R. Rossi, and E. Oñate. Fluid-structure interaction problems with strong added-mass effect. *International Journal for Numerical Methods in Engineering*, 80(10):1261–1294, 2009.

[56] S. R. Idelsohn, J. Marti, A. Souto-Iglesias, and E. Oñate. Interaction between an elastic structure and free-surface flows: experimental versus numerical comparisons using the pfem. *Computational Mechanics*, 43(1):125–132, 2008.

[57] M. Juntunen and R. Stenberg. Nitsche's method for general boundary conditions. *Mathematics of Computation*, 78:1353–1374, 2009.

[58] D. Kamensky, M.-C. Hsu, D. Schillinger, J. Evans, A. Aggarwal, Y. Bazilevs, M. Sacks, and T. Hughes. An immersogeometric variational framework for fluid-structure interaction: Application to bioprosthetic heart valves. *Computer Methods in Applied Mechanics and Engineering*, 284:1005 – 1053, 2015.

[59] J. Kim, D. Kim, and H. Choi. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *Journal of Computational Physics*, 171(1):132 – 150, 2001.

[60] D. Knoll and D. Keyes. Jacobian-free newton-krylov methods: a survey of approaches and applications. *Journal of Computational Physics*, 193(2):357 – 397, 2004.

[61] U. Küttler, C. Förster, and W. A. Wall. A solution for the incompressibility dilemma in partitioned fluid–structure interaction with pure dirichlet fluid domains. *Computational Mechanics*, 38(4):417–429, Sep 2006.

[62] U. Küttler, M. Gee, C. Förster, A. Comerford, and W. A. Wall. Coupling strategies for biomedical fluid–structure interaction problems. *International Journal for Numerical Methods in Biomedical Engineering*, 26(3–4):305–321, 2010.

[63] U. Küttler and W. A. Wall. Fixed-point fluid–structure interaction solvers with dynamic relaxation. *Computational Mechanics*, 43(1):61–72, 2008.

[64] M. Landajuela, M. Vidrascu, D. Chapelle, and M. A. Fernández. Coupling schemes for the fsi forward prediction challenge: Comparative study and validation. *International Journal for Numerical Methods in Biomedical Engineering*, 33(4):e2813, 2017. e2813 cnm.2813.

[65] R. Livaoğlu and A. Doğangün. Simplified seismic analysis procedures for elevated tanks considering fluid-structure-soil interaction. *Journal of Fluids and Structures*, 22(3):421 – 439, 2006.

[66] R. Löhner, J. D. Baum, E. Mestreau, D. Sharov, C. Charman, and D. Pelessone. Adaptive embedded unstructured grid methods. *International Journal for Numerical Methods in Engineering*, 60(3):641–660, 2004.

[67] R. Löhner, J. Cebral, F. Camelli, S. Appanaboyina, J. Baum, E. Mestreau, and O. Soto. Adaptive embedded and immersed unstructured grid techniques. *Comput. Methods Appl. Mech. Engrg.*, 197(25):2173–2197, 2008.

[68] A. Lucas. *Wind, water, work: ancient and medieval milling technology*, volume 8. Brill, 2006.

[69] M. M. Gad-el Hak. Fluid mechanics from the beginning to the third millennium. *International Journal of Engineering Education*, 14, 01 1998.

[70] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part I: Poisson and stokes problems. *Journal of Computational Physics*, 372:972 – 995, 2018.

[71] A. Main and G. Scovazzi. The shifted boundary method for embedded domain computations. Part II: Linear advection-diffusion and incompressible navier-stokes equations. *Journal of Computational Physics*, 372:996 – 1026, 2018.

[72] A. Massing, M. Larson, A. Logg, and M. Rognes. A stabilized Nitsche fictitious domain method for the Stokes problem. *Journal of Scientific Computing*, 61(3):604–628, 2014.

[73] A. Massing, B. Schott, and W. Wall. A stabilized Nitsche cut finite element method for the Oseen problem. *Comput. Methods Appl. Mech. Engrg.*, 328:262–300, 2018.

[74] A. Meurer, C. Smith, M. Paprocki, O. Čertík, S. Kirpichev, M. Rocklin, A. Kumar, S. Ivanov, J. Moore, S. Singh, T. Rathnayake, S. Vig, B. Granger, R. Muller, F. Bonazzi, H. Gupta, S. Vats, F. Johansson, F. Pedregosa, M. Curry, A. Terrel, Š. Roučka, A. Saboo, I. Fernando, S. Kulal, R. Cimrman, and A. Scopatz. Sympy: symbolic computing in python. *PeerJ Computer Science*, 3:e103, 2017.

[75] S. Minami and S. Yoshimura. Performance evaluation of nonlinear algorithms with line-search for partitioned coupling techniques for fluid–structure interactions. *International Journal for Numerical Methods in Fluids*, 64(10–12):1129–1147, 2010.

[76] R. Mittal, H. Dong, M. Bozkurttas, F. Najjar, A. Vargas, and A. von Loebbecke. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *Journal of Computational Physics*, 227(10):4825 – 4852, 2008.

[77] D. Mok, W. Wall, and E. Ramm. Accelerated iterative substructuring schemes for instationary fluid-structure interaction. *Computational fluid and solid mechanics*, 2:1325–1328, 2001.

[78] J. Nitsche. Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 36(1):9–15, 1971.

[79] J. Oleson. *Greek and Roman mechanical water-lifting devices: the history of a technology*, volume 16. Springer Science & Business Media, 1984.

[80] E. Oñate, F. G. Flores, and J. Marcipar. *Membrane Structures Formed by Low Pressure Inflatable Tubes. New Analysis Methods and Recent Constructions*, pages 163–196. Springer Netherlands, Dordrecht, 2008.

[81] E. Oñate, A. Franci, and J. M. Carbonell. Lagrangian formulation for finite element analysis of quasi-incompressible fluids with reduced mass losses. *International Journal for Numerical Methods in Fluids*, 74(10):699–731, 2014.

[82] F. Otto, R. Trostel, and F. K. Schleyer. *Tensile structures: design, structure, and calculation of buildings of cables, nets, and membranes*, volume 69. MIT Press London, 1967.

[83] C. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252 – 271, 1972.

[84] C. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220 – 252, 1977.

[85] C. Peskin. The immersed boundary method. *Acta Numerica*, 112:479 – 517, 2002.

[86] C. Peskin and D. McQueen. A three-dimensional computational method for blood flow in the heart i. immersed elastic fibers in a viscous incompressible fluid. *Journal of Computational Physics*, 81(2):372 – 405, 1989.

[87] A. Quarteroni. *Navier-Stokes equations*, pages 457–510. Springer International Publishing, Cham, 2017.

[88] P. Roache. Code verification by the method of manufactured solutions. *J. Fluids Eng.*, 124(1):4–10, 2002.

[89] R. Rossi and E. Oñate. Analysis of some partitioned algorithms for fluid-structure interaction. *Engineering Computations*, 27(1):20–56, 2010.

[90] R. Rossi, P. B. Ryzhakov, and E. Oñate. A monolithic FE formulation for the analysis of membranes in fluids. *International Journal of Space Structures*, 24(4):205–210, 2009.

[91] S. D. Roth. Ray casting for modeling solids. *Computer Graphics and Image Processing*, 18(2):109 – 144, 1982.

[92] P. A. Rudenick, B. H. Bijnens, D. García-Dorado, and A. Evangelista. An in vitro phantom study on the influence of tear size and configuration on the hemodynamics of the lumina in chronic type B aortic dissections. *Journal of Vascular Surgery*, 57(2):464 – 474.e5, 2013.

[93] P. Ryzhakov, E. Soudah, and N. Dialami. Computational modeling of the fluid flow and the flexible intimal flap in type B aortic dissection via a monolithic Arbitrary Lagrangian/Eulerian fluid-structure interaction model. *International Journal for Numerical Methods in Biomedical Engineering*, n/a(n/a):e3239, 2019. e3239 cnm.3239.

[94] P. B. Ryzhakov, R. Rossi, S. R. Idelsohn, and E. Oñate. A monolithic Lagrangian approach for fluid–structure interaction problems. *Computational Mechanics*, 46(6):883–899, 2010.

[95] M. Saeedi, R. Wüchner, and K.-U. Bletzinger. Fluid-structure interaction analysis and performance evaluation of a membrane blade. *Journal of Physics: Conference Series*, 753:102009, 2016.

[96] K. Salari and P. Knupp. Code verification by the method of manufactured solutions. Technical report, United States, 2000. Research Org.: Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US).

[97] T. Sawada and A. Tezuka. LLM and X-FEM based interface modeling of fluid–thin structure interactions on a non-interface-fitted mesh. *Computational Mechanics*, 48(3):319–332, 2011.

[98] B. Schott, S. Shahmiri, R. Kruse, and W. Wall. A stabilized Nitsche-type extended embedding mesh approach for 3D low- and high-reynolds-number flows. *International Journal for Numerical Methods in Fluids*, 82(6):289–315, 2016.

[99] R. Scotta, M. Lazzari, E. Stecca, J. Cotela, and R. Rossi. Numerical wind tunnel for aerodynamic and aeroelastic characterization of bridge deck sections. *Computers & Structures*, 167:96 – 114, 2016.

[100] S. Sicklinger, V. Belsky, B. Engelmann, H. Elmqvist, H. Olsson, R. Wüchner, and K.-U. Bletzinger. Interface Jacobian-based Co-simulation. *International Journal for Numerical Methods in Engineering*, 98(6):418–444, 2014.

[101] S.Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Siences*. Springer-Verlag New York, 1 edition, 2003.

[102] E. Soudah, P. Rudenick, M. Bordone, B. Bijnens, D. García-Dorado, A. Evangelista, and E. Oñate. Validation of numerical flow simulations against in vitro phantom measurements in different type B aortic dissection scenarios. *Computer Methods in Biomechanics and Biomedical Engineering*, 18(8):805–815, 2015.

[103] T. Spenke, N. Hosters, and M. Behr. A multi-vector interface quasi-newton method with linear complexity for partitioned fluid-structure interaction. *Computer Methods in Applied Mechanics and Engineering*, 361:112810, 2020.

[104] L. Taelman, J. Bols, J. Degroote, V. Muthurangu, J. Panzer, J. Vierendeels, and P. Segers. Differential impact of local stiffening and narrowing on hemodynamics in repaired aortic coarctation: an FSI study. *Medical & Biological Engineering & Computing*, 54(2):497–510, 2016.

[105] G. Taylor, R. Nudds, and A. Thomas. Flying and swimming animals cruise at a Strouhal number tuned for high power efficiency. *Nature*, 425(6959):707–711, 2003.

[106] J. Urquiza, A. Garon, and M.-I. Farinas. Weak imposition of the slip boundary condition on curved boundaries for Stokes flow. *Journal of Computational Physics*, 256:748–767, 2014.

[107] G. Valdés. *Nonlinear Analysis of Orthotropic Membrane and Shell Structures Including Fluid-Structure Interaction*. PhD thesis, Universitat Politècnica de Catalunya, 2007.

[108] H. Van Brummelen. Added mass effects of compressible and incompressible flows in fluid-structure interaction. *Journal of Applied Mechanics*, 76, 2009.

[109] Ö. Wikander et al. *Handbook of ancient water technology*. Brill Leiden, 2000.

[110] M. Winter, B. Schott, A. Massing, and W. Wall. A Nitsche cut finite element method for the Oseen problem with general Navier boundary conditions. *Comput. Methods Appl. Mech. Engrg.*, 330:220–252, 2018.

[111] J. Wolf, D. Baumgärtner, R. Rossi, P. Dadvand, and R. Wüchner. *Contribution to the Fluid-Structure Interaction Analysis of Ultra-Lightweight Structures using an Embedded Approach*. International Center for Numerical Methods in Engineering, 2015.

[112] F. Xu, D. Kamensky, V. Varduhn, C. Wang, S. A. Wasion, B. Sotomayor-Rinaldi, C. N. Darling, D. Schillinger, and M.-C. Hsu. *An Immersogeometric Method for the Simulation of Turbulent Flow Around Complex Geometries*, pages 111–125. Springer International Publishing, Cham, 2016.

[113] J. Young, S. Walker, R. Bomphrey, G. Taylor, and A. Thomas. Details of insect wing design and deformation enhance aerodynamic function and flight efficiency. *Science*, 325(5947):1549–1552, 2009.

[114] L. Zhang, A. Gerstenberger, X. Wang, and W. K. Liu. Immersed finite element method. *Computer Methods in Applied Mechanics and Engineering*, 193(21):2051 – 2067, 2004. Flow Simulation and Modeling.

[115] Z.-Q. Zhang, G. R. Liu, and B. C. Khoo. A three dimensional immersed smoothed finite element method (3D IS-FEM) for fluid–structure interaction problems. *Computational Mechanics*, 51(2):129–150, 2013.

[116] H. Zhao, J. B. Freund, and R. D. Moser. A fixed-mesh method for incompressible flow-structure systems with finite solid deformations. *Journal of Computational Physics*, 227(6):3114 – 3140, 2008.

[117] L. Zhu and C. Peskin. Interaction of two flapping filaments in a flowing soap film. *Physics of Fluids*, 15(7):1954–1960, 2003.

[118] O. Zienkiewicz, R. Taylor, P. Nithiarasu, and J. Zhu. *The finite element method*, volume 3. McGraw-hill London, 1977.

[119] R. Zorrilla, A. Larese, and R. Rossi. A modified Finite Element formulation for the imposition of the slip boundary condition over embedded volumeless geometries. *Computer Methods in Applied Mechanics and Engineering*, 353:123 – 157, 2019.

[120] R. Zorrilla, A. Larese, and R. Rossi. A discontinuous Nitsche based Finite Element formulation for the imposition of the general Navier-slip boundary condition over embedded volumeless geometries. *Computer Methods in Applied Mechanics and Engineering*, under review.

[121] R. Zorrilla, R. Rossi, R. Wüchner, and E. Oñate. An embedded Finite Element framework for the resolution of strongly coupled fluid–structure interaction problems. Application to volumetric and membrane-like structures. *Computer Methods in Applied Mechanics and Engineering*, 368:113179, 2020.

[122] R. Zorrilla, E. Soudah, and R. Rossi. Computational modeling of the fluid flow in type B aortic dissection using a modified finite element embedded formulation. *Biomechanics and Modeling in Mechanobiology*, 2020.