COMPARISON OF SHAPE PARAMETRIZATION TECHNIQUES FOR FLUID-STRUCTURE INTERACTION PROBLEMS

Relatore:
Prof. Luca Formaggia

Correlatori:
Prof. Alfio Quarteroni
Dr. Ing. Gianluigi Rozza

Candidato:
Davide Forti
Matricola 750112

*Dedicata a coloro che in questi anni mi hanno
sempre sostenuto e supportato: alla mia cara
nonna Graziosa e nonno Quinto, ai miei
Genitori e ad Aurora.*

Virtute duce, comite fortuna.
Cicerone (Epist. fam., X, 3)

Abstract

This master thesis describes the development in the framework of Fluid-Structure Interaction (FSI) problems of an efficient and flexible technique treating the fluid-structure interface and mesh motion problems. The main idea is to build, through a new hierarchical approach, a tool with accurate identification capabilities for both the structural rigid movement (translation/rotation) and the elastic deformation (displacement), with the possibility of facing arbitrary structural and fluid discretization schemes.

Starting from a review of the state of the art methods, used for these applications, the different shape representation techniques applied, like Free Form Deformation (FFD), Radial Basis Function (RBF) and Inverse Distance Weighting (IDW) are introduced and then compared to test their performances in terms of computational costs and achievable mesh quality. Then, in order to reduce the complexity of the geometrical model and its description, ad hoc innovative optimization techniques, like a selective approach of the RBF interpolation sites as well as a domain-decomposition approach for FFD, are presented showing clear reductions in term of computational costs. Some applications and test-cases, solved by using an open-source Finite Element library (LifeV), dealing with unsteady viscous (internal and external) flows, characterized by different Reynolds number, are shown to highlight the quality and the accuracy of the methods and their stability. For the implementation of the schemes developed, an efficient C++ object oriented code language was used, relying also on Trilinos packages.

Keywords: fluid-structure interaction, mesh motion, non-conformal meshes, shape parametrization, computational efficiency, shape registration, surface movements, free-form deformations, radial basis functions, inverse distance weighting.

Sommario

Gli obiettivi di questa tesi consistono nell'analisi e nello sviluppo, nell'ambito di problemi di interazione fluido-struttura, di uno strumento accurato e versatile per un'efficiente soluzione delle problematiche legate all'interfaccia tra la griglia rappresentante il dominio fluido e quella rappresentante la struttura, tenendo conto del loro movimento durante il processo di simulazione. Le motivazioni di questo lavoro risiedono nel fatto che in questa tipologia di problemi le difficoltà associate all'interazione tra due diversi domini fisici nascono sia dall'utilizzo di griglie non conformi all'interfaccia, sia dall'esigenza di mantenere una buona qualità delle griglie a seguito della loro interazione fisica. Pertanto, alla luce di tali considerazioni, si mostrano le motivazioni seguite per lo sviluppo di questa metodologia, attraverso un approccio di tipo gerarchico, volto da un lato a poter interfacciare domini descritti da schemi di analisi indipendenti con discretizzazioni arbitrarie, dall'altro alla possibilità di poter descrivere sia movimenti rigidi (traslazioni e rotazioni), sia deformazioni elastiche (spostamenti).
A seguito dell'analisi dello stato dell'arte di alcuni degli schemi utilizzati in questo ambito, con l'obiettivo di metterne in evidenza punti di forza e criticità, vengono introdotte le diverse tecniche di rappresentazione geometrica utilizzate, tra cui Free-Form Deformations (FFD), Radial Basis Functions (RBF) ed Inverse Distance Weighting (IDW), e confrontate in termini di valutazione dei costi computazionali, prestazioni e qualità ottenibile delle griglie. Successivamente, per ridurre la complessità ed il numero di parametri legati alle rappresentazioni geometriche adottate (per FFD ed RBF), vengono proposte delle tecniche di ottimizzazione sviluppate ad hoc (basate su domain-decomposition e selezione adattiva dei punti di controllo) che permettono, inoltre, ai fini dell'analisi, di ottenere una diminuzione dei tempi di calcolo.
Mediante l'utilizzo di una libreria ad Elementi Finiti disponibile in fluidodinamica computazionale (LifeV), vengono infine analizzati diversi casi test, basati su problemi non stazionari di flussi viscosi (interni ed esterni) caratterizzati da diversi numeri di Reynolds, volti ad evidenziare l'accuratezza, l'efficienza e la versatilità degli schemi di calcolo proposti. Per quanto concerne la parte implementativa è stata realizzata una libreria di supporto ed interfacciabile scritta in linguaggio C++ orientato agli oggetti, basata su alcuni dei pacchetti di Trilinos.

Parole chiave: interazione fluido-struttura, interfaccia fluido-struttura, movimentazione griglie di calcolo, free-form deformations, radial basis functions, inverse distance weighting.

Contents

List of Tables

List of Figures

Introduction

In the last two decades the interest for Fluid-Structure Interaction (FSI) prob-
lems has increased significantly and simulations have become more feasible thanks
to the advances in computers technology and numerical methods. Generally we
deal with a fluid-structure interaction problem when the effect of fluid dynamics
on elastic bodies and vice-versa is of primary interest. Many engineering appli-
cations, like aeroelastic instabilities in aircrafts [43, 44, 92, 93], turbomachinery
design [90], modeling of cardiovascular system [28, 29, 81] and high performance
boat dynamics [74], involve fluid-structure interaction phenomena whose analysis
is crucial for a safe and efficient design and sizing procedure. The main aspect
involved in this kind of problems is the unsteady computation of the dynamics
of different physical domains (i.e. multiphysics) and their coupling with moving
boundary conditions and meshes.
The first issue to be faced is the choice of the approach [35, 46] (either mono-
lithic or partitioned) to be used for the time integration of the coupled system of
fluid-structure equations: solving the problem in a monolithic way means solving
simultaneously flow and structure equations, while using a partitioned approach
we get separately the solution of the fluid and structural subdomains. Mono-
lithic algorithms maintain all the non linearities of the original problem and the
coupling could be taken into account directly, which is advantageous for the sta-
bility of the simulation, while partitioned ones, although they require a coupling
algorithm for the interaction between fluid and structure domains, are more flex-
ible by allowing the possibility of using different numerical schemes and separate
solvers for both physics and coupling conditions.
The second issue to deal with is the correct transfer of physical information
between the fluid and structure subdomains [52, 91, 119]: generally, in fluid-
structure interaction simulations, due to different mesh requirements for fluid and
structure, grids are not matching at the interface. The reasons for non-conformal
meshes come from different requirements on the solution, as the necessity to re-
solve viscous boundary layers for the fluid, the need of different mesh element sizes
for numerical stability issues, and, last but not least, the occurence of coupling
codes generated by different communities. For all these reasons we should be
able to deal, through interpolation and projection operators, with fluid-structure
interface problems involving non-matching meshes and, eventually, with the pres-

ence of gaps and overlaps between grids. The physical principle generally used within fluid-structure interface problems is the energy conservation [88, 120] at the interfaces, thus resulting in a conservative approach: in that way it will be possible to define an operator for transferring physical quantities (pressure loads, displacements) from one domain to the other.

The third issue concerns the dynamic mesh handling. Dealing with FSI problems, within the framework of continuum mechanics, at each iteration we need to move, once the structural subproblem has been solved and its domain has been deformed, the fluid mesh accordingly to structural deformations. Thus, in order to tackle such a problem, a mesh deformation strategy must be designed with additional requirements like (a) preservation of mesh quality and (b) low computational efforts.

Summarizing, the main aspects to be faced when dealing with a fluid-structure interaction problem are:

(i) choice of the approach and the algorithm type: either monolithic or partitioned;

(ii) setting up a methodology for transferring data across interface when different discretization grids are used for both the fluid and the structural domains;

(iii) employing an efficient and accurate mesh motion procedure for moving domains during the multiphysics interaction.

A wide variety of approaches, within monolithic and partitioned algorithms, can be found in literature dealing with different formulations of the problem and solution strategies. In this work, in order to numerically solve FSI problems, we have chosen to employ a monolithic approach based on a Geometric-Convective Explicit (GCE) time discretization scheme. This work shows the development of a tool dealing both with the fluid-structure interface (ii) and moving mesh (iii) problems. Our goal, concerning the latter, is achieved through a hierarchical approach based on the kind of displacement we have to describe: in fact we can split a generic motion into a first global large displacement (rigid translations and/or rotations) and another local one, represented by small elastic deformations. At the base of this separation we can build a standard path to be used for moving mesh problems characterized by the use of shape representations techniques like FFD [67, 82, 109], RBF [18, 31, 73] and IDW [123]. In particular, by investigating their limitations and advantages we will classify them into a standard path from which we will be able to select, on the base of the displacement to be described, the most appropriate technique to be adopted. These techniques are based on the construction of flexible and low dimensional maps, both for the fluid and the structure domains, by introducing a small set of control points whose displacements induce the shape deformation.
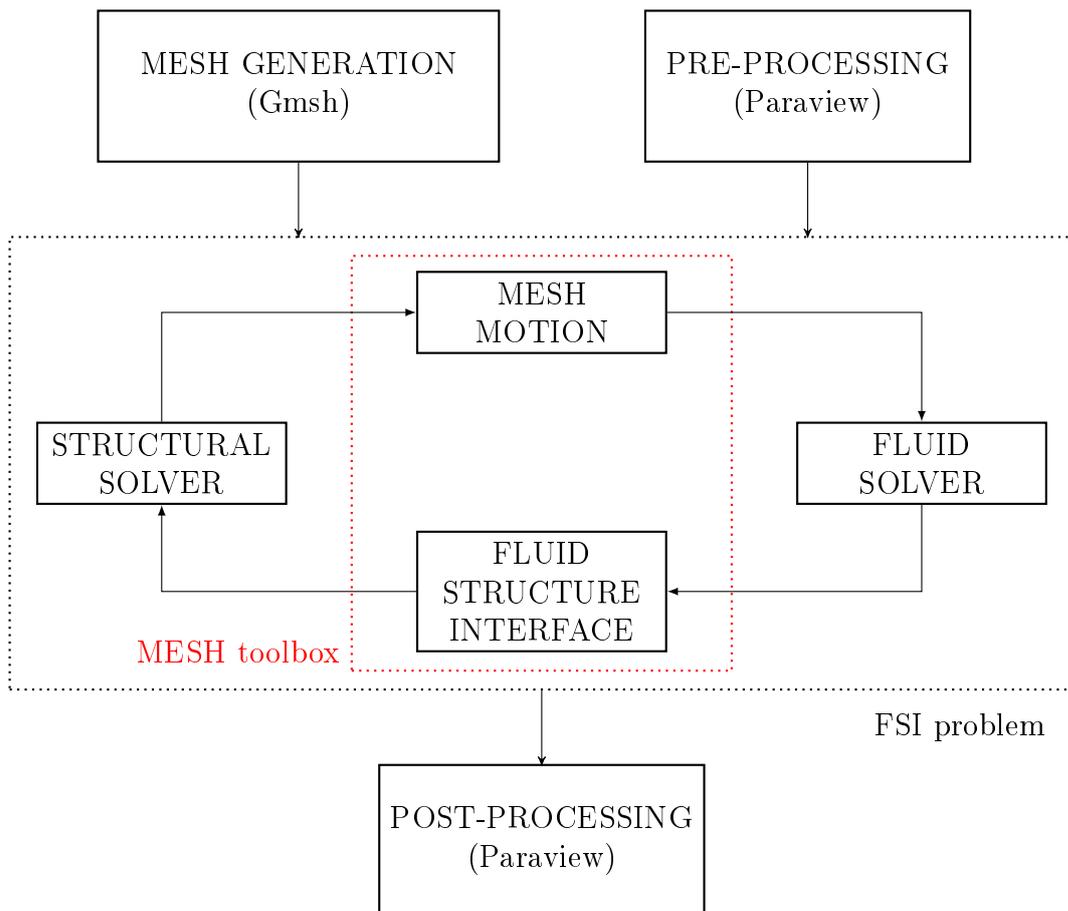
Figure 1: Algorithm of the FSI simulation process.

To be more specific, we can formalize the problem as follows: denoting by $\hat{\Omega}^F \subset \mathbb{R}^3$ and $\hat{\Omega}^S \subset \mathbb{R}^3$ the initial domain configurations for the fluid and the structure, respectively, thanks to the solution of the structural sub-problem we obtain the displacement field $\varsigma \in \mathcal{D}_S$ leading to structural deformed domain Ω^S:

$$\varsigma : \hat{\Omega}^S \longrightarrow \Omega^S \tag{1}$$

by indicating with $\mathcal{D}_S := H_0^1(\hat{\Omega}^S)$ the space of kinematically admissible structural displacements. At this point, once we update the structural configuration we need also to deform, before advancing the FSI simulation in time, the fluid domain accordingly. Therefore, such a problem could be set as:

find the fluid displacement field $\zeta : \hat{\Omega}^F \to \Omega^F$, such that $\zeta \in \mathcal{D}_\mathcal{F} := H_0^1(\hat{\Omega}^F)$,
that guarantees no compenetrations at the interfaces between fluid and structure
domains, that is $\Gamma^F = \Gamma^S$,

$$\tag{MMP}$$

where Γ^F and Γ^S indicate the fluid and structure interfaces in the deformed configuration, respectively, and Ω^F the new fluid configuration. From a practical
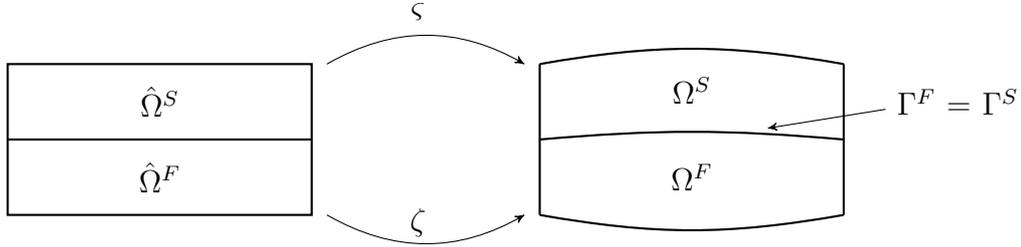
Figure 2: Schematic visualization of the condition to be satisfied at the fluid-structure interface.

point of view, extracting the displacement values ς on Γ^S, in order to find ζ for the fluid domain we proceed as follows:

(i) split ζ in two components: ζ_L representing local deformations, and ζ_G, for global (rigid translations and rotations) ones;

(ii) identify, through an eigenvalue analysis of the structural inertia matrix for the initial and final configurations, the global component ζ_G;

(iii) recover, employing FFD, RBF or IDW methods, local elastic deformation ζ_L;

(iv) compute the total fluid displacement field as sum of the previous components evaluated: $\zeta = \zeta_G + \zeta_L$.

Then, in order to reduce the computational costs involved in (iii), ad hoc innovative techniques will be proposed, like a selective approach developed to find the RBF interpolation sites as well as a domain-decomposition approach used to handle FFD techniques. In particular, concerning RBF strategies, our purpose is to identify a suitable methodology, within the FSI framework, to find the minimum number and the best location of control points used by such shape parametrization techniques to describe shape deformations. In order to tackle this problem, through the computation of the structural eigenmodes within a bandwidth of interest, it will be possible to have an *a priori* knowledge of the structural behavior, as well as its shape deformations, during the FSI simulation process. To this end, considering each eigenmode computed as a target shape that we should be able to describe, by means of an optimization procedure it will be possible to identify the (possible) best set of control points to be employed within FSI simulations.

The goal of this thesis is to assess the quality and efficiency of the strategies introduced applying them to three dimensional unsteady fluid-structure interaction problems of both internal and external viscous flows; the simulations will be performed using an *open-source* Finite Element (FE) library, namely LifeV [3, 49]. The core of such a library is written in C++ language characterized by a fully flexible hierarchical design. All the C++ object oriented modules developed here have been collected into a tool library, and the purpose of these examples is also to highlight the versatility of the this tool linked, here, to the FSI solver adopted.

With respect to the previously introduced background, the whole work has been organized with the following structure:

Chapter 1: in this first chapter a brief review of the state of the art of the methods commonly used to deal with fluid-structure interface and mesh motion problems is presented; they are analyzed to highlight their advantages and limitations in order to outline all the features to be introduced and/or improved by this work.

Chapter 2: we introduce the shape representation techniques used like FFD, RBF, as well as the multivariate interpolation method employed, namely IDW. We discuss their properties and, by applying them to several test, we compare their computational costs and geometrical representation capabilities. Some applications of these techniques are also presented to highlight their potential for their use within Reduced Order Modeling (ROM).

Chapter 3: the methodology used for the identification of the structure rigid translations/rotations is firstly described and then tested thanks to several examples. The optimization algorithms used for identifying the best number and position of control points for RBF and to handle FFD techniques are also presented and applied to some 2D and 3D problems.

Chapter 4: this chapter provides a description of the equations and algorithms used to solve fluid-structure interaction problems. In particular we detail the monolithic approach and the solution strategy adopted for the numerical FSI analysis performed with LifeV.

Chapter 5: the methods developed are tested within two problems characterized by different Reynolds numbers in laminar regime. The first dealing with an FSI problem of an external fluid flow, while the other relies in an internal fluid flow inside a cylindrical straight vessel.

All the work presented here has been developed in the Chair of Modeling and Scientific Computing (CMCS) of the École Polytechnique Fédérale de Lausanne (Switzerland), in the framework of the international exchange European Union program Erasmus.

Milano and Lausanne, 5 October 2012.

Chapter 1

An introduction to FSI problems

Nowadays the complexity of geometrical models is increasing for modern design applications. It is not unusual for a Computer-Aided Design (CAD) model to use about a thousand curves and surfaces to represent complex shapes like an aircraft. This level of complexity underscores the importance of automation, and, in this way, the adoption of strategies in order to reduce the geometrical model order description would be necessary. Thus, within any multidisciplinary application like fluid-structure interaction, the problem of setting a consistent and accurate shape parameterization strategy arises. The shape parameterization should be also compatible with, and adaptable to, various analysis issues in the FSI framework, such as the fluid-structure interface problems, which consist in the transfer of data across non-matching grids, and the dynamic mesh handling, concerning the motion of the fluid mesh due to the structural displacement. Therefore, at the basis of shape parameterization strategies there is the idea of reducing the complexity in the description of the geometry of solid objects, as well as their deformations, such that we can represent them through low dimensional spaces, instead of using geometrical properties themself. In literature a wide range of applications dealing with shape parameterization strategies is present, for example within optimization problems in aerodynamics [104] and haemodynamics [82] fields, and sizing procedures (e.g. sail boat design [74]). It is clear that, in such situations, the adoption of shape parametrization strategies is necessary, since in order to modify iteratively an initial design shape, we will need to displace only a small set of control points that control the shape deformation, in order to gain the optimal one. Such an approach would increase on one hand the reliability of the optimization process, while, on the other, would reduce drastically the computational effort spent to reach the target shape configuration. At the basis of these previous considerations the idea of using shape parameterization strategies in the FSI framework relies: in such situations we deal with deformable objects, whose deformations are governed by a multiphysics interaction with fluid flows. Thus, in that case, shape parametrization techniques give the possibility of representing the dynamics of the structure and fluid mesh configurations by means

of a small set of parameters whose displacements control the shape deformations. In detail, at each FSI simulation time-step, once we solve the equations governing the physics of the fluid flow, we need to compute the structural deformations due to the effect of the fluid motion. In this way we get, as solution of the structural sub-problem, its deformations, resulting in a new domain (and mesh) configuration. It is clear, in such a way, that in order to advance the FSI simulation process in time, we need to update, also, the fluid mesh configuration in order to match the deformed structural one at the interface. In this framework, dealing with deformable object, the adoption of shape parameterization strategies like Free-Form Deformations, Inverse Distance Weighting and Radial Basis Functions, would give to possibility of describing complex structural configurations (e.g. a whole aircraft), and their deformations, by means of a small set of control points, thus reducing significantly the complexity of the geometrical model; indeed, in this way, we could achieve great reductions in the computational effort, within the updating of fluid mesh configuration, since to impose its deformations we have only to move few control points.

(a) Initial configuration.   (b) Deformed configuration.
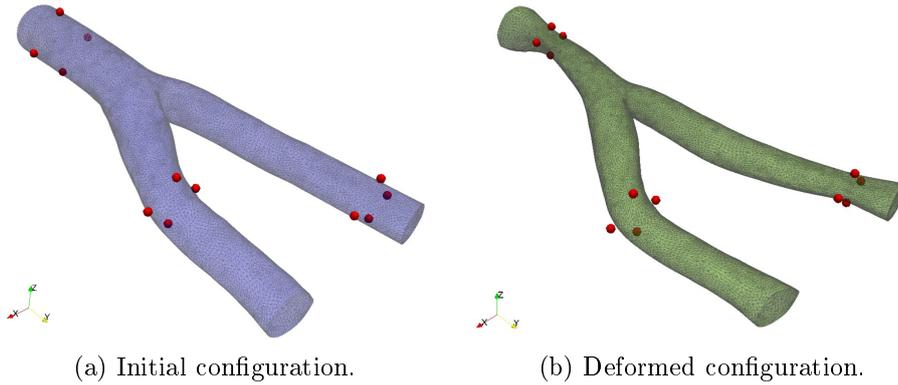
Figure 1.1: Example of application of shape parametrization within the mesh motion problem: deformation of a carotid artery bifurcation, discretized with 25436 grid nodes, centered by RBF adopting 12 control points (red bullets).

1.1 Motivations

Fluid-Structure Interaction problems are generally characterized by the unsteady motion of the domains representing both the fluid and the structural parts, due to their multiphysics interaction as shown in Figure 1.2. A possible approach to be employed, in order to deal with such problems, consists in solving the fluid flow equations on a fixed grid, within a so-called Eulerian approach [38]: it is a method of studying both the fluid motion and the mechanics of deformable bodies by considering volumes/elements at fixed locations in space.
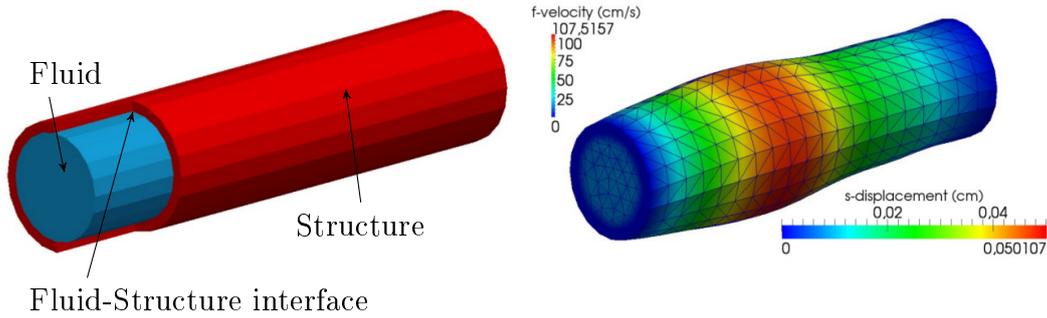
Figure 1.2: Example of a FSI problem: on the left-hand-side we illustrate by different colors the domains involved in the multiphysics problem; on the right-hand-side we report a numerical solution of the discretized problem (structural displacement and fluid velocity) obtained with LifeV.

On the other hand, in the framework of FSI problems, a classical approach that allows to overcome the adoption of a fixed computational grid is the Arbitrary Lagrangian Eulerian (ALE) method, where the fluid grid nodes are moved arbitrary inside the fluid domain, following the movement of its boundaries, and, in detail, of the fluid-structure interface due to structural deformations. Nevertheless, concerning the ALE approach, new issues arise: at each FSI time iteration, once known the deformed shape configuration of the structure domain, we need to:

(i) evaluate the displacement values of the fluid nodes lying over the interface on the basis of the known structural deformed configuration, when dealing with either non-conformal meshes or different discretization schemes used to represent both the fluid and the structure domains (fluid-structure interface problem);

(ii) update the fluid mesh configuration on the basis of the motion computed for the nodes lying over the fluid interface (mesh motion problem);

(iii) reduce, as most as possible, the computational effort related to the dynamic mesh handling;

(iv) preserve the quality of the whole fluid mesh during the mesh motion process;

(v) guarantee that the global solution does not depend on the mesh motion procedure;

(vi) be able to deal with complex domains geometries.

Therefore, within the ALE approach, a fluid-structure interaction problem would become not only a two fields (fluid and structure) but even a three fields coupling

problem (fluid, structure and mesh). A wide range of approaches have been proposed dealing with mechanical analogies [41] (spring or pseudo-solid analogy) or based on Laplacian smoothing [56]. In this work we aim at proposing, through a new hierarchical approach, a numerical strategy able to tackle all the issues highlighted. In this way, through a comparison of different shape parameterization methods, like Free-Form Deformation, Inverse Distance Weighting and Radial Basis Functions it would be possible to establish a standard path to be followed, in the FSI framework, within the fluid-structure interface and mesh motion problems. Summarizing, our idea relies on the use of shape parameterization strategies [103] in order to get both a reduction of the computational costs, and a great accuracy for the mesh handling [33, 64], with respect to the state of the art methods that have been used for such applications.

1.2   Background

Nowadays, an increasing number of engineering applications involve the study of fluid-structure interaction phenomena, whose solution is crucial, and of primary interest, for an efficient and safe design. In particular, it is possible to highlight the central role played by FSI analysis in those sizing procedures involving lightweight aircrafts [42, 99], long-span suspension bridges [114], haemodynamics devices [69, 71, 81] and high performance sail boats [73, 74].
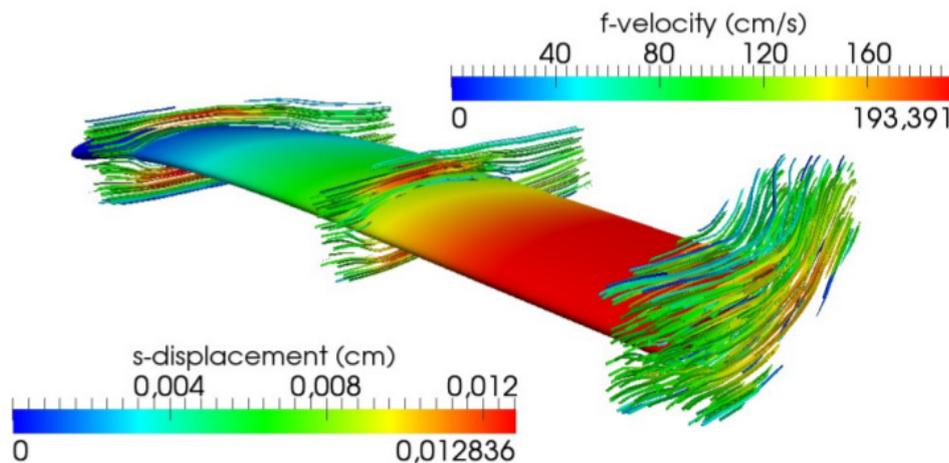
Figure 1.3: Example of a FSI solution, obtained with LifeV, concerning the fluid flow past a high-flexible wing-like structure at low Reynolds number ($\mathbb{Re} \approx 1000$): visualization of both the structural displacement contour and the flow streamlines.

Due to the increasing global competition and, also, to the advance in computer

technology, it has been possible in the last two decades to appreciate a spread of numerical strategies designed to solve efficiently FSI problems, in terms of reliability of the achieved results with reduced computational costs. In detail, within the framework of FSI problems, a great amount of research studies have been devoted to the development of low-costs numerical strategies able to perform on one hand the spatial coupling of multiphysics domains in order to deal with arbitrary discretization schemes adopted for both the fluid and the structure, and on the other, to reduce the computational costs involved in the dynamic mesh handling within the ALE-based approach. Within such a framework, it is worthwhile to survey the evolution in recent years of this scenario, in order to highlight both the limitations and the advantages of the available state of the art numerical strategies adopted to tackle both the fluid-structure interface and mesh motion problems to make a comparison between them and, at the basis of their drawbacks, to schedule the objectives to be achieved by this and related future works.

1.2.1 Fluid-structure interface problem

In many engineering applications involving FSI phenomena, generally, we need to be able to deal with different discretization schemes adopted for both the fluid and structure domains. In fact, in most cases, we need to use a much finer mesh for the fluid part instead of the structural one, due to the strict requirements related to the numerical stability of the flow analysis, and, moreover, since mainly the fluid interface zone is characterized by higher velocity gradients (e.g. boundary layer) [55]. In this way, one of the requirements of the FSI computation is related to the possibility of handling the transfer of physical data across non-matching grids. It is straightforward that, in order to deal with such a problem, we need projection and/or interpolation operators. In literature many different strategies have been developed to transfer data across non-matching grids, such as nearest neighbor interpolation [32], projection methods [25, 72, 79], and methods based on interpolation splines [18, 111, 112]. Most often, the main idea relies on the fact that energy should be conserved at the interface. Thus, the coupling scheme should preserve the equivalence of virtual work performed by fluid loads and structural forces [18]:

$$\delta W = \delta \boldsymbol{\varsigma}^T \mathbf{f}_S = \delta \boldsymbol{\zeta}^T \mathbf{f}_F, \tag{1.1}$$

being $\delta \boldsymbol{\varsigma}$ and $\delta \boldsymbol{\zeta}$ the nodal values of virtual displacements for the structure and fluid domains, while $\mathbf{f}_S$ and $\mathbf{f}_F$ indicate the structural and fluid loads, respectively.
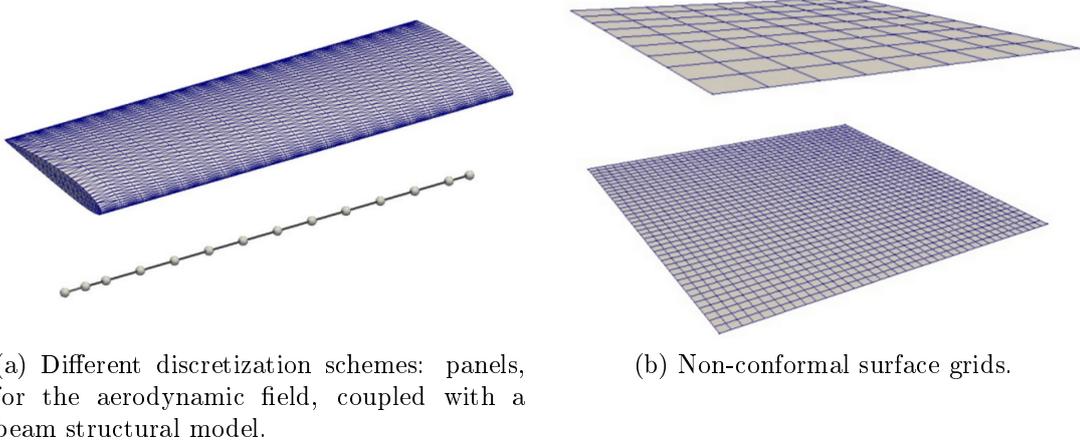
(a) Different discretization schemes: panels, for the aerodynamic field, coupled with a beam structural model.

(b) Non-conformal surface grids.

Figure 1.4: Visualization of possible non-matching fluid-structure grids.

In order to express the relation between the two displacement entities, it is possible to introduce a transfer matrix $\mathbf{H}$, such that:

$$\boldsymbol{\zeta} = \mathbf{H}\,\boldsymbol{\varsigma}. \tag{1.2}$$

When we assume that δW is equal to zero, for each arbitrary $\delta\boldsymbol{\varsigma}$, we gain the relation between the structural virtual displacements and the fluid ones

$$\delta\boldsymbol{\zeta} = \mathbf{H}\,\delta\boldsymbol{\varsigma}. \tag{1.3}$$

Thus, concerning the load transferring, thanks to (1.1) and (1.2), we obtain:

$$\mathbf{f}_S = \mathbf{H}^T\,\mathbf{f}_F. \tag{1.4}$$

In this way, when a transfer matrix $\mathbf{H}$ exists, it is possible to obtain the so-called "conservative" coupling approach, that relies on the fact that energy should be conserved at the interface [18, 93, 121]. With such a strategy, only one transformation matrix is needed to perform both the transfer of displacements and pressure loads between the two discrete interfaces. On the other hand, in order to tackle fluid-structure interface problems, two different matrices could be involved in transferring pressure loads and displacements, leading to a "consistent" approach. In general, to obtain a consistent interpolation, a constant pressure/displacement field should be exactly interpolated across the interfaces; in practice, such an approach, results in the fact that the row-sum of $\mathbf{H}$ should be equal to one. A wide range of coupling methods, based on multivariate interpolation schemes, have been used in order to find the coupling matrix $\mathbf{H}$. Among

them, the Nearest Neighbor Interpolation (NNI) strategy [115] is very likely the simplest: in this method, the generic nodal value belonging to the unknown grid, is forced to be equal to the value of its closest node belonging to the known grid. In this way the transformation matrix becomes a boolean matrix, with a single one in each row, leading, indeed, to a consistent interpolation approach. Such a strategy, although on one hand is characterized by a certain efficiency of implementation (basically based on a search algorithm), on the other suffers from some limitations. As a matter of fact, when the field to be interpolated is not smooth enough, or when the fluid and structure grids differ significantly, (e.g. the case represented in Figure 1.4b), the achievable results would be inaccurate. In the framework of the numerical strategies used to deal with fluid-structure interface problems, it is possible to highlight the class of Weighted Residual Methods (WRM) [25]. With such an approach, relation (1.3) is found through the imposition of the conservation of loads and displacements, using a weak formulation, across the interface thanks to the introduction of weighting functions $w_i(\mathbf{x})$. In this way, once we have introduced a set of weighting functions $w_i(\mathbf{x})$, denoting by Γ the fluid-structure interface, the equality relation between the fluid and the structure displacements on Γ could be expressed as follows:

$$\int_\Gamma w_i(\mathbf{x})\, \boldsymbol{\zeta}(\mathbf{x})\, d\gamma = \int_\Gamma w_i(\mathbf{x})\, \boldsymbol{\varsigma}(\mathbf{x})\, d\gamma. \tag{1.5}$$

At this point, by introducing the expression representing the approximation for each displacements fields, through, for example, a Ritz-Galerkin method, we obtain:

$$\boldsymbol{\zeta}(\mathbf{x}) = \sum_{k=1}^{n_f} N_{F,k}(\mathbf{x})\, \boldsymbol{\zeta}_k, \qquad \boldsymbol{\varsigma}(\mathbf{x}) = \sum_{j=1}^{n_s} N_{S,j}(\mathbf{x})\, \boldsymbol{\varsigma}_j, \tag{1.6}$$

where $N_{F,S}$ represent the interpolation basis functions for the fluid and the structure domains, respectively, while $\boldsymbol{\zeta}_k$ and $\boldsymbol{\varsigma}_j$ indicate the nodal displacement values for the fluid and the structure. In this way, we can firstly introduce (1.6) into (1.5), and, then, by choosing as weighting functions the ones used in (1.6), accordingly to the Galerkin method [94], we obtain:

$$\sum_{k=1}^{n_f} \left[\int_\Gamma N_{F,k}(\mathbf{x}) N_{\beta,i}(\mathbf{x})\, d\gamma \right] \boldsymbol{\zeta}_k = \sum_{j=1}^{n_s} \left[\int_\Gamma N_{S,j}(\mathbf{x}) N_{\beta,i}(\mathbf{x})\, d\gamma \right] \boldsymbol{\varsigma}_j, \quad \text{for } i = 1, \ldots, n_\beta \tag{1.7}$$

where β indicates either the fluid or the structural domain, such that $n_\beta \in \{n_s, n_f\}$. It is possible to achieve (1.3) by defining:

$$\mathbf{H}_{\beta F} \;=\; \sum_{k=1}^{n_f} \left[\int_{\Gamma} N_{F,k}(\mathbf{x}) N_{\beta,i}(\mathbf{x})\, d\gamma \right], \quad \text{for } i = 1, \ldots, n_\beta \qquad (1.8)$$

$$\mathbf{H}_{\beta S} \;=\; \sum_{j=1}^{n_s} \left[\int_{\Gamma} N_{S,j}(\mathbf{x}) N_{\beta,i}(\mathbf{x})\, d\gamma \right], \quad \text{for } i = 1, \ldots, n_\beta \qquad (1.9)$$

yielding:

$$\mathbf{H}_{\beta F}\, \boldsymbol{\zeta} = \mathbf{H}_{\beta S}\, \boldsymbol{\varsigma}. \qquad (1.10)$$

Finally, we can define the transformation matrix as it follows:

$$\boldsymbol{\zeta} = \underbrace{\mathbf{H}_{\beta F}^{-1}\, \mathbf{H}_{\beta S}}_{\mathbf{H}}\, \boldsymbol{\varsigma} \qquad (1.11)$$

A third class of strategies that have been employed in order to tackle the fluid-structure interface problem is based on the use of Radial Basis Functions [18, 20]. Expansions in Radial basis functions allow to approximate both multivariate functions and scattered data. They have been known, tested and analyzed for several years now and many interesting interpolation properties have been identified [121]. Generally, it is possible to define a radial basis function as a real-valued function whose values depends only on the distance from the origin, such that:

$$\phi(\mathbf{x}) = \tilde{\phi}(\|\mathbf{x}\|), \qquad (1.12)$$

or alternatively on the distance from another point $\mathbf{c}$, called center, that leads to:

$$\phi(\mathbf{x}, \mathbf{c}) = \tilde{\phi}(\|\mathbf{x} - \mathbf{c}\|), \qquad (1.13)$$

where the norm represents, usually, the Euclidean distance. Therefore, any function ϕ that satisfies the property $\phi(\mathbf{x}) = \tilde{\phi}(\|\mathbf{x}\|)$ is a radial function. Indeed, it is possible to classify RBFs in two different categories: the first made up of globally supported functions, the second of compactly supported functions. In Section 2.3 we will investigate the behavior of the local/global supported basis functions, and derive the mathematical formulation leading to the definition of the interpolation matrix $\mathbf{H}$ of (1.3), suited for the RBF strategy. Such a shape parametrization technique is characterized by a very high flexibility since it is possible to get it involved both in the fluid-structure interface and mesh motion problems. Furthermore, one of the greatest advantages of this method lies in its applicability

in almost any dimension (whence its versatility) because there are generally little restrictions on the way the data are prescribed. In literature it is possible to find a wide range of its applications, spanning many fields like numerical aeroelasticity [44, 93], high performance sail boat design [73, 74] and haemodynamics [81], involving the data-transfer across non-matching discretization schemes adopted for the structure and the fluid grids. In such cases, RBF strategy has demonstrated to represent an high-reliable methodology since it allows the spatial coupling between arbitrary schemes, adopted for both the fluid and structural domains, providing an high-accuracy of the achieved results. At the basis of the previously introduced background, the state of the art methods commonly adopted to tackle the transfer of data across non matching fluid-structure grids within FSI problems are:

(i) Nearest Neighbor Interpolation [115];

(ii) Weighted Residual Methods [25];

(iii) Radial Basis Functions [20].

1.2.2   Dynamic mesh handling

In this Section we focus our attention on the dynamic mesh handling within fluid-structure interaction problems. In order to solve the governing equations for both the fluid and structure on moving domains, accordingly to the ALE approach, we need a mesh motion routine: in fact, due to the multiphysics interaction between the fluid and the structure, during the FSI simulation the computational grids will move, resulting in geometrically deforming boundaries. The displacement of the interface boundary nodes is considered to be given as solution of the structural sub-problem. Moreover, in those cases involving time-varying boundaries, one of the most important aspects to deal with, concerns the requirement of keeping a high mesh quality: in fact, when elements or volumes of the computational grids suffer from excessive deformations, the solution of both the fluid and the structural sub-problems would get inaccurate, leading also to instabilities of the coupled FSI process. Therefore, the main requisites that must be respected when handling the mesh motion problem rely on the (i) preservation of the mesh validity (no negative Jacobian), keeping also (ii) a high mesh quality (in terms of elements orthogonality and skewness). In literature, a wide range of mesh deformation methods have been presented using different approaches to calculate the motion of the computational grids. When dealing with structured meshes, the Transfinite Interpolation technique [32] represents the most suitable method to be adopted: within this strategy the displacement of the interface boundary points is interpolated along grid lines through the entire computational mesh to find the displacements of all interior mesh points. Nevertheless, this strategy does not best suite those cases involving unstructured

meshes, employed to deal with complex geometries (where the use of unstructured meshes represents more often the unique choice to be adopted). One of the most popular mesh deformation strategy, applicable to both structured and unstructured meshes, is represented by the so called spring analogy [41]: it basically relies in the representation, by means of a linear/torsional spring, of the point-to-point connection of every two neighboring mesh points. This method has been proved to lack in robustness, in particular when dealing with arbitrarily unstructured meshes, since an high mesh quality is preserved only when setting specific spring stiffness values, in a very problem-dependent way. In the framework of mesh motion strategies for FSI, it is possible to highlight a class of techniques that involve the solution of a partial differential equation, defined on the fluid mesh region, with boundary conditions, expressed in terms of displacement, given on the fluid-structure interface. Within such strategies, the most common adopted techniques are represented by the Laplacian [56] and Solid Body Rotation Stress (SBRS)[39] methods. Concerning the first, its main idea relies on the possibility of assigning largest displacement values close to the moving boundaries while small ones at large distances. The Laplace equation to be solved is defined as it follows:

$$\nabla \cdot (\gamma \nabla \mathbf{u}) = 0, \tag{1.14}$$

being $\mathbf{u}$ the displacement of the grid nodes, and γ the coefficient of diffusivity, considered in order to control the magnitude of the nodal displacement. Namely, a common choice for the diffusivity coefficient reads as:

$$\gamma = \frac{1}{\|\mathbf{x}\|^\alpha}, \tag{1.15}$$

which expresses an inversely proportional relation between the diffusivity coefficient and the Euclidean norm of the distance from the deforming boundary, $\|\mathbf{x}\|$, raised to the α power. In order to maintain robustness and a smooth mesh motion, α is generally chosen to be equal to two guaranteeing a quadratically decreasing diffusion coefficient. Concerning SBRS, the problem to be solved to evaluate the mesh motion is governed by the linear elasticity equation:

$$\nabla \cdot \boldsymbol{\sigma} = \mathbf{f}, \tag{1.16}$$

where $\boldsymbol{\sigma}$ represents the stress tensor, while $\mathbf{f}$ the body forces term. By expressing (1.16) in terms of strain, thanks to the elastic constitutive relation, we get:

$$\nabla \cdot [\lambda Tr(\boldsymbol{\epsilon})\mathbf{I} + 2\mu\boldsymbol{\epsilon}] = \mathbf{f}, \tag{1.17}$$

being $Tr(\cdot)$ the trace operator, while λ and μ are the Lamé constants defined as

it follows:

$$\lambda = \frac{\nu E}{(1 + \nu)(1 - 2\nu)}, \qquad \mu = \frac{E}{2(1 + \nu)}, \qquad (1.18)$$

where E represents the Young modulus and ν the Poisson coefficient [39]. We can also write the kinematic law, between the strain and displacement fields:

$$\boldsymbol{\epsilon} = \frac{1}{2}(\nabla \mathbf{u} + \nabla \mathbf{u}^T + \nabla \mathbf{u}^T \cdot \nabla \mathbf{u}), \qquad (1.19)$$

where it is important to observe the presence of the term $\nabla \mathbf{u}^T \cdot \nabla \mathbf{u}$ that allows rigid rotations of the mesh. At this point, rearranging (1.19), (1.17) and (1.16) we obtain the final expression for the solid body rotation stress equation:

$$\nabla \cdot [\,\mu(\nabla \mathbf{u} + \nabla \mathbf{u}^T) - \lambda \nabla \cdot \mathbf{u}\,] = 0. \qquad (1.20)$$

In this context, $E > 0$ may be seen as the stiffness of the material, where large E indicates rigidity. Poisson's ratio ν represents a measure of how much the material shrinks in the lateral direction as it extends in the axial one. Such robust strategy leads to an high mesh quality although, like for the Laplacian-based method, a tuning operation for the coefficients λ and μ is necessary. Nevertheless, solving the Laplace or the SBRS equation will drive to sparse systems, such that standard iterative techniques, like the Pre-conditioned Conjugate Gradient (PCG) method, can be used to solve the problem.

Among the state of the art methods employed in order to handle the mesh motion problem for FSI, within an ALE approach, it is possible, finally, to highlight the RBF strategy. As introduced in Section 1.2.1, the application field of radial basis functions is very wide, since they have been used in computer graphics, geophysics, and in multiphysics simulations like fluid-structure interaction. Mesh deformation strategies based on RBF interpolation result in high quality meshes, as will be shown in Section 2.3, even in those cases involving large structure rotation angles [84]. Generally, the RBF interpolation procedure is used to transfer the known interface boundary node displacements to the interior fluid grid nodes: it is based on the adoption of a set of interpolation sites, or control points, whose displacement values are known, in order to get the ones related to the inner fluid mesh. In this way it is possible to adopt, as interpolation sites, all the nodes lying on interface boundary, since any restriction limits both the number and the locations of the control points. Therefore, one of the possible aspect to be studied, within this strategy, consists in automating and standardizing the choice of the interpolation sites, in terms of their number and locations. Summarizing, some of the strategies most commonly adopted to handle the mesh motion problem within FSI are:

(i) Transfinite Interpolation techniques [32];

(ii) Laplacian-based methods [56];

(iii) Solid Body Rotation Stress [39];

(iv) Radial Basis Functions [73].

At the basis of the previously introduced background we can also state the possibility of substituting the most common adopted Laplacian-based methods (for example in LifeV [49]), within the dynamic mesh handling, by RBF strategies: to this end, in this work we will investigate both RBF and Laplacian strategies to handle the mesh motion problem within many examples of FSI problems (as reported in Chapter 5).

1.3   Objectives and approaches

In many engineering applications the correct analysis of fluid-structure interaction problems is crucial. In this multiphysics framework we have different fields to be coupled that are represented not only by the fluid and the structure domains, but, also, by their computational grids, or meshes. In fact, dealing with an ALE approach, adopted in order to solve this complex phenomena, we need to be able to tackle the problem related to the preservation of high mesh quality in order to maintain a high accuracy of the flow solver. Furthermore, in this scenario, it is usual to adopt different discretization schemes for both the fluid and the structure: in this way the need for interface schemes arises, based on interpolation and/or projection algorithms. The main goal of this work consists in the comparison of different shape parametrization schemes involved in the framework of the FSI problems, and, in particular, to handle the dynamic mesh motion problem and also the possibility of transferring data across non-matching computational grids. Through the adoption of shape parametrization strategies we would firstly analyze their suitability for these contexts, and, at the basis of the achieved results, to suggest a hierarchical path to be followed when dealing with such problems. Our hierarchical approach could be seen as a result gained at the basis of the many test-cases solved in 1D, 2D and 3D frameworks, as it will be shown in Chapter 2. In detail, thanks to the comparison in terms of computational cost and accuracy of the achieved results, we aim to identify, among Free-Form Deformation, Radial Basis Functions and Inverse Distance Weighting, the most suitable application field for each of them in the framework of both the fluid-structure interface and mesh motion problems. Besides the implementations and the tests of these numerical strategies for FSI applications, from the analysis of the state of the art methods summarized in Section 1.2.1 and 1.2.2, we decided to improve some features that characterize shape parametrization

strategies studied. In particular we aim at introducing several ad hoc procedures able to perform a dimensionality reduction for the shape parametrization: in this context we will introduce novel techniques to solve the curse of dimensionality, especially for the RBF method, in terms of both the number of parameters to be employed to control the shape deformations and their locations. Moreover, at the basis of the possibility of dealing with FSI problems involving the multiphysics interaction between fluid flows and deformable free-bodies (not constrained), we decided to develop a numerical routine able to recover the rigid movements that will eventually characterize the structural dynamics. In fact, in those cases, the global structural displacement could be seen as made up of two different parts: a (small) deformative component and a (big) rigid one. While for the first we employ shape parametrization techniques, for the other we need to correctly identify the entity of rigid translation and/or rotations as output from the structural solver.

Summarizing, the main goals to be achieved by this work read as:

(i) the employment, in the framework of FSI problems, of several shape parametrization techniques in order to deal with:

  (a) fluid-structure interface problems;

  (b) mesh motion problems;

(ii) the evaluation and comparison, through several test-cases, of their performances in terms of computational costs and accuracy of the achievable results, in order to establish between them a hierarchical pattern to face the problems (a) and (b);

(iii) the development of numerical strategies in order to further reduce the dimensionality for the geometrical model described by shape parameterization techniques;

(iv) the identification of the rigid displacement components, translations and/or rotations, as output from the structural solver;

(v) to validate the strategies outlined within the study of 3D FSI unsteady problems, dealing with both internal and external fluid flows, considering, also, different Reynolds numbers;

In order to meet the objectives outlined for this work, we have decided to build a tool library, written in C++ code within an object-oriented framework, to be possibly linked to the different software employed to solve the set of partial differential equations that govern the fluid-structure interaction problem (that will be outlined in Chapter 4). In detail, to test the mesh-tool library developed within FSI problems, we considered an opensource library, namely LifeV [49]: it

would be adopted for the study of both an application in the haemodynamics framework, concerning, in detail, an internal fluid flows at low Reynolds number through a high-elastic structure, as well as an external fluid flow past a flexible obstacle within the field of large structural displacements. On the basis of the fully-flexible implementation of the numerical schemes developed, together with the high-fidelity Finite Element library adopted, it will be possible to test the proposed hierarchical strategy to show both its accuracy and its computational costs with respect to the results achieved.

Chapter 2

Shape parametrization techniques

This Chapter introduces some shape parametrization techniques used for treating both fluid-structure interface and mesh motion problems. In general shape representation strategies rely on mapping a reference domain, $\Omega \subset \mathbb{R}^n$, through suitable parametric maps $T(\mathbf{x}, \boldsymbol{\mu})$, $\mathbf{x} \in \Omega$, built by introducing a small set of control points, the so-called parameters $\boldsymbol{\mu} \in \mathbb{R}^n$, whose possible displacements induce the shape deformations. In this way, thanks to control points displacements, it would be possible to represent efficiently a family of admissible shape configurations. In this class of methods two strategies are represented by Free Form Deformation and Radial Basis Functions; we can also classify the latter with Inverse Distance Weighting techniques as methods belonging to multivariate interpolation strategies, which is an area of data fitting used to find the surface by providing an exact fit to a series of multidimensional data points. It is called multivariate since the data points are supposed to be sampled from a function of several variables.

(a) Target function. (b) RBF interpolant, Gaussian basis.
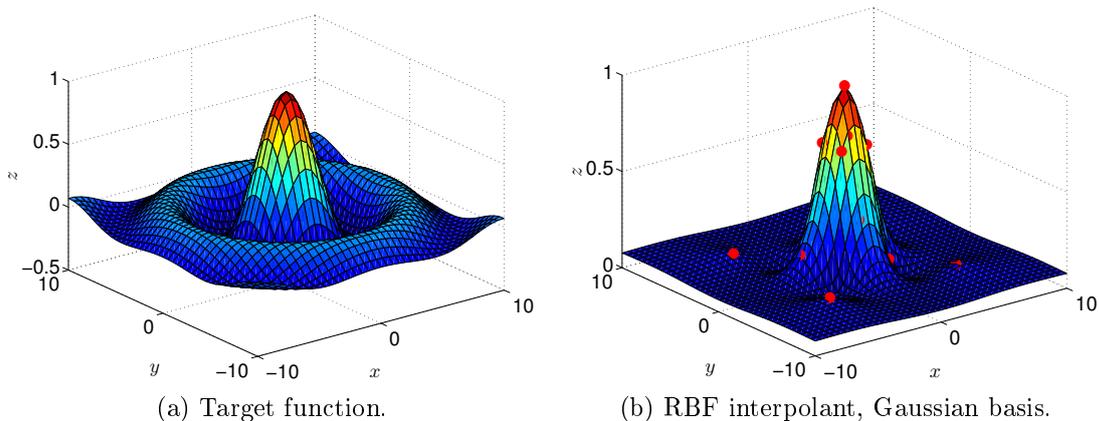
Figure 2.1: RBF interpolant, obtained by means of 13 interpolation sites (red bullets), applied to the Octave sombrero function [4].

By using RBF (as schematically shown in Figure 2.1), the function values at the unknown points is estimated as a sum, weighted by mean of radial basis functions, of the function values known at the scattered (or interpolation) points; with IDW the interpolating surface is obtained as a weighted average of the scattered points and the weight assigned to each one diminishes as the distance from the interpolation point to the scattered one increases.

In this Chapter, in order to face the fluid-structure interface and moving mesh problems, all the shape parametrization techniques proposed will be employed as illustrated in Figure 2.2: in particular, we are going to apply, in order to solve efficiently the problem of data transferring across non-matching grids, both IDW and RBF, while for the dynamic mesh handling we will also consider FFD techniques.
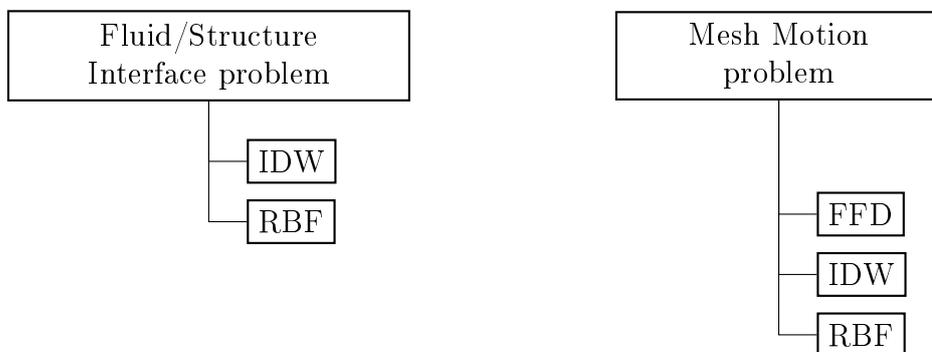
Figure 2.2: Application fields for shape parametrization techniques.

This Chapter has been organized as it follows: in Sections 2.1, 2.2 and 2.3 the shape parametrization strategies considered, as well as IDW, are introduced and detailed; in Section 2.4 we propose some applications of coupling schemes relying on multivariate interpolation methods for treating the transfer of data between non-matching grids, while in Section 2.5 we deeply analyze the mesh motion problem and we apply the numerical schemes introduced with two- and three-dimensional test-cases. Starting from these applications we will both compare methods through a computational costs analysis, as shown in Section 2.4 and 2.5, and outline a selective way for their use on the base of their properties.

2.1　Free Form Deformation

In the recent past the problem of shape parametrization of solid objects bounded by topologically complex surfaces has become of great importance and a significant issue in a few research fields, particularly in the framework of optimization design problems and within Reduced Order Modeling (ROM). In that way a

variety of techniques has been developed and, among them, Free Form Deformation: employed in the beginning [109] as an important tool for computer-assisted geometric design and animation, nowadays it is mostly adopted in several engineering application fields concerning optimal design. Indeed, within optimization sizing procedures, when it is necessary to iterate on several shape configurations to reach the optimal condition, thanks to the possibility of describing complex solid objects shapes and their deformations by a low dimensional space, FFD represents one of the best options to be possibly taken into account. In literature it is possible to find a wide range of applications of FFD techniques used to deal with optimization problems, and it has been adopted, particularly, within aerodynamics [9], haemodynamics [13] and high performance boat design [74]. With respect to the previously introduced background it must be remarked the novelty, introduced by this work, of applying FFD to mesh motion problems for FSI. As shown in Figure 2.3, FFD technique does not operate directly on the geometrical properties themself, but it leads to local and/or global shape deformations thanks to the use of a parametric map $T_{FFD}(\cdot, \boldsymbol{\mu})$ on a reference and parameter-independent domain configuration Ω by introducing a small set of control points who play the role of parameters, $\boldsymbol{\mu}$, to be perturbed (displaced) in order to obtain the shape deformation.

(a) $\qquad\qquad\qquad\qquad\qquad\qquad$ (b)
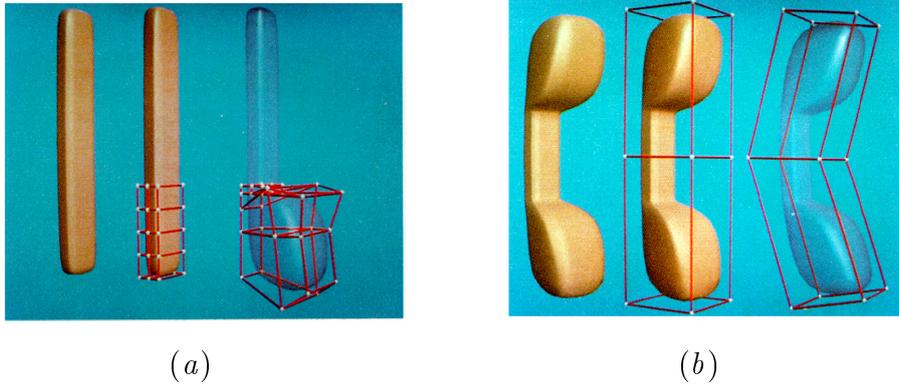
Figure 2.3: Example of local (a) and global (b) shape deformations obtained using FFD technique [109].

One of the key-points of this method relies in the possibility of parametrizing arbitrary solid object shapes and their own meshes: by introducing a perturbation to the control points positions we can simply get a family of admissible shape deformations and, consequently, of mesh deformations.

Let us now consider the FFD mathematical formulation for the three-dimensional case: in this way, we start considering a physical reference domain Ω of coordinates $\mathbf{x} = (x_1, x_2, x_3)$, and D, such that $\Omega \subset D$, with D representing a parallelepiped that could be considered as a fasten box containing our domain. Mapping D through an affine, differentiable and invertible map $\Psi(\mathbf{x})$, we transform

it to the unit cube domain $\hat{D}$ of coordinates $\hat{\mathbf{x}} = (s, t, u)$, with $\Psi(D) = (0, 1) \times (0, 1) \times (0, 1)$, and we introduce a user-defined lattice of $(L+1) \times (M+1) \times (N+1)$ control points $\mathbf{P}^0_{l,m,n}$:

$$\mathbf{P}^0_{l,m,n} = \{l/L, \, m/M, \, n/N\}^T, \quad \text{with} \quad \begin{aligned} l &= 0, \dots, L \\ m &= 0, \dots, M \\ n &= 0, \dots, N \end{aligned} \qquad (2.1)$$

where L, M and N represent the number of control points in the x_1, x_2 and x_3 directions, respectively.

fixed reference domain

deformed parametric domain
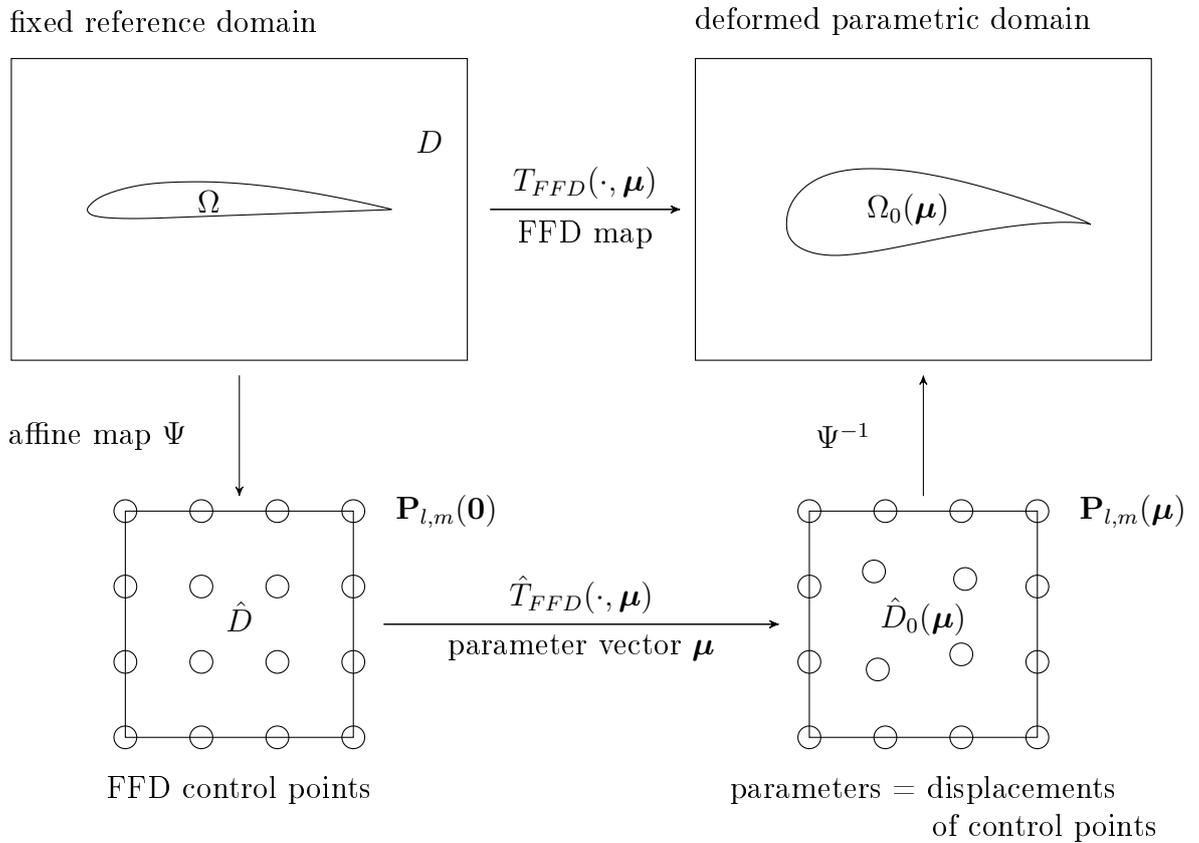
Figure 2.4: Schematic view of the FFD shape deformation procedure: we remark that, although in this example we do not displace the control points lying on the external boundaries, it is possible in general to displace them to get the deformed parametric domain.

In order to describe possible deformations of the solid object embedded within D we introduce a parameter vector $\boldsymbol{\mu}_{l,m,n}$ of dimensions $((L + 1) \times (M + 1) \times (N + 1)) \times 3$ representing control points displacements in s, t and u directions, as schematically shown in Figure 2.5:
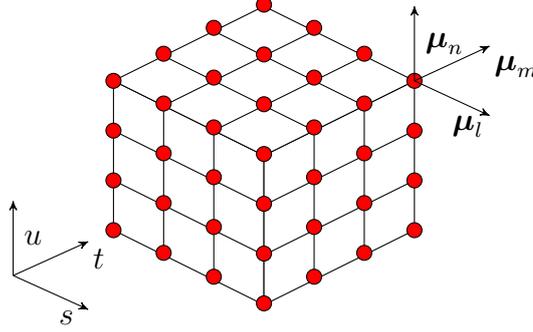
Figure 2.5: Schematic visualization of unperturbed control points (red bullets) $\mathbf{P}^0_{l,m,n}$ and their admissible displacements.

Once we have introduced the parameters vector $\boldsymbol{\mu} \in \mathbb{R}^{N_C \times 3}$, where N_C represents the number of the parameters, we denote the perturbed control points configurations as:

$$\mathbf{P}_{l,m,n}(\boldsymbol{\mu}_{l,m,n}) = \mathbf{P}^0_{l,m,n} + \boldsymbol{\mu}_{l,m,n};$$
(2.2)

thus, it is possible to define the parametric map $\hat{T}_{FFD} : \hat{D}_0 \times \mathbb{R}^{N_C \times 3} \mapsto \hat{D}$ as follows:

$$\hat{T}_{FFD}(\hat{\mathbf{x}}, \boldsymbol{\mu}) = \left(\sum_{l=0}^{L} \sum_{m=0}^{M} \sum_{n=0}^{N} b^{L,M,N}_{l,m,n}(\hat{\mathbf{x}}) \mathbf{P}_{l,m,n}(\boldsymbol{\mu}_{l,m,n}) \right),$$
(2.3)

where

$$
\begin{aligned}
b^{L,M,N}_{l,m,n}(\hat{\mathbf{x}}) &= b^L_l(s) b^M_m(t) b^N_n(u) \\
&= \binom{L}{l} \binom{M}{m} \binom{N}{n} (1-s)^{L-l} s^l \, (1-t)^{M-m} t^m \, (1-u)^{N-n} u^n
\end{aligned}
$$
(2.4)

represents the tensor product of unidimensional Bernstein basis polynomials defined on $\hat{D}_0$:

$$
\begin{aligned}
b^L_l(s) &= \binom{L}{l} (1-s)^{L-l} s^l, \\
b^M_m(t) &= \binom{M}{m} (1-t)^{M-m} t^m, \\
b^N_n(u) &= \binom{N}{n} (1-u)^{N-n} u^n.
\end{aligned}
$$
(2.5)

Finally, in order to find the free form deformation map $T_{FFD}(\mathbf{x}, \boldsymbol{\mu})$, we have to compose Ψ and $\hat{T}_{FFD}$ as schematically depicted in Figure 2.4 and formalized below:

$$T_{FFD}(\mathbf{x}, \boldsymbol{\mu}) = (\Psi^{-1} \circ \hat{T}_{FFD} \circ \Psi)(\mathbf{x}, \boldsymbol{\mu}) \qquad \text{(FFD map)}$$

From a computational point of view, the whole process represented in Figure 2.4 could be split in two different steps: an *offline* (expensive) stage, to be completed before running the FSI simulation, that consists in building and storing the FFD map on the initial reference domain, and the *online* (cheaper) one, involving real-time evaluations of the map on the new domain configuration obtained as deformation of the initial one during the simulation process.

It must be remarked that, although FFD is characterized by high flexibility and easiness of handling, it suffers from some limitations. The first lies in the fact that design variables may have no physical significance: they are defined in a parametric domain that could not be expressed into a particular unit of measurement by definition. Moreover FFD is not interpolatory and, therefore, in order to solve (MMP), parameter values could not be found directly, as will be fully detailed in Section 2.1.1. Finally, all the control points are restricted to lie on a regular lattice and, in that way, local refinements could not be performed.

2.1.1   FFD application for mesh motion problems

We focus now on the application of the FFD technique in the framework of moving mesh problems for FSI: as shown schematically in Figure 2.2, in this work, that method will get involved only for that purpose. When dealing with an FSI simulation, once we compute the solution (in terms of stresses and displacements) of the structural sub-problem, we need to update the fluid domain configuration accordingly to the structural displacement. Therefore the objective is to avoid (or minimize) the presence of localized gap and/or overlap zones that could arise between the fluid and structural domain interfaces when updating the domain configurations. In order to tackle this problem, by employing FFD techniques, two first clear difficulties should be underlined which rely in (*i*) the non-interpolatory nature of this method and in (*ii*) the lack of physical links between control points perturbation and domain deformations. We can formalize the mesh motion problem as it follows: let's denote as $\hat{\Omega}^S \subset \mathbb{R}^3$ and as $\hat{\Omega}^F \subset \mathbb{R}^3$ the initial reference domain configurations for the structure and the fluid domains, respectively, and we consider their discretizations (or computational meshes) $\hat{\Omega}_h^S \in \mathbb{R}^{n_s \times 3}$ and $\hat{\Omega}_h^F \in \mathbb{R}^{n_f \times 3}$, being n_s and n_f the chosen numbers for structural and fluid grid nodes. Moreover, we assume hereafter matching fluid-structure grids at the interface. As solution of the structural sub-problem

initial domains configuration   deformed domains
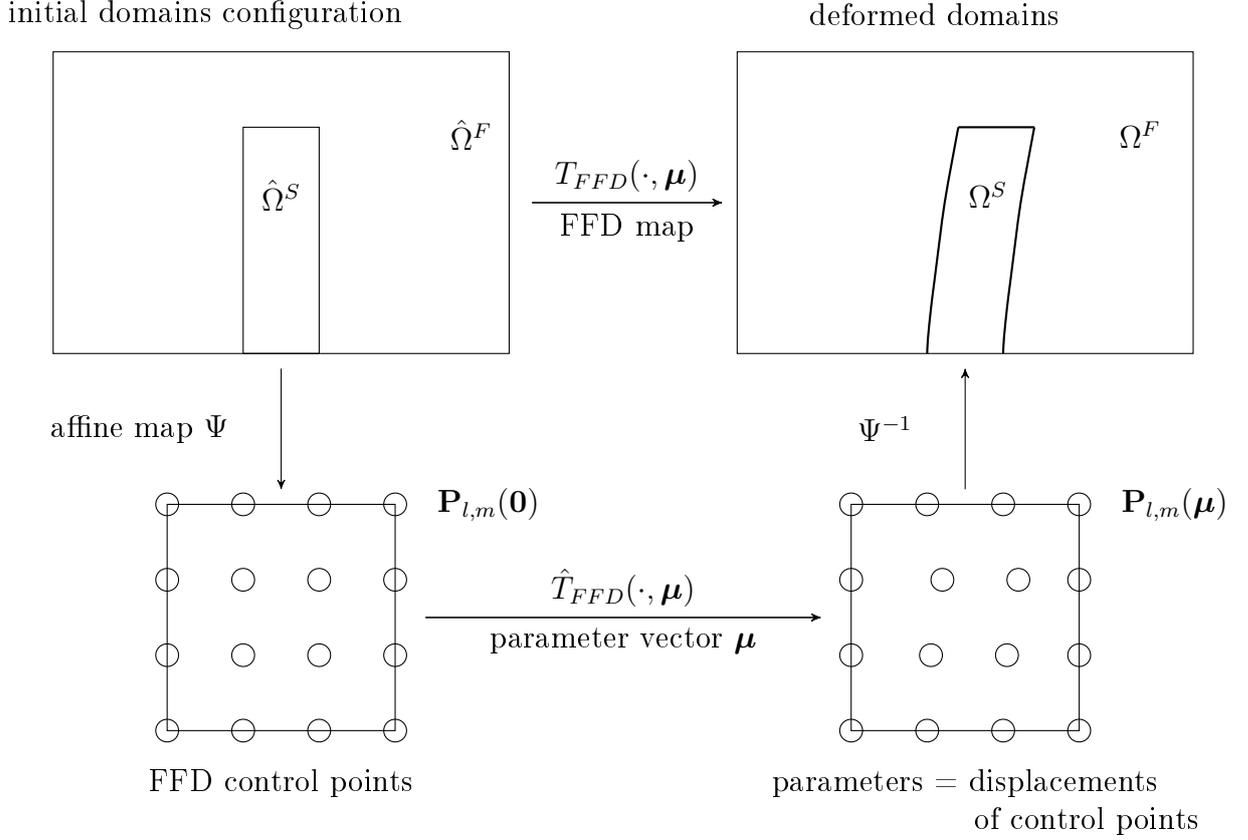
Figure 2.6: FFD shape deformation procedure applied to the mesh motion problem: with Ω^F and Ω^S we indicate the deformed fluid and structural domains, respectively.

we get the nodal displacement values $\hat{\mathbf{d}}_S$, such that $\hat{\mathbf{d}}_S : \hat{\Omega}_h^S \to \Omega_h^S$, where Ω_h^S indicates the final deformed structural mesh configuration. By introducing Γ^S and Γ^F, two linear selection operators which allow to extract from the whole list of domain nodes those who lie on the interfaces, the mesh motion problem consists in finding the fluid nodal displacement values $\hat{\mathbf{d}}_F$, leading to the deformed fluid configuration

$$\hat{\mathbf{d}}_F : \hat{\Omega}_h^F \to \Omega_h^F, \tag{2.6}$$

that guarantees no-compenetrations across interfaces between fluid and structure (to avoid the situation depicted in Figure 2.7):

$$\Omega_h^F \cap \Omega_h^S = \emptyset. \tag{2.7}$$

Thus, when the goal is to move fluid domain in order to fit the updated structural configuration at the interface, necessarily an optimization algorithm must be employed. In detail, since to move the fluid grid nodes we need to perturb the
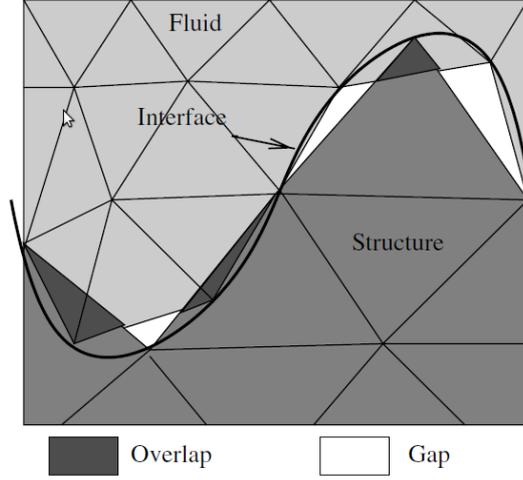
Figure 2.7: Presence of gaps and overlaps across interfaces to be avoided during the mesh motion process [33].

parameters vector $\boldsymbol{\mu}$, we can consider a cost functional:

$$\mathcal{J}(\boldsymbol{\mu}) = \parallel \Gamma^S(\Omega_h^S) - \Gamma^F(\Omega_h^F(\boldsymbol{\mu})) \parallel_{L_2}, \qquad (2.8)$$

that evaluates the L_2 norm of the difference between the structural and the fluid grid nodes positions over the interface. Thanks to the definition of $\mathcal{J}(\boldsymbol{\mu})$ we will be able to address the fluid mesh configuration matching the structural one at the interface by minimizing $\mathcal{J}(\boldsymbol{\mu})$ (as fully detailed in Algorithm 2.1). In this way, on the basis of the considerations previously introduced, in order to solve the mesh motion problem by employing FFD techniques we can proceed as it follows:

1. *offline* stage:

   (i) generate the fluid $\hat{\Omega}_h^F(\mathbf{x})$ and structure $\hat{\Omega}_h^S(\mathbf{x})$ meshes;

   (ii) map $\hat{\Omega}_h^F(\mathbf{x})$ to the unit cube domain $D(\hat{\mathbf{x}})$ through Ψ (affine map operator):

   $$\Psi(\Omega_0^F(\mathbf{x})) = (0,1) \times (0,1) \times (0,1) = D(\hat{\mathbf{x}}); \qquad (2.9)$$

   (iii) introduce, in the new parametric domain $D(\hat{\mathbf{x}})$, a lattice of $(L+1) \times (M+1) \times (N+1)$ control points $\mathbf{P}_{l,m,n}^0$;

   (iv) *compute* and *store* the FFD map $T_{FFD}(\mathbf{x}, \boldsymbol{\mu})$;

2. *online* stage, for each FSI time iteration t^n, with $n > 0$:

   (i) solve the fluid sub-problem, compute pressure and velocity fields;

(ii) transfer the computed fluid solution to the structural solver to update the boundary conditions;

(iii) solve the structural sub-problem to get $\hat{\mathbf{d}}_S$, that defines the new structural configuration $\Omega_h^S(\mathbf{x})$;

(iv) *evaluate* the stored FFD map on $\hat{\Omega}_h^F(\mathbf{x})$;

(v) define the parameters vector $\boldsymbol{\mu} \in \mathcal{D} = [\mu_{min}, \mu_{max}]^{N_c}$, being $\mathcal{D}$ the space of the admissible N_c parameters values;

(vi) initialize $\boldsymbol{\mu}$ to zero values;

(vii) compute the new fluid mesh configuration $\Omega_h^F(\mathbf{x})$ by solving the following optimization problem:

$$\min_{\boldsymbol{\mu} \in \mathcal{D}} \ \mathcal{J}(\boldsymbol{\mu}) = \min_{\boldsymbol{\mu} \in \mathcal{D}} \ \| \Gamma^S(\Omega_h^S) - \Gamma^F(\tilde{\Omega}_h^F(\boldsymbol{\mu})) \|_{L_2} . \tag{2.10}$$

With respect to equation (2.10), we indicated with $\tilde{\Omega}_h^F(\mathbf{x}, \boldsymbol{\mu})$ the fluid mesh configuration modified by mean of control points perturbations, such that for zero values of $\boldsymbol{\mu}$ we have $\tilde{\Omega}_h^F(\mathbf{x}, \boldsymbol{\mu}) = \hat{\Omega}_h^F(\mathbf{x}, \boldsymbol{\mu})$ while, when $\mathcal{J}(\boldsymbol{\mu})$ reaches its minimum value, $\tilde{\Omega}_h^F(\mathbf{x}, \boldsymbol{\mu}) = \Omega_h^F(\mathbf{x}, \boldsymbol{\mu})$.

We can now describe the optimization algorithm involved for the solution of problem (2.10). In order to find a set of suitable control point displacements able to update the fluid mesh configuration leading to no-compenetrations between domains across the interface, we adopted a *greedy* algorithm [97]. In particular, within our context, at each stage we firstly evaluate the cost functional $\mathcal{J}(\boldsymbol{\mu})$; then, in the hypothesis of dealing with conformal meshes at the FS interface, we evaluate the difference in the Euclidean norm between $\Gamma^S(\Omega_h^S)$ and $\Gamma^F(\tilde{\Omega}_h^F(\boldsymbol{\mu}))$. In detail, we firstly compute:

$$e_{x,k} = \Gamma^S(\Omega_{h\,(x,k)}^S) - \Gamma^F(\tilde{\Omega}_{h\,(x,k)}^F) \quad \text{for} \quad k = 1, \ldots, n_\Gamma, \tag{2.11}$$

$$e_{y,k} = \Gamma^S(\Omega_{h\,(y,k)}^S) - \Gamma^F(\tilde{\Omega}_{h\,(y,k)}^F) \quad \text{for} \quad k = 1, \ldots, n_\Gamma, \tag{2.12}$$

$$e_{z,k} = \Gamma^S(\Omega_{h\,(z,k)}^S) - \Gamma^F(\tilde{\Omega}_{h\,(z,k)}^F) \quad \text{for} \quad k = 1, \ldots, n_\Gamma, \tag{2.13}$$

being n_Γ the number of nodes lying on the FS interface and $\{\Omega_{h\,(x,k)}^{S,F}, \ \Omega_{h\,(y,k)}^{S,F}, \ \Omega_{h\,(z,k)}^{S,F}\}$ the x, y and z coordinates of the k-th fluid and structural grid node. Moreover, with $e_{l,k}$ we indicate the difference at the interface grid node k in the l direction. At this point by evaluating $\mathbf{e}(\mathbf{x})$, whose node-wise expression reads:

$$e_k = \sqrt{e_{x,k}^2 + e_{y,k}^2 + e_{z,k}^2} \quad \text{for} \quad k = 1, \ldots, n_\Gamma, \tag{2.14}$$

we further look for the node of coordinates $\tilde{\mathbf{x}} = (\tilde{x}, \tilde{y}, \tilde{z})$ corresponding to the

maximum value of $\mathbf{e}(\mathbf{x})$, such that $\max\{e_k\}_{k=1}^{n_\Gamma} = \tilde{e}$ for $k = \tilde{k}$, in order to perturb its closest control point of a priori fixed quantity $d\mu$ along the direction:

$$
\begin{cases}
x, & \text{if} \quad |e_{x,\tilde{k}}| > |e_{y,\tilde{k}}| \quad \text{and} \quad |e_{x,\tilde{k}}| > |e_{z,\tilde{k}}|; \\
y, & \text{if} \quad |e_{y,\tilde{k}}| > |e_{x,\tilde{k}}| \quad \text{and} \quad |e_{y,\tilde{k}}| > |e_{z,\tilde{k}}|; \\
z, & \text{if} \quad |e_{z,\tilde{k}}| > |e_{x,\tilde{k}}| \quad \text{and} \quad |e_{z,\tilde{k}}| > |e_{y,\tilde{k}}|.
\end{cases}
\tag{2.15}
$$

This methodology lies on the idea of approximating a global minimum by computing local ones. Algorithm 2.1 shows schematically all the steps involved to solve the problem 2.10.

Algorithm 2.1 Greedy algorithm used to solve problem (2.10)

Require: initialize $\boldsymbol{\mu} = 0$, $\hat{\mathbf{d}}_F = 0$
    Let $n = 0 \rightarrow \tilde{\Omega}_{h,n}^F = \hat{\Omega}_{h,n}^F$
    repeat
        for $k = 1, \ldots, n_\Gamma \rightarrow$ compute $e_{x,k}$, $e_{y,k}$, $e_{z,k}$ and e_k
        Find $\tilde{\mathbf{x}}$ s.t. $\displaystyle\max_{\mathbf{x} \in \Gamma^F(\tilde{\Omega}_h^F) \cap \Gamma^S(\Omega_h^S)} \parallel \mathbf{e}(\mathbf{x}) \parallel$
        Evaluate the direction x, y or z where to perturb the control point
        Move nearest control point to $\tilde{\mathbf{x}}$: $\mu_{n+1} = \mu_n + d\mu$
        Evaluate $\hat{\mathbf{d}}_{n+1}^F$ through FFD map $T_{FFD}(\mathbf{x}, \boldsymbol{\mu}_{n+1})$
        Update the fluid configuration: $\hat{\mathbf{d}}_{n+1}^F(\tilde{\Omega}_{h,n}^F(\mathbf{x}, \boldsymbol{\mu})) \rightarrow \tilde{\Omega}_{h,n+1}^F(\mathbf{x}, \boldsymbol{\mu})$
        Evaluate the cost functional $\mathcal{J} = \parallel \Gamma^S(\Omega_h^S(\mathbf{x})) - \Gamma^F(\tilde{\Omega}_{h,n+1}^F(\mathbf{x}, \boldsymbol{\mu})) \parallel_{L_2}$
        $n \rightarrow n + 1$
    until stopping criteria $\mathcal{J} \leq TOL$ is met.
    Set $\Omega_h^F(\mathbf{x}) = \tilde{\Omega}_{h,n}^F(\mathbf{x}, \boldsymbol{\mu})$

As stated before, by acting in this way, the convergence to a global minimum is not guaranteed: moreover, within applications involving FFD techniques, it must be underlined that, generally, the optimal solution may also not exist. In fact, when our goal is to describe with the FFD map a local displacement due to, for example, elastic localized deformations the behavior of the methodology proposed could fail, if considering a small number of parameters (control points). In that case, we can classify our system as not observable resulting in a non-existing optimal solution to be reached. In that way, when dealing with local displacements to be described, we should consider a great number of control points in order to guarantee the observability and the controllability of our process. Since now we remark that the choice we have made, related to the displacement of one control point per iteration, although it would decelerate the process leading to convergence, it represents the unique possible way to be adopted when the structural displacement to be recovered is localized (e.g. see example illustrated in Figure 2.9).

In order to clarify the mesh motion procedure proposed, now two examples of two- and three-dimensional test-cases are shown. For both of them the goal consists in updating the fluid mesh configuration to match the simulated structural deformation: in particular, as illustrated in Figure 2.9, for the 2D case we will consider a localized deformation while, for the 3D, a global one. In that way it will be possible to analyze the achievable mesh quality and computational costs involved. Moreover, with them, we will clarify also the idea of the observability of the systems to be studied in relation to different control point settings that will be adopted for the simulations. In our first example we deal with a rectangular (green) obstacle, representing the structure, inside an outer (blue) rectangle, representing the fluid domain. We will describe, employing the FFD technique, the imposed structural deformed configuration (Ω^S) depicted in Figure 2.9. For this example we considered different sets of control points in order to show on one hand the achievable mesh quality, and on the other both the convergence behavior of the greedy algorithm adopted and the computational cost related to the different numbers of parameters. Concerning the measure of the mesh quality adopted, we will consider the Jacobian measure, which calculates the partial derivatives of the quad-element shape functions with respect to the Cartesian coordinate system, scaled within the range [0,1] [6].
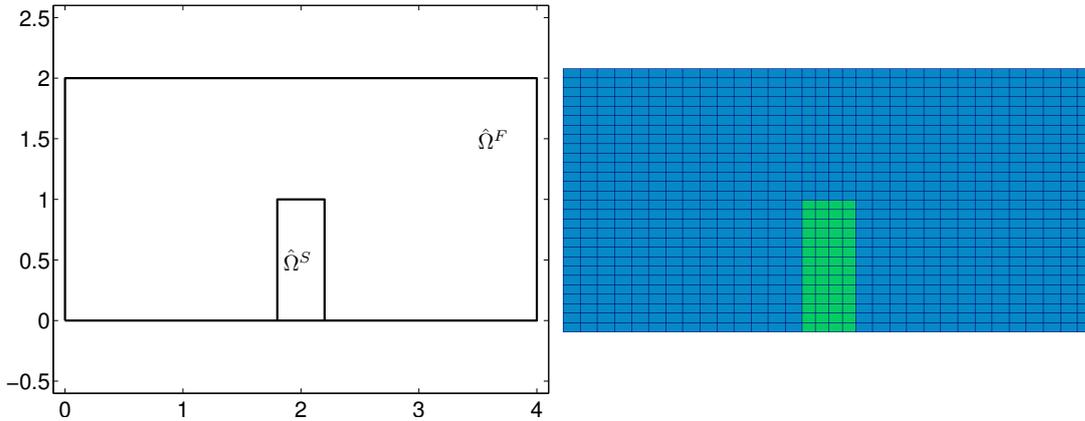
Figure 2.8: Initial fluid-structure domain configurations (left) and their discretization (right) for the 2D example considered.

Figure 2.10 illustrates some results obtained for this test-case and Table 2.1 shows the computational costs involved. From these results is possible to observe how high is the sensitivity of the FFD technique to the number of control points: only if we consider a large number of parameters, satisfactory results could be achieved. One of the drawbacks of this strategy lies in the positioning of control points: in this case it would be enough to locate few control points (by creating a so-called "patch") around the structure domain, but, in order to place them inside a regular lattice defined on the whole domain, this number necessarily increases.
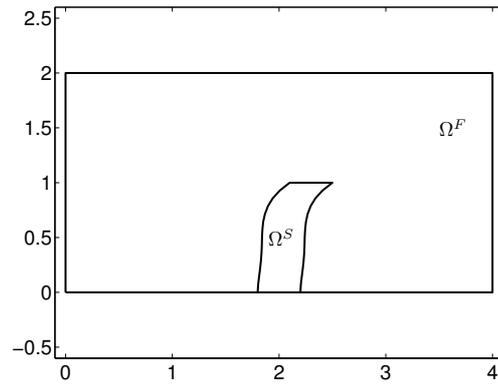
Figure 2.9: Visualization of both the final deformed fluid and structure domains; the deformed structural configuration was obtained by means of FFD technique by mapping only $\hat{\Omega}^S$.
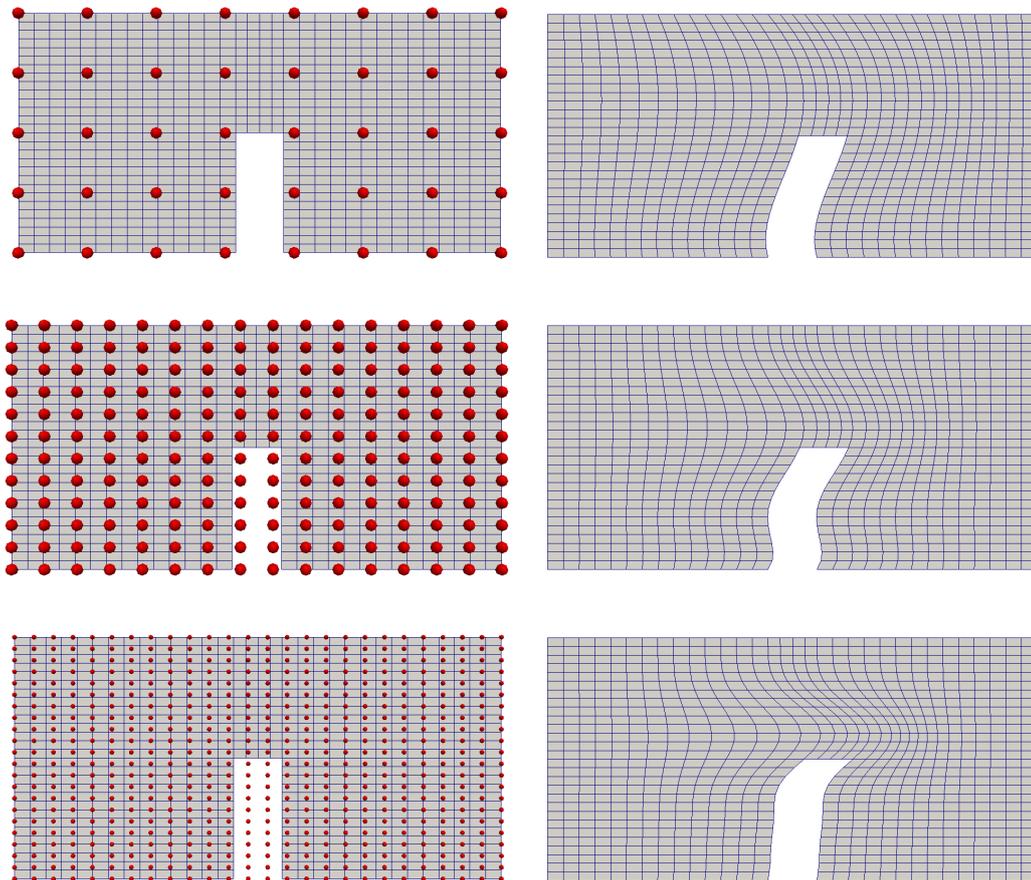
Figure 2.10: Visualization of the different simulation settings considered (left) in terms of control points adopted, red bullets, and corresponding fluid meshes updated (right).

We could also observe that, owing to the non-interpolatory nature of the method, also with a number of control points of the order of fluid mesh nodes ($n_f = 915$), it would be never achieved the condition of perfect matching between fluid and structural domains at the interface in the final deformed configuration, as shown in Figure 2.11. With respect to Figure 2.12, the contour plot of the mesh quality measure considered are illustrated for all the three different simulations settings: it is possible to appreciate how the scaled Jacobian assumes values within the range [~ 0.7, 1], meaning high mesh qualities. The quad-elements that compose the fluid mesh, for all the three cases considered, do not exhibit significant distortions in their deformed configuration, leading to high quality measures: in fact, from this point of view, we observe how FFD techniques lead to a smooth grid deformation that, in detail, involves mostly that region of the fluid domain closest to the deformed structural one. With respect to Figure 2.11 it is possible to notice that, once we have updated the fluid configuration, localized gaps between the fluid-structure interfaces are still present.
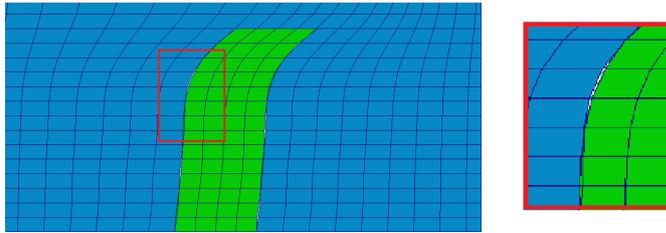
Figure 2.11: Zoom visualization of the non-matching fluid-structure interfaces condition (simulation performed using 676 control points).

This is due to the fact that FFD, generally, should be employed to describe global displacements instead of local ones. In Table 2.1 an analysis of the computational costs involved for this numerical simulation are reported, while Figure 2.12a shows the cost functional $\mathcal{J}(\boldsymbol{\mu})$ behavior versus the number of iterations of the greedy algorithm adopted for all the different simulation settings.

Control points number	$\mathcal{J}$ reduction [%]	CPU time per iteration [s]	Iterations
25	59.3%	0.01	212
126	87.1%	0.013	4000
234	89.83 %	0.02	800
456	94.52 %	0.03	1500
588	96.7 %	0.05	2000

Table 2.1: Visualization of both the FFD computational costs and the reduction rate of the cost functional with respect to its initial value.

We still underline the impossibility of the method proposed to achieve an optimal result when considering a small number of control points. In fact, even if the greedy algorithm is able to find a solution, which is the best related to the insufficient number of parameters used to control the shape deformation, the quality of the solution may not be so good. In order to find satisfactory solutions a greater number of control points should be adopted. Particular attention should also be dedicated to the number of iterations needed by the greedy algorithm to reach the convergence: by increasing the number of control points we also increase the space of the admissible parameters displacements to be activated by the algorithm, thus requiring more iterations of the greedy algorithm, as shown in Figure 2.12a.
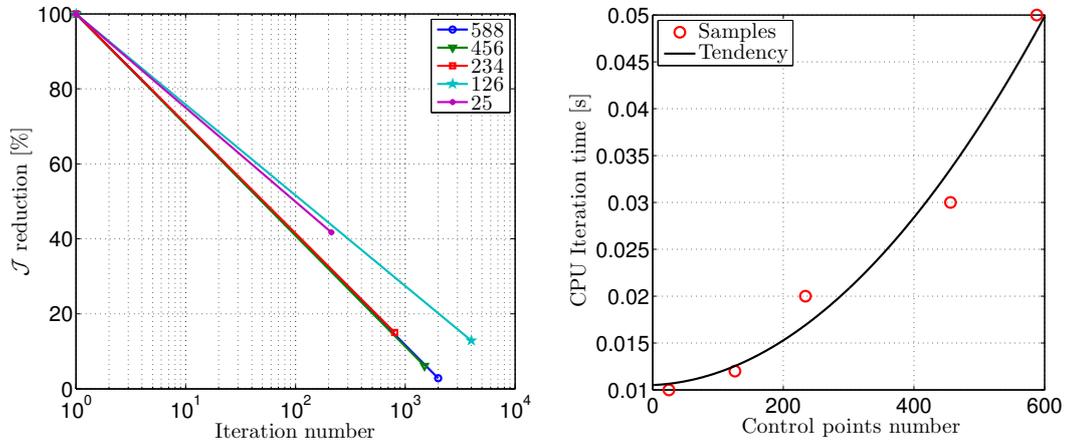
(a) Convergence of the Greedy algorithm for different numbers of control points.

(b) Visualization of the tendency of the computational costs.

A final consideration, concerning the 2D example: greedy algorithm convergence slope is independent of control points number after a reasonable number of control points activated; moreover, we state the parabolic rise of CPU computational time, needed per iteration, when increasing the number of parameters.
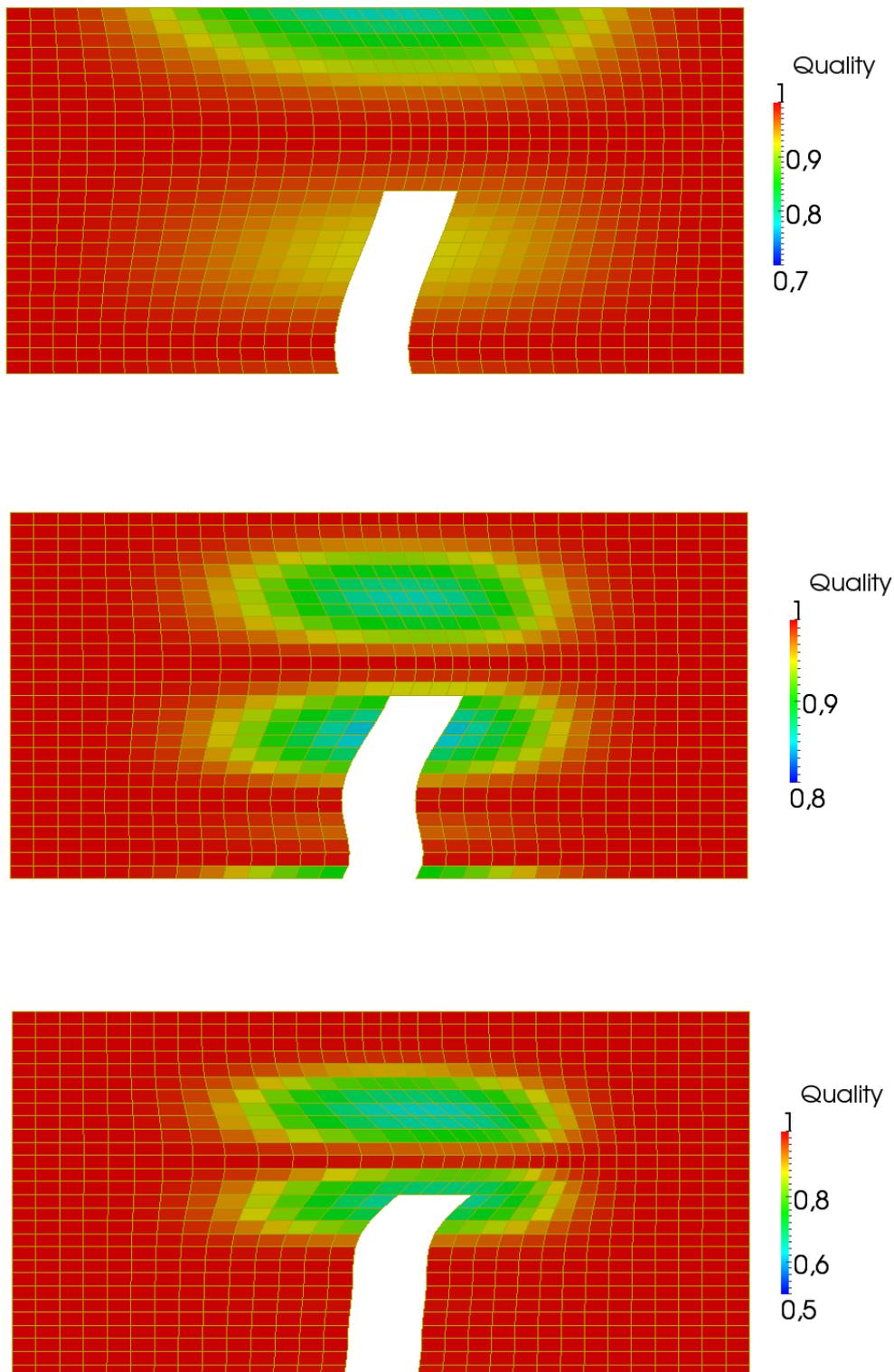
Figure 2.12: Mesh quality achieved for the 2D example. The computed quad-elements Jacobian shows conservation of the grids high-quality for all the simulation settings.

In the next example, as shown in Figure 2.13, we deal with a 3D test-case (representing a quite classical configuration for an internal fluid flow). As noticed within the previous example, FFD technique lacks in describing localized shape deformations, therefore, the aim of this test is to analyze and test FFD behavior into a three-dimensional framework and, also, to highlight its suitability to manage global shape deformations.
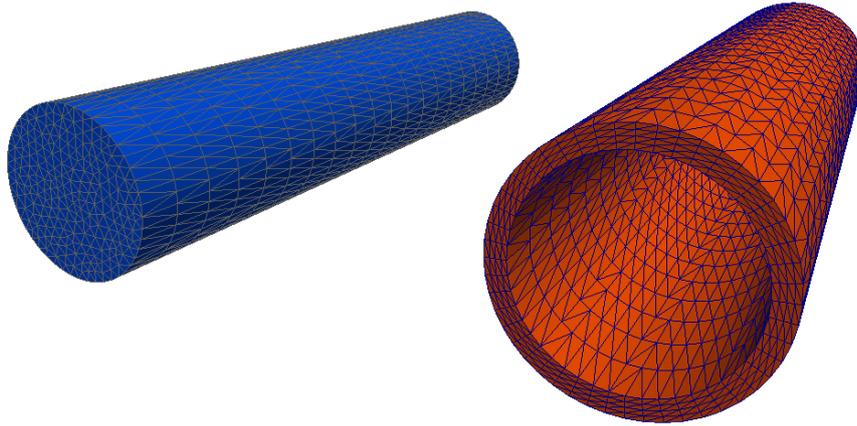
Figure 2.13: Fluid (blue) and structure (red) discretized domains considered for the 3D example.

As previously done, we impose a structural deformation, illustrated in Figure 2.14, by displacing structural grid nodes from their initial positions: such a deformed configuration is assumed to be global, so that it involves the whole structural domain. This deformed structural configuration has to be considered, also, as a target shape to be fitted at the interface by the new configuration of the fluid domain.
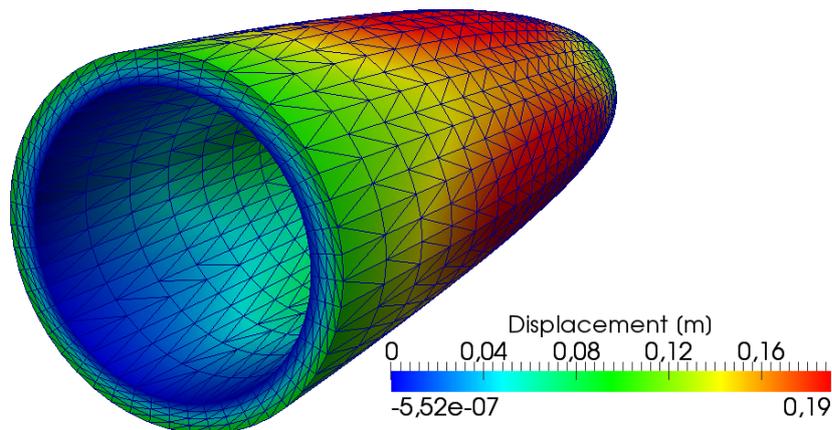
Figure 2.14: Structural deformation simulated.

In order to study the properties of the method, for this three-dimensional test we considered different control points setting characterized by an increasing refinement in all the three directions. Figure 2.16 illustrates one of the configuration adopted: in particular, here we considered $L = 3$ control points in the x direction, $M = 3$ for y and $N = 6$ for z, respectively, resulting in a total number of 54 parameters.

(a): Front view.   (b): Lateral view.
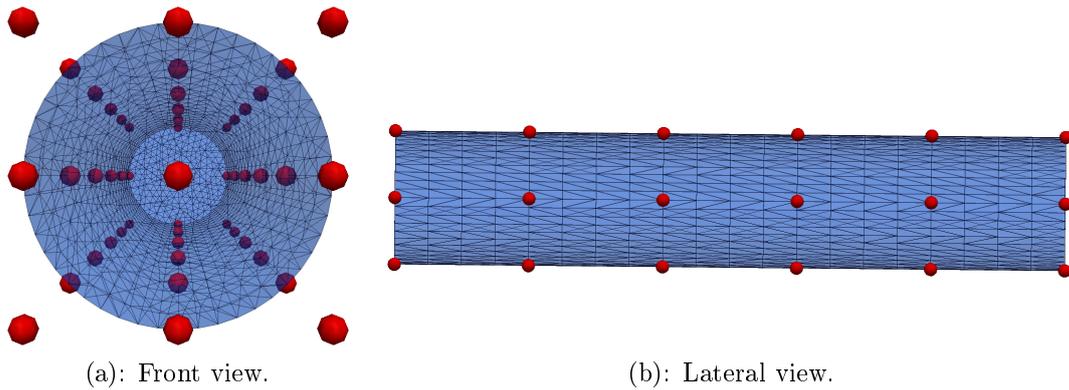
Figure 2.15: Visualization of one control points setting adopted for the 3D example analysis.
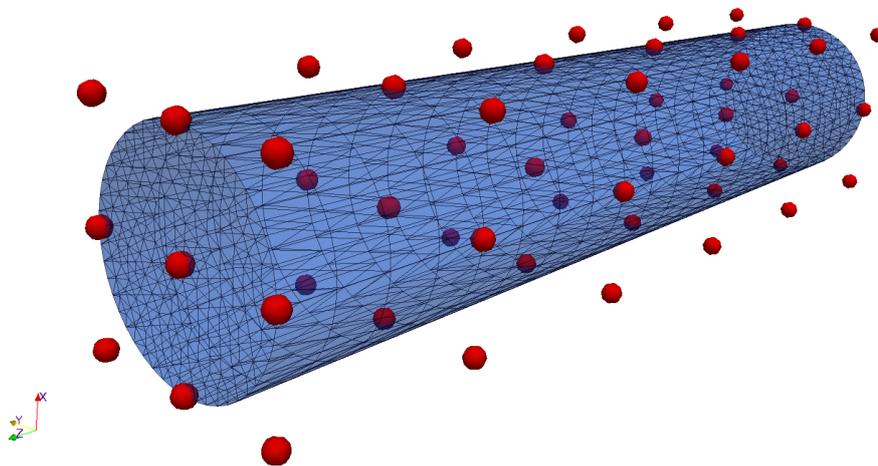
Figure 2.16: Isometric view of control points (red bullets) and fluid mesh (blue).

In Table 2.2 the main results obtained employing FFD technique in order to update the fluid mesh configuration are reported. We expect to obtain a better convergence, in terms of the number of iterations needed to reach acceptable

results, since the final shape to be recovered does not involve localized deformations.

Control points number	$\mathcal{J}$ reduction [%]	CPU time per iteration [s]	Iterations
9	59.3 %	0.01	8
125	84.7 %	0.04	10
490	87.45 %	0.20	49
972	97.55 %	0.51	362
1694	99.2 %	0.95	584

Table 2.2: FFD computational costs for the 3D example.

As the number of control points increases as better solution can be achieved: in particular when considering 1694 control points (with a number of fluid grid nodes $n_f = 4998$) we could get the best result to the detriment of computational costs. As expected, here, dealing with global shape deformations, FFD technique shows better results than the ones gained within the 2D case: satisfactory performances are reached by means of a smaller number of iterations to be done to reach convergence. Moreover, when we consider 3D configurations, as clearly shown in the Table above, the iteration time raises significantly. In Figure 2.17a the behavior of the functional cost $\mathcal{J}$ versus iteration number is depicted, while in Figure 2.17b we evaluate the computational costs, in terms of iteration time, increasing the number of the control points.
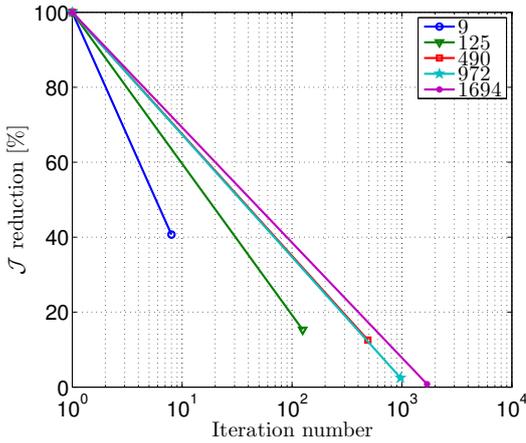
(a) Greedy algorithm convergence for different control points settings.
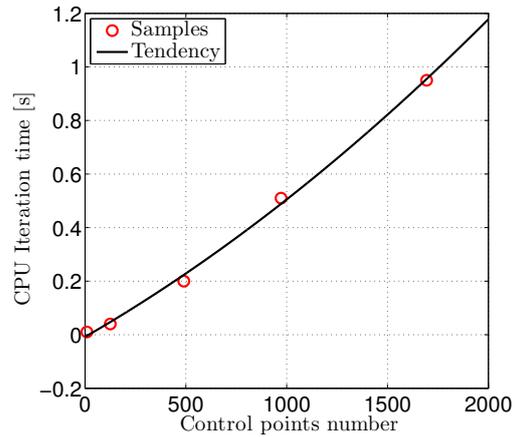
(b) Visualization of the computational costs tendency.

Also for the 3D case it is possible to appreciate the sensitivity of the convergence of the greedy algorithm with respect to the number of control points: in detail the greedy algorithm needs more iterations to reach convergence as the

number of the control points increases too. Concerning the final value of the cost functional, acceptable results are achieved only by using large numbers of parameters; finally, we can also uphold the computational cost tendency, that, as for the 2D case, shows a parabolic behavior by increasing control points number.

2.2   Inverse Distance Weighting

One of the most commonly used techniques for the interpolation of scattered data is the Inverse Distance Weighting strategy [110]. With respect to Figure 2.2 we will employ this method in order to solve both the fluid-structure interface and mesh motion problems. The IDW algorithm is an average moving interpolation that is usually applied to highly variable data. The main idea of this interpolation strategy lies in fact that it is not desirable to honour local high/low values but rather to look at a moving average of nearby data points and estimate the local trends [33]. The IDW technique, therefore, calculates a value for each grid node by examining surrounding data points that lie within a user-defined search radius. The node value is calculated by averaging the weighted sum of all the points. Data points that lie progressively farther from the node influence much less the computed value than those lying closer to the node. This point-to-point interpolation technique has the flexibility to handle arbitrary mesh topologies and, moreover, it is robust in case of large deformations. In contrast to RBF interpolation (that will be introduced in the next Section 2.3) this method results in an algebraic expression for the internal (flow) point displacements (fluid mesh) as function of the boundary deformation (representing the deformation of the structural interface). This explicit evaluation reduces the computational costs significantly and simplifies a lot the implementation of the mesh deformation routines.

We can now consider the simplest form of inverse distance weighting interpolation proposed by Shepard [110]: a generic way to find an interpolated value u at a given point $\mathbf{x}$ from a set of samples $u_k = u(\mathbf{x}_k)$, with $k = 1, 2, \ldots, \mathcal{N}$, is given by:

$$u(\mathbf{x}) = \sum_{k=1}^{\mathcal{N}} \frac{w(\mathbf{x}, \mathbf{x}_k)}{\sum_{j=1}^{\mathcal{N}} w(\mathbf{x}, \mathbf{x}_j)} \, u_k \qquad (2.16)$$

where, in general, $w(\mathbf{x}, \mathbf{x}_i)$ represents the weighting function:

$$w(\mathbf{x}, \mathbf{x}_i) = \parallel \mathbf{x} - \mathbf{x}_i \parallel^{-p}, \qquad (2.17)$$

being $\parallel \mathbf{x} - \mathbf{x}_i \parallel \geq 0$ is the Euclidean distance between $\mathbf{x}$ and data point $\mathbf{x}_i$, and
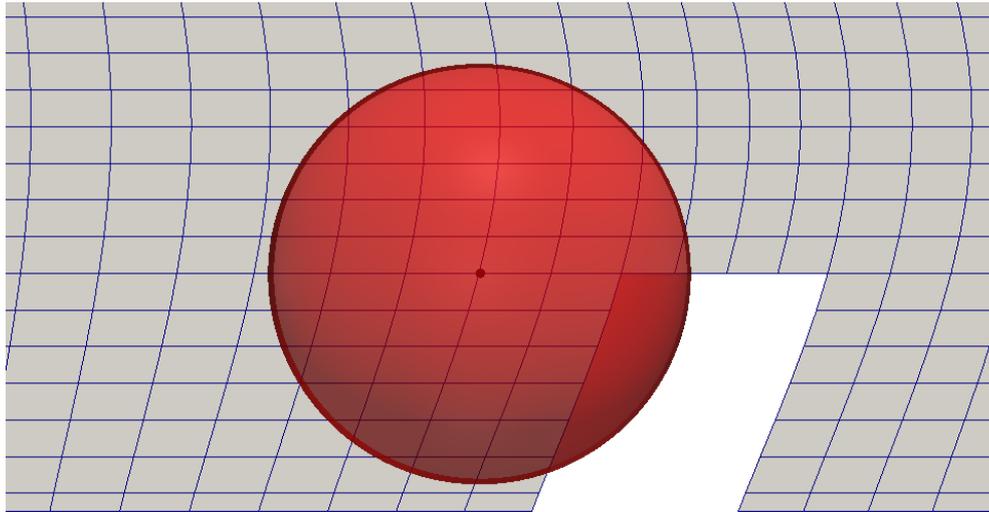
Figure 2.17: Visualization of the influence area (red) considered for the IDW method, within the 2D example introduced in Section 2.1.1.

p is a power parameter, typically equal to 2. Such an algorithm has both clear benefits and drawbacks. Among its benefits we could underline the easiness of its implementation, the possibility of dealing with N-dimensional spaces and the capability of working with any type of grid. On the other hand we have, within its original *Shepard's formulation*, low performances due to high interpolation costs (which are of the of order of $O(\mathcal{N})$): since too much weight is assigned also to distant nodes by considering, with respect to (2.16), $\mathcal{N}$ equal to the total number of grid nodes. In this way, in order to get such a situation depicted in Figure 2.17, simple control checks could be adopted. In particular, we decided, in order to reduce the computational effort, to involve in (2.16) a reduced set of grid nodes lying in surroundings of the $\mathbf{x}$, by simply implementing a selective test based on the Euclidean distance between $\mathbf{x}$ and the scattered data positions $\mathbf{x}_k$. Therefore, dealing with a set of scattered data $\mathcal{N}$, we can build, in order to perform the inverse distance weighting interpolation (2.16), a subset $\hat{\mathcal{N}}$ such that $\hat{\mathcal{N}} \subset \mathcal{N}$, where each element $u(\hat{\mathbf{x}}) \in \hat{\mathcal{N}}$ satisfies the following relation [123]:

$$\| \mathbf{x} - \hat{\mathbf{x}} \| \leq R, \tag{2.18}$$

where R indicates the user-defined radius of the sphere of influence adopted to improve the performances of the method. In Sections 2.4 and 2.5 some applications of the method proposed will be presented within fluid-structure interface and mesh motion problems, in the framework of two- and three-dimensional configurations.
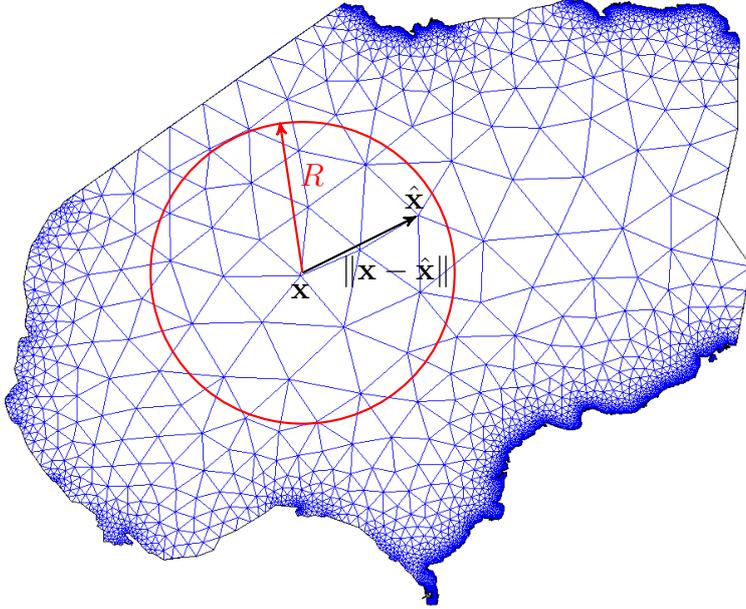
Figure 2.18: Schematic representation of the Inverse Distance Weighting interpolation process.

2.3 Radial Basis Functions

We focus now on the introduction of Radial Basis Functions interpolation technique in order to face both the fluid-structure interface and mesh motion problems, as illustrated in Figure 2.2. Interpolation with RBF nowadays has become a powerful tool in nonlinear multivariate approximation theory through scattered data and in the shape parametrization field due to its great approximation properties [20]. This technique is still based on the use of a set of parameters, the so-called "control points", as for FFD, but RBF is interpolatory. In fact, thanks to RBF interpolation process, the function values computed at the control points are exactly the ones of the function to be interpolated. This is due to the definition of the RBF map (2.19), which is made up of a sum of basis functions ϕ, centered at each control point $\mathbf{X}_C$, that allow to recover the exact function value at least over the interpolation sites[1] (as shown schematically in Figure 2.1). Another important key-point of RBF strategy relies in the way we can locate control points: in fact, instead of FFD where control points needed to be placed inside a regular lattice, with RBF we have no more limitations. In this way, instead of FFD techniques, we have with RBF the possibility to perform localized control points refinements.

---

[1] Within RBF strategy, the terms "interpolation sites" and "control points" assume exactly the same meaning [18].

RBF shape parametrization technique is based on the definition of a map, $\mathcal{M}(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}^n$, that allows the possibility of transferring data across non-matching grids and facing the dynamic mesh handling. The map introduced is defined as it follows:

$$\mathcal{M}(\mathbf{X}) = p(\mathbf{X}) + \sum_{i=1}^{\mathcal{N}_c} \gamma_i \, \phi(\| \mathbf{X} - \mathbf{X}_{C_i} \|), \qquad (2.19)$$

where $p(\mathbf{X})$ is a low-degree polynomial term, γ_i is the weight, corresponding to the a-priori selected $\mathcal{N}_c$ control points, associated to the i-th basis function, and $\phi(\| \mathbf{X} - \mathbf{X}_{C_i} \|)$ a radial basis function based on the Euclidean distance between the control points position $\mathbf{X}_{C_i}$ and $\mathbf{X}$. A radial basis function, generally, is a real-valued function whose value depends only on the distance from the origin, so that $\phi(\mathbf{x}) = \tilde{\phi}(\| \mathbf{x} \|)$. Among the most common used radial basis functions for modeling 2D and 3D shapes, we can consider:

(a) Gaussian splines [20]: $\qquad\qquad\qquad\qquad\quad \phi(\|\mathbf{X}\|) = e^{-\|\mathbf{X}\|^2/r^2}.$

(b) Multi-quadratic biharmonic splines [105]: $\qquad\quad \phi(\|\mathbf{X}\|) = \sqrt{\|\mathbf{X}\|^2 + r^2}.$

(c) Inverted multi-quadratic biharmonic splines [20]: $\quad \phi(\|\mathbf{X}\|) = (\|\mathbf{X}\|^2 + r^2)^{-1/2}.$

(d) Thin-plate splines [37]: $\qquad\qquad\qquad\qquad\quad \phi(\|\mathbf{X}\|) = \|\mathbf{X}/r\|^2 \ln\|\mathbf{X}/r\|.$

(e) Beckert and Wendland C^2 basis [18]: $\qquad\quad \phi(\|\mathbf{X}\|) = (1 - \|\mathbf{X}\|/r)^4_+(4\|\mathbf{X}\|/r + 1).$

where r represents a scaling factor controlling the basis shape, the subscript $_+$ indicates that only positive values must be considered[2]. In order to show the basis function behavior within interpolation process and to better understand the role of the parameter r, with respect to Figure 2.18, all the basis functions introduced are plotted. It must be remarked, in that way, the key-role played by the scaling factor r, whose values affects significantly [20] both the stability and the accuracy of the interpolation process. In fact, as we increase the value of r, we increase the influence area of each control point as well. Therefore, it is standard practice to choose values of r such that each basis function includes, at least, all the closest neighbor control and interpolation points.

---

[2]Negative values are set to zero.

(a) Gaussian splines (GS).

(b) Inverted multi-quadratic biharmonic splines (IMQ).

(c) Multi-quadratic biharmonic splines (MQ).

(d) Thin-plate splines (TPS).

(e) Beckert and Wendland C^2 basis (B&W).
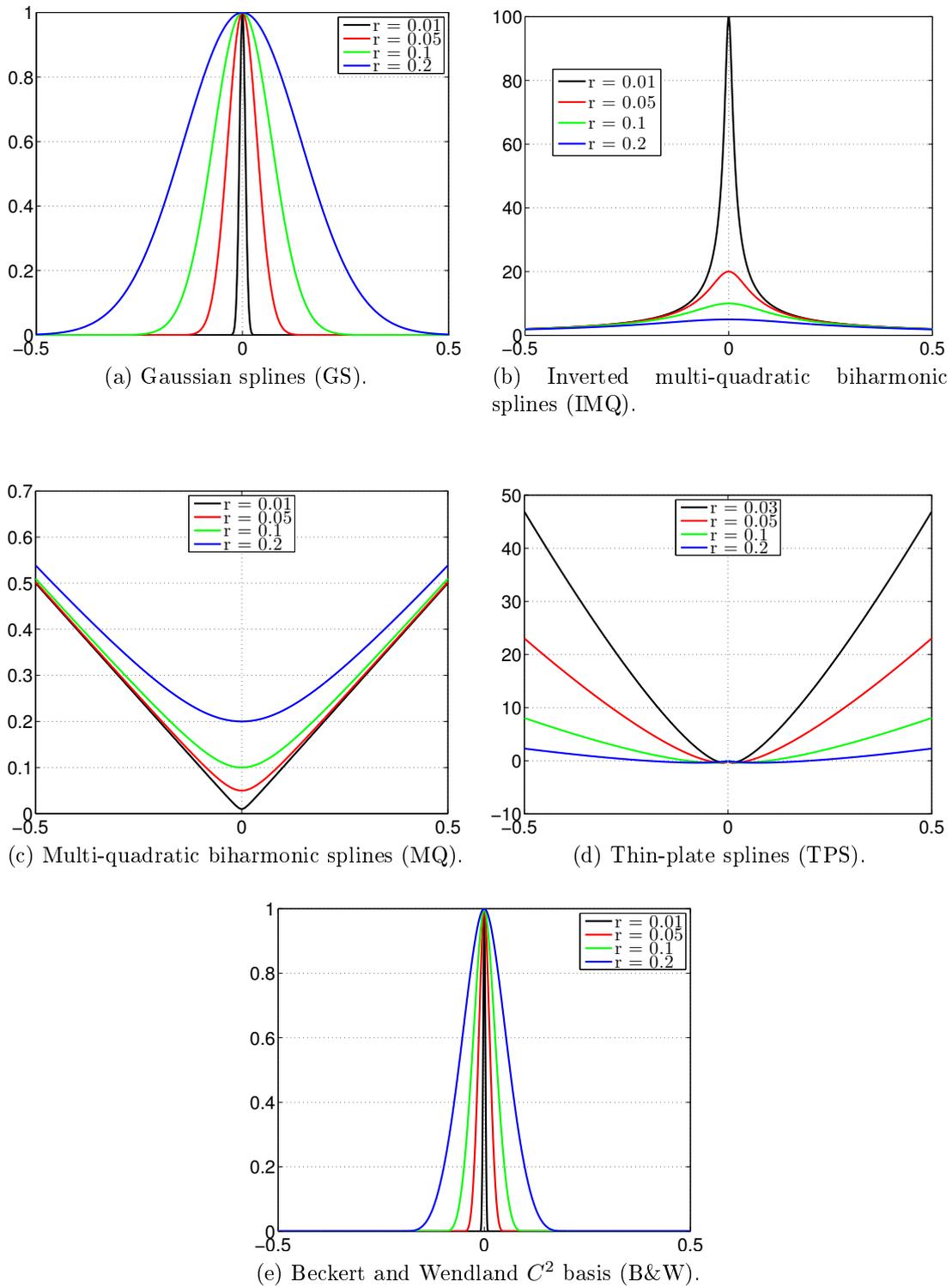
Figure 2.18: Visualization of different radial basis functions considered (a)-(e) and their sensitivity to the scaling factor r.

With respect to (2.19), generally a linear function is chosen to represent the polynomial term $p(\mathbf{X}) = \mathbf{c} + \mathbf{Q}\mathbf{X}$, where, for the three-dimensional case, $\mathbf{c} \in \mathbb{R}^3$ and $\mathbf{Q} \in \mathbb{R}^{3\times3}$. In detail, $\mathbf{c}$ and $\mathbf{Q}$ represent the constant and the linear terms of the polynomial $p(\mathbf{X})$, respectively. In that way we can rewrite the RBF map expression as it follows:

$$\mathcal{M}(\mathbf{X}) = \mathbf{c} + \mathbf{Q}\mathbf{X} + \mathbf{W}^T\mathbf{d}(\mathbf{X}) \tag{2.20}$$

being $\mathbf{d}(\mathbf{X}) = \{\phi(\|\mathbf{X}-\mathbf{X}_{C_1}\|), \ldots, \phi(\|\mathbf{X}-\mathbf{X}_{C_{\mathcal{N}_c}}\|)\} \in \mathbb{R}^{\mathcal{N}_c}$ and $\mathbf{W} = [\gamma_1, \ldots, \gamma_{\mathcal{N}_c}] \in \mathbb{R}^{\mathcal{N}_c\times3}$. The map (2.20), at this point, is function of $3 \times \mathcal{N}_c + 12$ unknowns coefficients to be computed thanks to the interpolation conditions: in fact, we look for a transformation which is defined by moving control points from the initial shape configuration to the deformed one. In this way, defining their initial $\mathbf{X}_C = [\mathbf{X}_{C1}, \ldots, \mathbf{X}_{C_{\mathcal{N}_c}}] \in \mathbb{R}^{\mathcal{N}_c\times3}$ and the final $\mathbf{Y}_C = [\mathbf{Y}_{C1}, \ldots, \mathbf{Y}_{C_{\mathcal{N}_c}}] \in \mathbb{R}^{\mathcal{N}_c\times3}$ positions, the unknown $\mathcal{N}_c \times 3$ weights collected in $\mathbf{W}$ are found by fulfilling the interpolation constraints:

$$\mathcal{M}(\mathbf{X}_{C_i}) = \mathbf{Y}_{C_i}, \qquad \forall i = 1, \ldots, \mathcal{N}_c. \tag{2.21}$$

When considering the polynomial term $p(\mathbf{X}) = \mathbf{c} + \mathbf{Q}\,\mathbf{X}$, in order to find the residual 12 unknown coefficients, the system is completed by additional constraints, which can represent the conservation of total force and momentum [85], such that:

$$\sum_{i=1}^{\mathcal{N}_c} \gamma_i = 0, \tag{2.22}$$

$$\sum_{i=1}^{\mathcal{N}_c} \gamma_i X_{C1,i} = \sum_{i=1}^{\mathcal{N}_c} \gamma_i X_{C2,i} = \sum_{i=1}^{\mathcal{N}_c} \gamma_i X_{C3,i}, \tag{2.23}$$

being $\mathbf{X}_{C_i} = (X_{C1,i}, X_{C2,i}, X_{C3,i})$. Moreover, in order to express, in our parametrized framework, the control points position in the deformed configuration, we can write:

$$\mathbf{Y}_c(\mathbf{\Pi}) = \mathbf{X}_c + \mathbf{\Pi} \tag{2.24}$$

where $\mathbf{\Pi} \in \mathbb{R}^{\mathcal{N}_c\times3}$ represents the displacement of control points, which are generally located on the boundary of the shape to be deformed. Finally, within the variables introduced, we can write the parametric RBF mapping operator $\mathcal{M}(\cdot, \mathbf{\Pi}): \mathbb{R}^3 \to \mathbb{R}^3$ as it follows:

$$\mathcal{M}(\mathbf{X}, \mathbf{\Pi}) = \mathbf{c}(\mathbf{\Pi}) + \mathbf{Q}(\mathbf{\Pi})\mathbf{X} + \mathbf{W}(\mathbf{\Pi})^T\mathbf{d}(\mathbf{X}), \tag{RBF map}$$

where the coefficients $\mathbf{c}(\mathbf{\Pi})$, $\mathbf{Q}(\mathbf{\Pi})$ and $\mathbf{W}(\mathbf{\Pi})$ satisfy the constraints (2.21), (2.22) and (2.23). We can rewrite the constraint relations into a more suitable way, signally to best-fit the fluid-structure interface and mesh motion problem applications: in that way it will be possible to compute the unknown coefficients values as solution of a linear system such that the parametrization term is held in the right-hand-side:

$$\underbrace{\begin{bmatrix} \mathbf{D}_{CC} & \mathbf{1}_C & \mathbf{X}_C \\ \mathbf{1}_C^T & \mathbf{0} & \mathbf{0} \\ \mathbf{X}_C & \mathbf{0} & \mathbf{0} \end{bmatrix}}_{\tilde{\mathbf{H}}_{CC}} \begin{Bmatrix} \mathbf{W} \\ \mathbf{c}^T \\ \mathbf{Q}^T \end{Bmatrix} = \begin{Bmatrix} \mathbf{Y}_C(\mathbf{\Pi}) \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix}, \tag{2.25}$$

where $\mathbf{D}_{CC} \in \mathbb{R}^{\mathcal{N}_c \times \mathcal{N}_c}$ represents the interpolation matrix, component-wise $\mathbf{D}_{CC_{i,j}} = \phi(\|\mathbf{X}_{C_i} - \mathbf{X}_{C_j}\|)$, and $\mathbf{1}_C \in \mathbb{R}^{\mathcal{N}_c \times 1}$ represents a unit vector. It must be underlined that when considering, with respect to (2.19), the polynomial term $p(\mathbf{X})$, the RBF interpolation problem always admits a unique solution, being the left-side matrix symmetric and positive definite [20]. When dealing with interpolation problems involving a small number of control points the system (2.25) would be solved easily by means of suitable direct methods. Notwithstanding, difficulties may arise due to matrix high-condition number related to both the employment of large control points number and high scaling factor values [20]. In fact, if we consider very wide and flat basis functions we would not be able to distinguish the contribution of each control point on a near interpolation point, resulting in a close to singular matrix. Once calculated through (2.25) the unknown coefficients of the RBF interpolant, it will be possible to evaluate the global RBF map. From a practical point of view, in order to perform the interpolation on a set of $\mathcal{N}_I$ interpolation points $\{\mathbf{X}_{I_k}\}_{k=1}^{\mathcal{N}_I}$ we can directly compute their function values $\mathbf{Y}_I$ as it follows:

$$\mathbf{Y}_I = \underbrace{\begin{bmatrix} \mathbf{D}_{IC} & \mathbf{1}_I & \mathbf{X}_I \end{bmatrix}}_{\tilde{\mathbf{H}}_{IC}} \begin{Bmatrix} \mathbf{W} \\ \mathbf{c}^T \\ \mathbf{Q}^T \end{Bmatrix}, \tag{2.26}$$

where $\mathbf{D}_{IC} \in \mathbb{R}^{\mathcal{N}_I \times \mathcal{N}_C}$, and component-wise reads $\mathbf{D}_{IC_{i,j}} = \phi(\|\mathbf{X}_{I_i} - \mathbf{X}_{C_j}\|)$, with $\mathbf{1}_I \in \mathbb{R}^{\mathcal{N}_I \times 1}$ indicating a unit vector and $\mathbf{X}_I \in \mathbb{R}^{\mathcal{N}_I \times 3}$ represents the matrix containing the coordinates of the interpolation points. At this point, by introducing expression (2.25) in (2.26) we obtain:

$$\mathbf{Y}_I = \begin{bmatrix} \mathbf{D}_{IC} & \mathbf{1}_I & \mathbf{X}_I \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{D}_{CC} & \mathbf{1}_C & \mathbf{X}_C \\ \mathbf{1}_C^T & \mathbf{0} & \mathbf{0} \\ \mathbf{X}_C & \mathbf{0} & \mathbf{0} \end{bmatrix}^{-1}}_{\tilde{\mathbf{H}}} \begin{Bmatrix} \mathbf{Y}_C(\mathbf{\Pi}) \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} \tag{2.27}$$

We can define the interpolation matrix $\mathbf{H}$ from control points to the interpolation ones by considering the first $\mathcal{N}_C$ columns of $\tilde{\mathbf{H}}$ (due to the definition of the vector in the right-hand-side in (2.27), whose latest four elements are zero), such that:

$$\mathbf{Y}_I = \mathbf{H} \; \mathbf{Y}_C \tag{2.28}$$

By acting in this way no orthogonal projection and search algorithm is needed, but the computation involves the inversion of a relatively small (depending on the number of the control points) matrix. From a computational point of view, it is not useful to assemble the whole matrix $\mathbf{H}$, due to high costs involved $O(\mathcal{N}_I \times \mathcal{N}_c^2)$. In that way it is better to first multiply $\tilde{\mathbf{H}}_{CC}^{-1}$, defined in (2.25) with $[\mathbf{Y}_C, \mathbf{0}, \mathbf{0}]^T$ and then, multiplying the resulting vector with $\tilde{\mathbf{H}}_{IC}$, defined in (2.26), for a computational effort of order $O(2\mathcal{N}_I \times \mathcal{N}_c)$. Moreover, the interpolation matrix could be also evaluated once for all (*offline*) at the beginning and stored, so that in run-time (*online*) the first method would cost only $O(\mathcal{N}_I \times \mathcal{N}_c)$, twice as fast as the second approach. At that point, in order to highlight the potentiality of this shape parametrization strategy, some applications of RBF technique are proposed in Figure 2.19 and 2.20.

(a) Initial configuration.   (b) Final deformations.
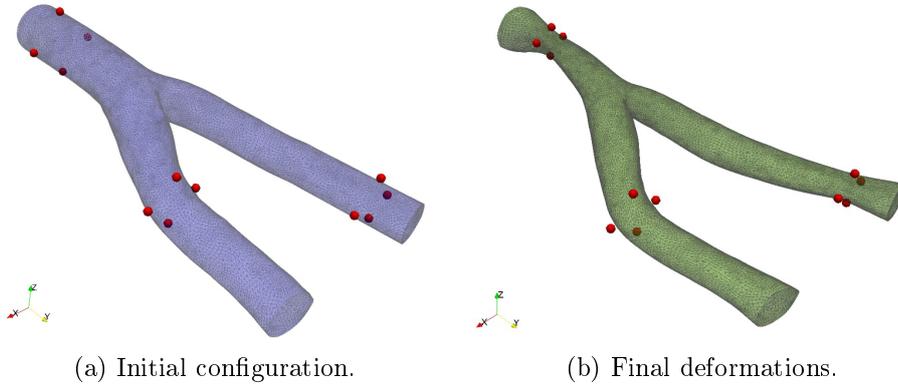
Figure 2.19: Geometrical transformation on a carotid artery bifurcation operated by radial basis functions with 12 control points.

(a) Initial configuration, from below.

(b) Initial configuration, from above.

(c) Deformations and comparison.
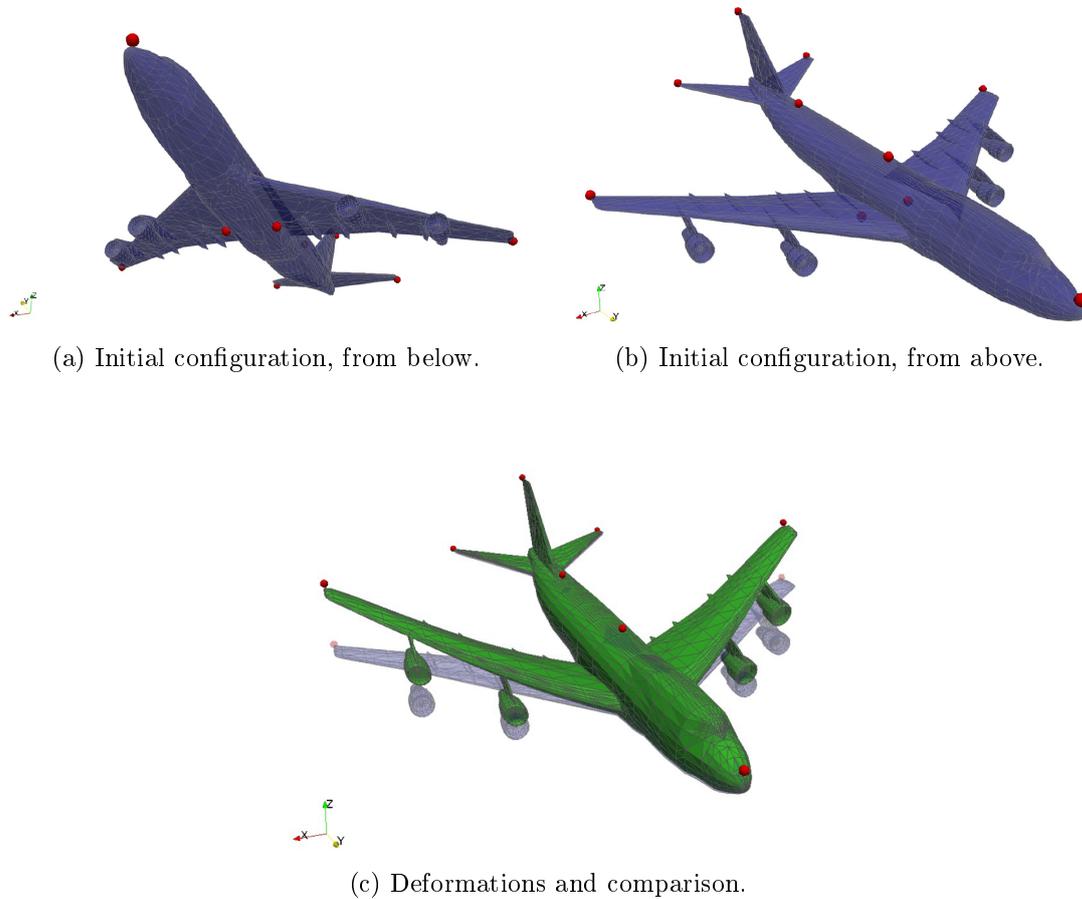
Figure 2.20: Simulated aeroelastic deformations on a complete commercial aircraft configuration operated by radial basis functions, centered in 10 control points.

2.4 Some applications of multivariate interpolation methods

We focus now on some applications of the numerical schemes proposed, in order to investigate both their interpolation properties and computational costs. In particular we will study two different interpolation problems concerning one and two-dimensional cases. In the first 1D example [33], the multivariate interpolation strategies considered in this work, RBF and IDW, are compared within a smooth analytical test-problem in order to be able to investigate their properties separately. In that way, on an analytical-defined boundary, we have considered different discretization for both the fluid and structure grids. In detail the boundary shape is described by:

$$b(x) = 0.5 \sin(2\pi x), \quad x \in [-0.5, 0.5]. \tag{2.29}$$

On this boundary we state that fluid and structural meshes are non-conformal since they differ by their discretization: in detail, we fix the number of fluid grid nodes to be equal to $n_f = 2560$, while the number of the structural ones will change with the following law:

$$n_s = 2^k \cdot 10, \quad k = 0,\ 1,\ 2,\ 3,\ 4,\ 5,\ 6. \tag{2.30}$$

In the structural points we imposed an assigned displacement in the form of a cosine $d(x) = \cos(2\pi x)$ to obtain the final analytical deformed configuration. Within this example, we will test the capabilities of the methods proposed and we will compare, in the framework of RBF technique, all the radial basis functions introduced and depicted in Figure 2.18. With respect to the previously introduced background, in Figure 2.21 we illustrate the problem settings:
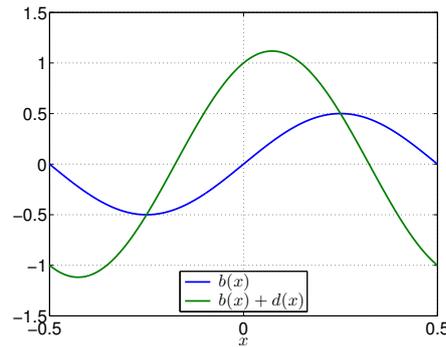
Figure 2.21: 1D interpolation problem settings.

The displacement imposed at the structural points is interpolated, thanks to RBF and IDW techniques, in order to obtain the displacement values in the fluid domain. With respect to Figure 2.22, the L_2-error of the displacement in the flow point versus the number of structural points is shown, with different values for the shape influence factor r:

(a) $r = 0.001$.   (b) $r = 0.005$.
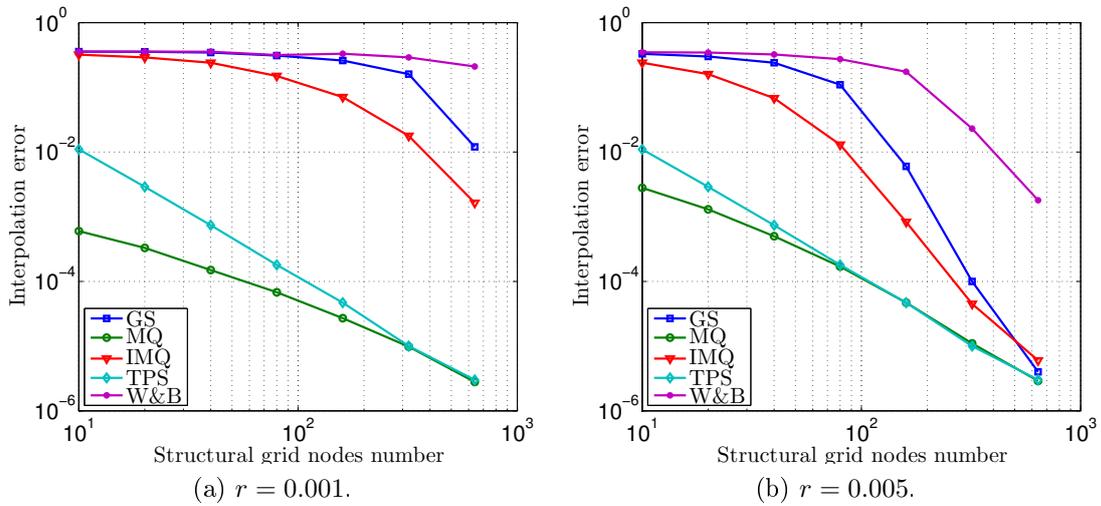
Figure 2.22: L_2-error of the displacement in the flow point obtained by considering two different values for the shape factor.

On the base of these results it is possible to underline the sensitivity of the radial basis functions adopted, related to the shape factor r.
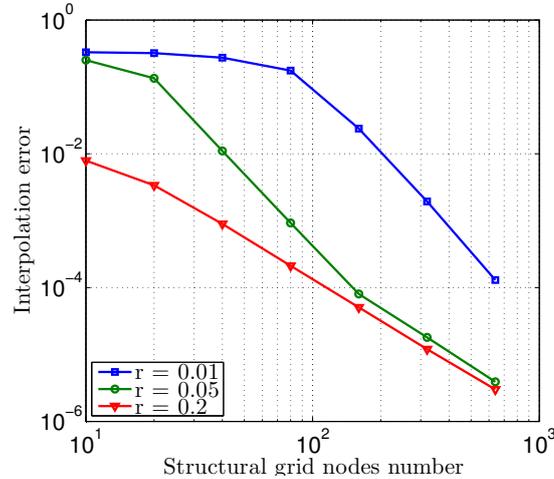
Figure 2.23: Wendland and Beckert (W&B) C^2 basis functions sensitivity to scale factor r.

In particular, since both the Multi-quadratic biharmonic (MQ) and Thin-plate splines (TPS) are globally supported functions, as shown if Figure 2.18c and 2.18d, respectively, their behavior is not influenced significantly by changes in the shape factor. On the contrary, being the GS, TPS and W&B compactly supported radial basis functions, they suffer from scale factor values variations: particularly,

for larger values of r we can achieve more accurate solutions [73]. Certain high sensitivity to r is related to the Wendland and Beckert C^2 basis functions: for this reason we investigated their behavior considering greater values for r, as shown in Figure 2.23. From this analysis both the MQ and TPS basis functions would represent a good choice in terms of accuracy of the achievable solution and non-sensitivity related to the scale parameter r. On the other hand, with respect to Figure 2.24, they are characterized by a slightly higher computational effort if compared to the other radial basis functions adopted (even if it is generally much smaller than the time needed by the flow solver, within the FSI problem).
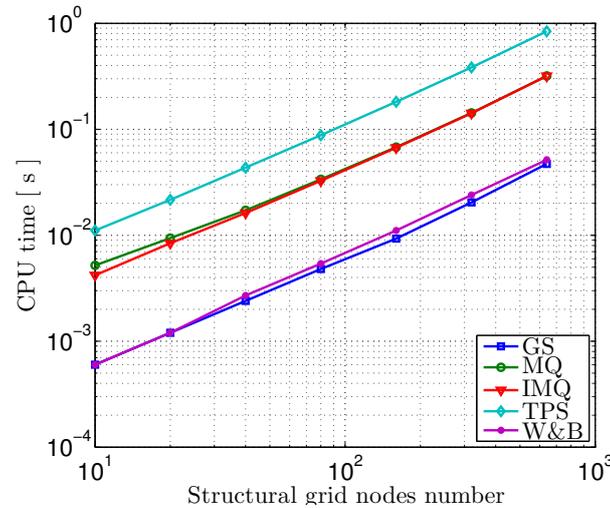
Figure 2.24: Computational effort for the fluid-structure interface 1D example, with $r = 0.005$.

At this point, we focus our attention on the inverse distance weighting technique in order to solve the same 1D problem: as seen in Section 2.2 we consider a parametric analysis by adopting different values for the radius of influence R. Since the problem to be solved is characterized by smoothness of the displacement to be transferred between non-matching grids, we expect to obtain great results in terms of accuracy of the solution with low computational costs as shown in Figures 2.25 and 2.26, where it is possible to look at the improvement in the results achieved. We obtained, as expected, good results in terms of interpolation error (comparable to the ones we achieved with RBF) with a slight computational effort that is quite not dependent from both the number of structural grid nodes and from the chosen value of R. By looking at Figure 2.25 we can also observe how the curve related to $R = 0.5$ shows better results, with respect to others, when dealing with the coarse structural grid; on the other hand, for the finest structural discretization, the curves related to smaller values than $R = 0.5$ (in detail for $R = 0.01, 0.05, 0.1$) would provide better results. In fact, as the number of structural grid nodes increases, to adopt smaller values for the radius
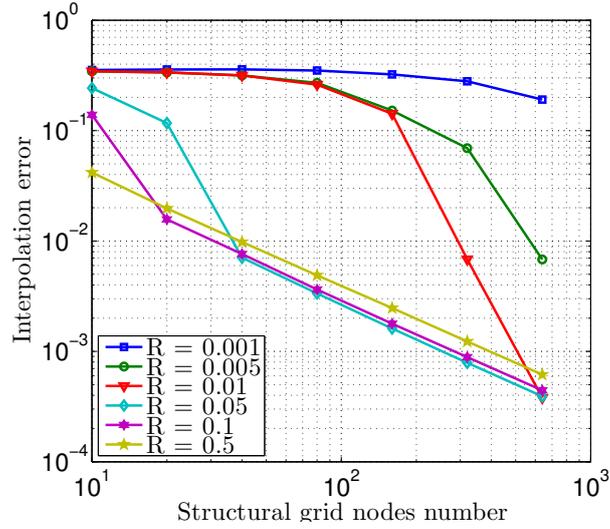
Figure 2.25: L_2 norm of the interpolation error for the parametric analysis of the 1D example using IDW technique.
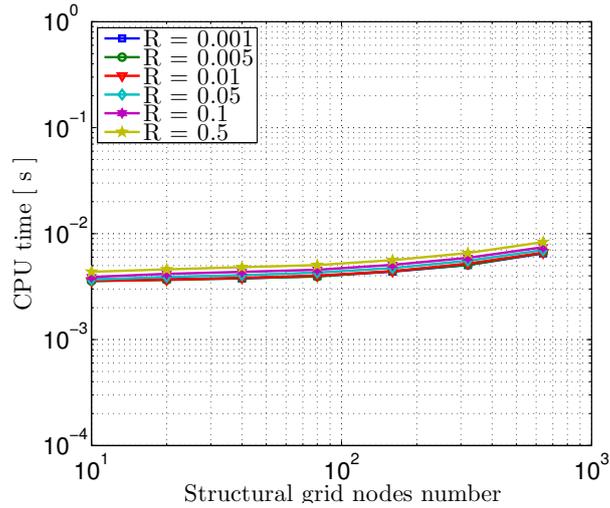
Figure 2.26: Computational costs involved using IDW technique for different radii of influence.

of influence represents a better choice, since to recover local deformations it is preferable to weight the contribution of those nodes lying in its closeness, thus by choosing small values for R. To better assess the interpolation properties of different radial bases and inverse distance weighting method, we have performed a second 2D test [73], that, unlike the first one, is characterized by a not smoothed function to be interpolated, defined as it follows:

$$f(x,y) = sin(2\pi x) \, cos(3\pi y) \, e^{xy}, \qquad x, y \in [-1, 1]. \tag{2.31}$$

We assumed that the solution is given over a 25×25 quadrilateral uniform structured grid (structure mesh) and it is interpolated over a uniform and structured (fluid) grid (whose number of nodes is equal to $n_f = \{900, 1600, 2500\}$ for each of the analysis that we performed), thus non-conforming to the structural one. Also, in this case, we consider as control points for the RBF interpolation the whole number of fluid grid nodes that compose the fluid mesh. In Figure 2.27a and 2.27b it is possible to visualize the structural mesh realized and its function contour plot:

(a) Structural mesh. (b) Function contour plot.
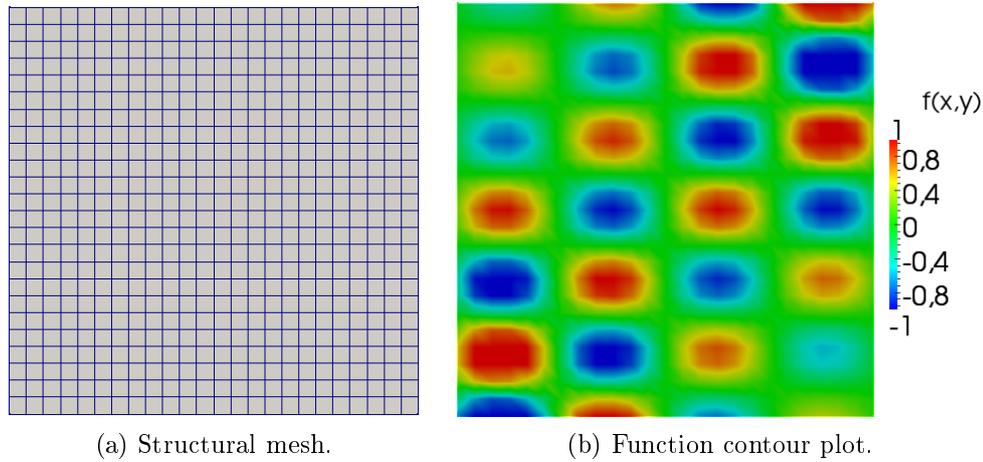
Figure 2.27: Representation of both the structural mesh (on the left-hand-side) and the function to be interpolated (on the right-hand-side) over the non-matching fluid grid.

In Figure 2.28 both the L_2 norm of the interpolation error and the computational costs are shown. Results are in agreement with the theory: in fact, the error decreases as the support radius increases; it is important to observe that, in relation to Gaussian splines (GS), the error decreases only when considering small values for the parameter r, while for larger values, the problem becomes unstable due to the ill-conditioning of the matrix $\tilde{\mathbf{H}}_{CC}$, defined in (2.25). In that case round-off errors get amplified and the interpolation becomes completely biased. Moreover, it is possible to state the best behavior, in terms of accuracy, of the MQ basis functions, such that they guarantee acceptable solutions independently both from the grid nodes number and from the scale factor. It is also important to remark the fact that the reported results have been computed without the use of the polynomial term in the interpolation process, due to the fact that, since all control points are coplanar, the corresponding matrix $\tilde{\mathbf{H}}_{CC}$ would become singular. Concerning the computational costs, as seen for the 1D example, the CPU time needed by the interpolation process is, for every basis considered, quite low

with a linear increasing tendency related with the number of the fluid grid nodes.

(a) 30×30 fluid nodes.

(b) 40×40 fluid nodes.

(c) 50×50 fluid nodes.

(d) Computational costs.
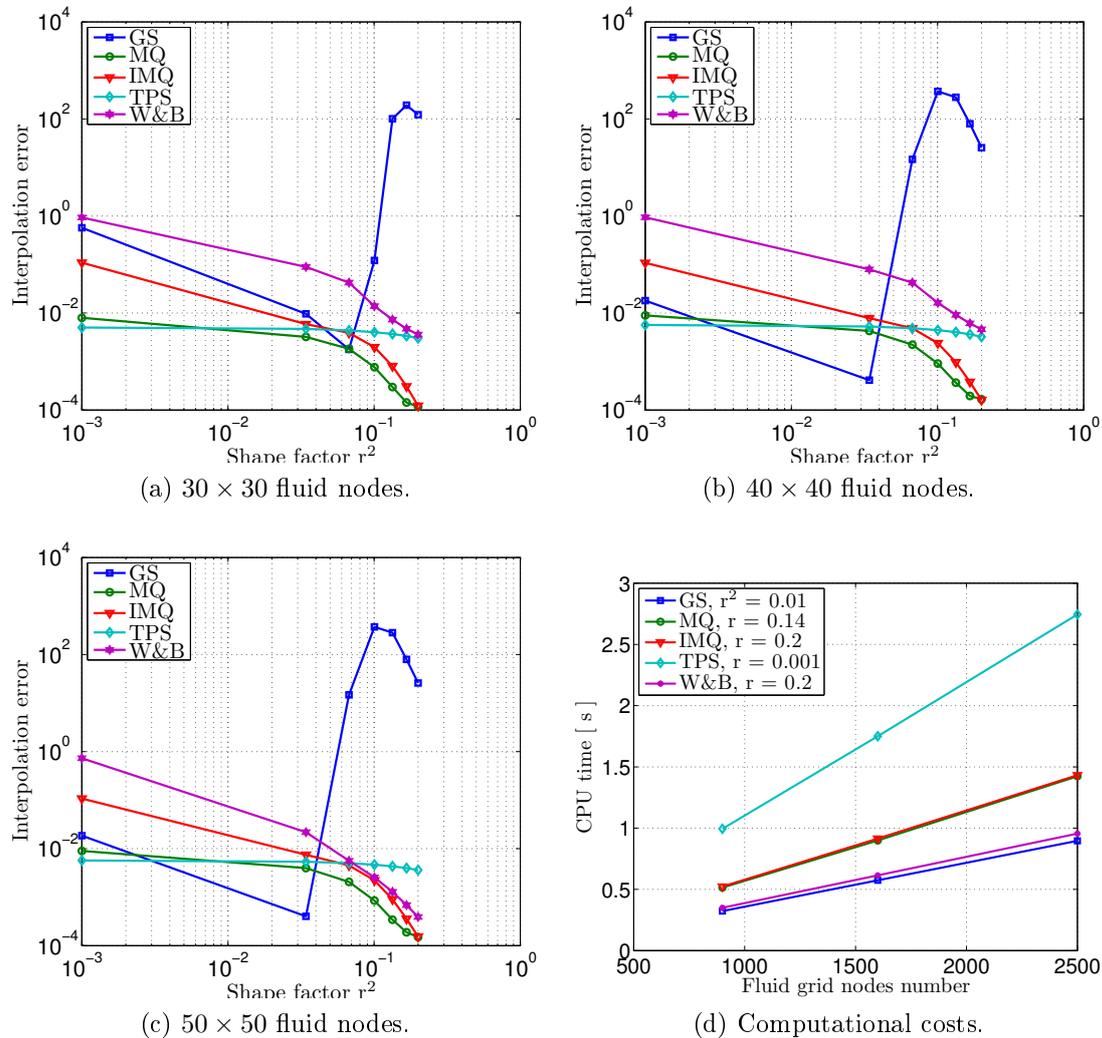
Figure 2.28: Results obtained within the 2D example adopting RBF method. Comparison of the accuracy achievable with different basis functions and their computational effort.

Finally, it is possible to show the results obtained thanks to IDW technique: in that case, we decided to speed-up the method by adopting, as criterion to be used to find those points considered in the weighting process, not the ones lying witin an inscribed sphere of radius R, but the closest N_P points to the interpolation one. At the base of this choice there is the fact that the function to be transferred across non-matching grid is not smoothed and its values change a lot from one point to the other: in such a case the adoption of a selection method like the one proposed is more suitable. In Figure 2.29 the main results obtained

within this example are illustrated .

(a)   (b)
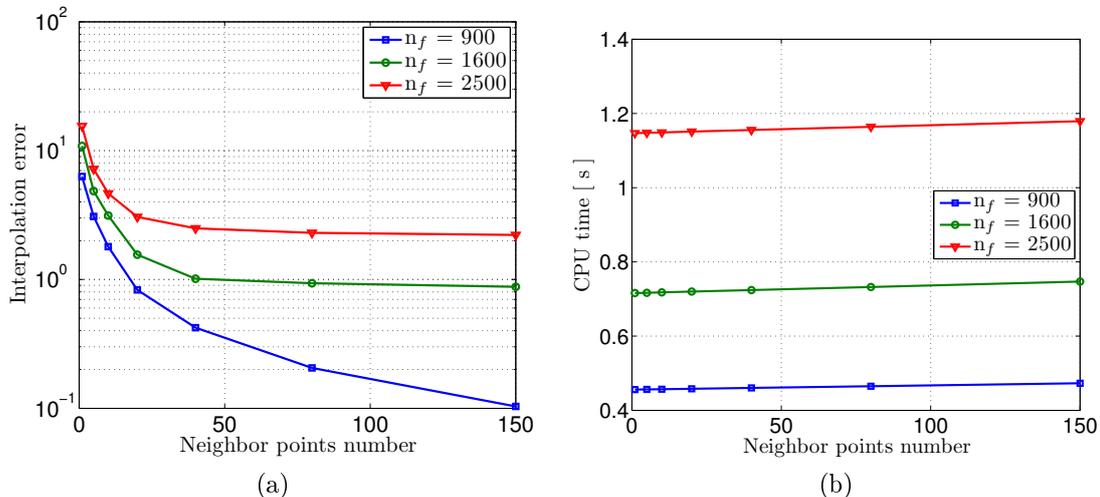
Figure 2.29: Results obtained within the 2D example adopting IDW method. Comparison of the accuracy achievable with different numbers of neighbor weighting points and their computational effort.

With respect to Figure 2.29a, we can observe that the curve related to the coarse fluid mesh ($n_f = 900$) drives to best results: this is due to the fact that the number of fluid nodes is is quite the same (instead of the other two cases where $n_f = 1600, 2500$) of the structural ones $n_s = 750$, resulting in nearly conformal fluid-structure meshes. We can conclude that, at the base of the jeopardizing behavior of the function to be interpolated, in this case the choice of radial basis functions strategy would be preferable. In fact, it is possible to underline that when dealing with fluid-structure interface problems characterized by the transfer of patchy data across interfaces, the use of RBF method represents quite a unique choice in order to get good results with an acceptable computational effort [52].

2.5   Comparison within dynamic mesh handling

While in the previous section we compared, within 1D and 2D examples, the interpolation properties of the multivariate interpolation methods in order to solve the fluid-structure interface problem, here we focus our attention on the potentialities of FFD, RBF and IDW techniques concerning the mesh motion problem. In particular, we are going to compare these strategies through the 3D example introduced in Section 2.1. In this way, at the base of the results previously achieved by adopting the FFD method, we will compare both the computational costs and the achievable mesh quality using both the IDW and RBF strategies. It must be remarked, concerning RBF method, that one of its

advantages is that the same map used for transferring data across non-matching grids, can be exploited for moving the fluid mesh, accordingly to the interface deformation. In that way it would be enough to enlarge the set of interpolation points to all the interior nodes of the fluid mesh. In detail we decided to test these methods with respect to such an example, in order to assess the most suitable strategy to be adopted within the test-case of internal fluid flow that will be proposed in the last chapter of this work. The analysis pattern has been organized as it follows: we will perform a set of parametric simulations involving the RBF strategy, by testing the different basis functions with several scale factor values, r; therefore, we will face the problem adopting the IDW method considering different values of the radius of influence R. Finally, at the base of the results obtained, it will be possible to identify the most suitable strategies to be taken into account in Chapter 5, when dealing with the FSI simulation. In Figure 2.30 are proposed the results obtained within the numerical simulations performed using RBF.

(a) Accuracy of RBF methods vs. scale factor r^2.

(b) Computational costs.
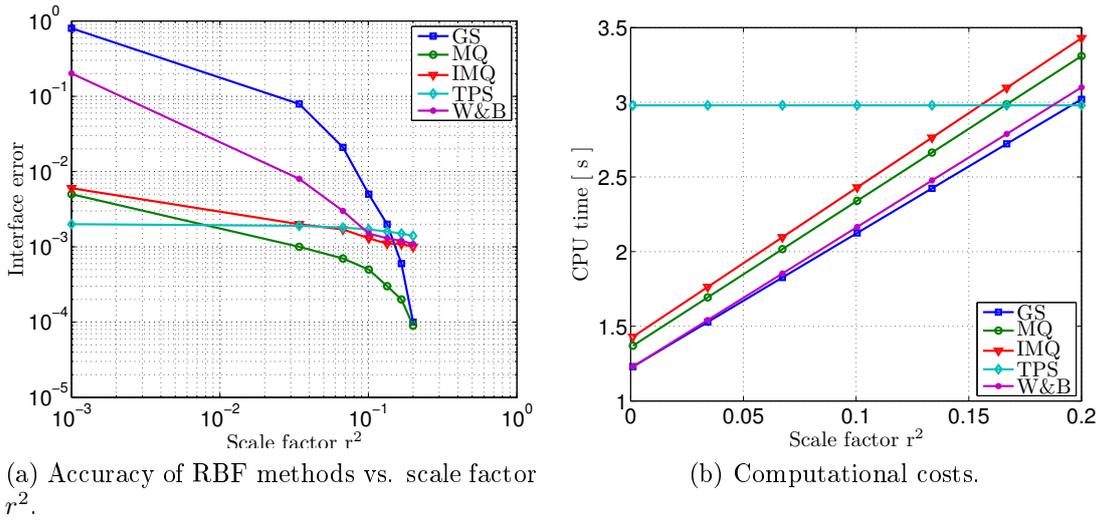
Figure 2.30: Results obtained within the 3D example adopting RBF method. Comparison of the accuracy achievable with different basis functions and their computational effort.

We considered, as control points, all the fluid nodes lying on the fluid-structure interface and, indeed, in order to highlight separately the performances related to solution of the mesh motion problem form the fluid-structure interface one, we have chosen to deal with conformal meshes. By looking at Figure 2.30a, for values of r greater than 0.3 the L_2 norm of the error between the structural and the fluid grid nodes positions over the interface is small ($\sim 10^{-3}$) and, moreover, it does not change from one basis function to the other. If we consider the

interpolation errors reported in Figures 2.28 and 2.30a, it is possible to state that Thin-Plate splines, among the other RBF basis functions, have a very small dependence on the choice of r, since the error is quite independent from variations in the support radius. Being TPS basis functions quite independent from r, its related computational time, as shown in Figure 2.30b, would not change with respect to variation of such a parameter. At the base of their better results (which are not highly dependent from the choice of r) and from their quite low computational costs, within RBF strategies we decided to select as basis functions to be adopted for the FSI analysis in Chapter 5 the Multi-Quadratic biharmonic splines and Thin-Plate splines. Finally, in Table 2.3 we report the main results obtained thanks to the use of IDW technique by varying the value of the radius of influence:

R	L_2 interface error	CPU time [s]
0.010	$1.445E^{-4}$	0.89
0.1008	$1.112E^{-4}$	1.82
0.2006	$8.732E^{-5}$	1.97
0.3004	$6.671E^{-5}$	2.02
0.4002	$2.198E^{-5}$	2.09
0.5000	$1.021E^{-5}$	2.15

Table 2.3: Results obtained within the 3D example, adopting IDW method. Comparison of the accuracy achievable with different radii of influence R, and their computational costs.

From these results it is possible to appreciate on one hand the accuracy of the method, in terms of achievable L_2 norm of the error (also when small values of R are taken into account), on the other hand its cheapness in terms of computational costs which, indeed, do not arise significantly from 2D to 3D environments. At the base of the previously obtained results we propose, within FSI moving mesh problems, the use of IDW technique instead of both RBF and FFD. In fact, although RBF leads to great results, it is whatever more expensive, and, also of primary importance but not handled in this work, it may not be highly suitable for a parallel computational environment due to the dense pattern of the interpolation matrix **H** [73]. With respect to FFD technique, on the base of the results obtained, we can state that it is unacceptably expensive since to update the fluid mesh configuration we have to solve an optimization sub-problem; moreover, due to its limitation of not being interpolatory, as well as its severe restrictions on control points locations, it seems to be not so suitable for this kind of applications.

We need to remark, at the end of this chapter, that although in this work shape parametrization strategies (FFD and RBF) will be adopted to handle both the mesh motion and interface problems for FSI, they represent also an excellent

tool within Reduced Order Modeling (ROM). Within such a framework the aim is to reduce as most as possible the computational and geometrical complexities in systems governed by partial differential equations, in particular for those applications requiring an ever increasing level of accuracy leading to high-complexity models and to a curse of dimensionality. In general, to the end of replacing a large scale model (from a computational and a geometrical point view) by a model of very lower dimensions, shape parametrization strategies would give the possibility to represent and describe complex geometries through low dimensional spaces instead of using geometrical properties themself. As we have shown, RBF and FFD represent good choices, since they allow significant reductions in the number of parameters able to describe and parametrize complex geometrical models. In literature a wide range of applications of RBF and FFD employed to handle shape parametrization within reduced order modeling are present. For example, in [67, 74], FFD techniques are used in optimal design of high-performance sail boats: in that case FFD aimed at shape optimization on ship-related geometries, like a bulb and a rudder of a sail boat. Indeed, within applications from the haemodynamic framework, in [81] RBF strategy has been used within the development of a new model reduction technique aimed at real-time blood flow simulations on a given family of geometrical shapes of arterial vessels: in detail, to describe and parametrize a set of arterial vessel geometries. Within FSI problems, a lot of work has also been devoted to the study of the parametric fluid-structure coupling, like in [69, 70, 71, 96], by means FFD techniques. Indeed, among shape optimization problems within the framework of ROM techniques, FFD [9, 82, 104] and RBF [62, 63] have been used to parametrize geometrical objects, with applications spanning many engineering fields like aerodynamics, haemodynamics and naval engineering. At the base of this brief bibliographic survey, devoted to highlight the central role of shape parametrization techniques within ROM strategies, relies the idea for future works where we aim at using FFD and RBF parametrization not only to tackle the mesh motion and interface problems for FSI, but also to extent their application fields into the ROM scenario.

Chapter 3

Dimensionality reduction for shape parametrization

In this Chapter we focus our attention on the development of shape parametrization techniques that will allow to overcome some of the limitations that characterize the state of the art of the strategies proposed, namely of FFD and RBF, within their applications to the mesh motion problem. In particular, the first issue to be faced, related to these techniques, lies in the choice of the control points setting, and, in detail, of both the number of parameters to be taken into account, and their locations. As we have seen in Chapter 2, both the accuracy and the computational costs of these techniques show a significant sensitivity related to the control points pattern adopted: here we will propose, concerning FFD and RBF, different strategies in order to be able to tackle these problems in a deterministic way, such that it will be possible to standardize this procedure, leading to a reduction of the computational costs too. In order to tackle such problems, in literature a wide range of empirical strategies is presented [113]: they try to identify the best control points setting by performing a sensitivity analysis of the achievable results considering different, *a priori*, parameters patterns, and, at the end of this simple procedure, they select the one leading to the best results in terms of mesh quality. It is possible to understand how, in such an empirical way, the final result would be affected by the user-defined choices made when building the candidate setting configurations for the control points, and, in particular, no guarantee is provided, related to the fact that the optimal configuration found represents really the best. Basically, here we propose, concerning RBF and FFD techniques, a methodology to overcome these problems through a high-fidelity deterministic procedure: the goal in the FSI framework will consist in the establishment of a standard path able to automatically drive to the optimal control points setting in order to represent both the initial shape configuration and its deformation during the fluid-structure interaction process. Moreover, another important issue to be taken into account when dealing with FSI problems is related to the possibility of studying the multiphysics interaction

between fluid and structure when the latter consists in a free-body, that, during the simulation process, would not only get deformed, but, it could be subject to rigid translations and/or rotations. In that way, we need to estimate the entity of such a structure movement, in order to treat correctly the mesh motion problem and to be able to update the fluid mesh accordingly to the structure behavior. With respect to the previously introduced background, this Chapter has been organized as it follows: in Section 3.1 we propose the strategy considered in order to tackle the problem of the identification of structural rigid translations and rotations, showing both its possible application fields and limitations. Within the identification of suitable control points settings for mesh motion problems within FSI, throughout Section 3.2 we present the procedure adopted for the adaptive selection of parameters for RBF technique. Furthermore, in Section 3.3, we propose a patch-approach for the FFD strategy, designed to overcome the FFD limitations in performing localized control points refinements. Finally, Section 3.4 detail the hierarchical pattern designed for the mesh handling tool realized, highlighting its potentialities and selective approach.

3.1 Identification of free-body rigid movements

In this section we aim at dealing with those fluid-structure interaction problems involving, as structure, free-bodies (not constrained ones): in such cases, during the simulation process, the structure would be subjected to both deformations and rigid movements due to the effect of fluid forces acting on structural external surfaces. Within Chapter 2 we dealt with shape parametrization techniques within FSI moving mesh problems, in order to update the fluid mesh configuration to match the computed structural one at the interface: in that case, the displacement to be described was related only to elastic deformations. In such a case involving free-bodies, in order to solve efficiently the mesh motion problem, we should be able to know the entity of both the two displacement components (rigid and deformative). Therefore we can split, within that framework, the structural displacement in two different components: a global large one, related to structural rigid translations and/or rotations and, the other, due to small elastic deformations. If we denote by $\varsigma \in \Omega^S$ the total structural displacement field, we can proceed performing its separation as it follows:

$$\varsigma = \varsigma_R + \varsigma_D \qquad (3.1)$$

where:

(*i*) ς_R, represents the structural rigid movements;

(*ii*) ς_D, indicates the small deformative component.

It is important to remark the fact that we have classified as *large* and *small*, the rigid and elastic components of the displacement, respectively, since the first involves a perturbation of the whole structural domain, while, the second, generally leads to localized shape deformations. Here, our goal, is to set up a methodology in order to identify both the rigid translations and rotations, while, for the recovery of the deformative component, it will be employed the shape parametrization techniques introduced within Chapter 2. In particular we can observe that the motion of the fluid mesh, when considering free-bodies, can be known a priori (e.g. the case of imposed forced oscillations of an airfoil, aileron deflections, etc.) or it can be the output returned by the structural sub-system: here we focus on the latter case, that leads to an identification problem of the structural translations and rotations. With respect to Figure 3.1 and 3.2, we illustrate the idea of separation of the global displacement in two different components: starting from an initial reference configuration of a commercial aircraft, we obtain the final one by simulating aeroelastic deformations and, then, applying a rigid pitch rotation.
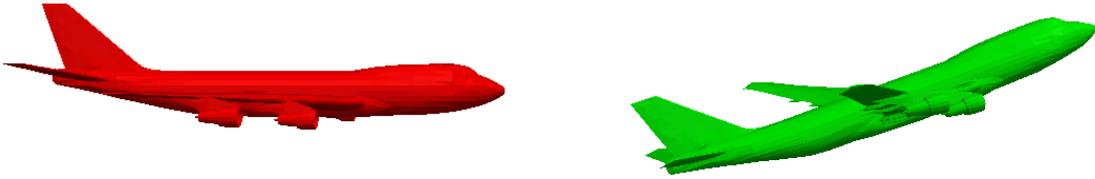
Figure 3.1: Initial (left-red) and target (right-green) shape configurations of a commercial aircraft.
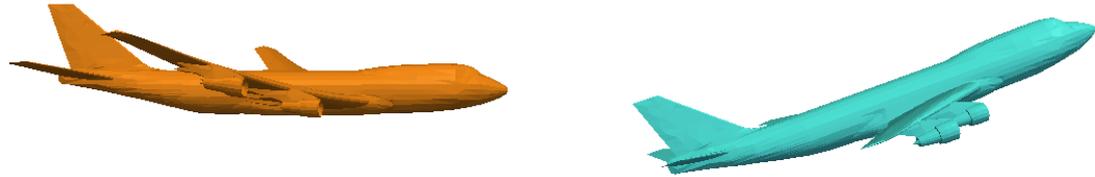
Figure 3.2: Visualization of the different displacement components defining the final configuration: deformative (left-orange) and rigid (right-blue) components.

At this point, after this brief introduction, it is possible to detail the methodology employed to recover both the rigid translations and rotations values. To this end we can denote by $\hat{\Omega}_{h,i}^S \in \mathbb{R}^{n_s \times 3}$ and by $\hat{\Omega}_{h,i+1}^S \in \mathbb{R}^{n_s \times 3}$ the structural mesh, with n_s indicating the number of structural grid nodes, at the generic time-steps t_i and t_{i+1}, respectively, expressed into a global fixed reference coordinate system $\mathcal{T}(\mathbf{X})$, with $\mathbf{X} = [\mathbf{x}^T, \mathbf{y}^T, \mathbf{z}^T]$, as illustrated in Figure 3.3. On the basis of the knowledge of the initial and final structural domain grids configurations, we would be able to find the unknown values of both the rigid translations,
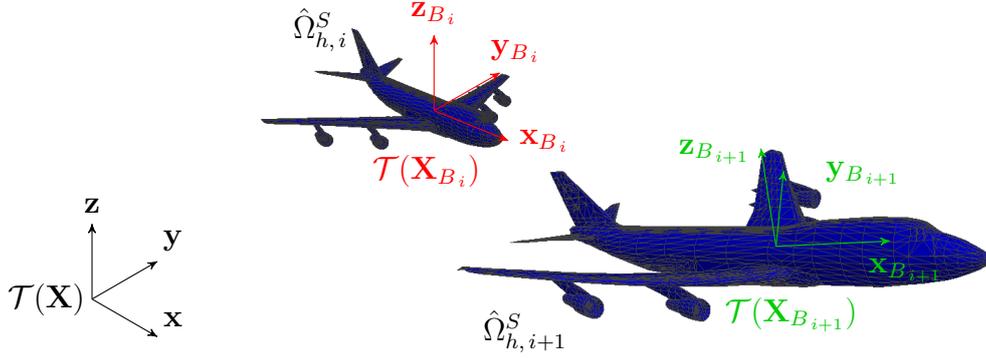
Figure 3.3: Structural mesh configurations at different time steps of the FSI simulation.

$\mathbf{s} = \{s_x,\ s_y,\ s_z\}^T$, and rotations, $\boldsymbol{\theta} = \{\theta_x,\ \theta_y,\ \theta_z\}^T$, that occurred between t_i and t_{i+1}. Considering both the structural discretized configurations, $\hat{\Omega}^S_{h,i}$ and $\hat{\Omega}^S_{h,i+1}$, it is possible to compute their barycenter, with respect of the coordinate system $\mathcal{T}(\mathbf{X})$, as it follows:

$$\mathbf{x}_B = \frac{\sum_{k=1}^{n_s} \mathbf{x}_k}{n_s}, \tag{3.2}$$

where $\{\mathbf{x}_k\}_{k=1}^{n_s}$ represent the structural grid nodes coordinates. Evaluating the expression (3.2) for both the two configurations, $\hat{\Omega}^S_{h,i}$ and $\hat{\Omega}^S_{h,i+1}$, we get their barycenter, $\mathbf{x}_{B_i}$ and $\mathbf{x}_{B_{i+1}}$, that represent the origin of the body coordinate systems $\mathcal{T}(\mathbf{X}_{B_i})$ and $\mathcal{T}(\mathbf{X}_{B_{i+1}})$. At this point, in order to identify the values of the rigid translations, it is possible to subtract the coordinates of the computed barycenters as it follows:

$$\mathbf{s} = \mathbf{x}_{B_{i+1}} - \mathbf{x}_{B_i}. \tag{3.3}$$

Once computed the values of rigid translations, we can proceed evaluating the three unknown rotations $\boldsymbol{\theta} \in \mathbb{R}^3$. First of all, we translate $\hat{\Omega}^S_{h,i+1}$, with its coordinate system $\mathcal{T}(\mathbf{X}_{B_{i+1}})$, of a quantity -$\mathbf{s}$, in order to match the origins of both the systems $\mathcal{T}(\mathbf{X}_{B_{i+1}})$ and $\mathcal{T}(\mathbf{X}_{B_i})$: in this way we isolate the rigid rotation term from the translation one. It is possible to evaluate the inertia matrices, related to $\hat{\Omega}^S_{h,i}$ and $\hat{\Omega}^S_{h,i+1}$, as it follows:

$$\mathbf{I} = \sum_{k=1}^{n_s} m_k \begin{bmatrix} (y_k^2 + z_k^2) & -x_k y_k & -x_k z_k \\ -x_k y_k & (x_k^2 + z_k^2) & -y_k z_k \\ -x_k z_k & -y_k z_k & (x_k^2 + y_k^2) \end{bmatrix}, \tag{3.4}$$

being x_k, y_k and z_k the coordinates of each structural grid nodes, defined with respect to $\mathcal{T}(\mathbf{X}_{B_i})$, and $\{m_k\}_{k=1}^{n_s} = 1$. Thanks to the evaluation of (3.4) for both $\hat{\Omega}_i^S$ and $\hat{\Omega}_{i+1}^S$, we obtain $\mathbf{I}_i$ and $\mathbf{I}_{i+1}$. In order to compute the orientations of each axis belonging to $\mathcal{T}(\mathbf{X}_{B_{i+1}})$ and $\mathcal{T}(\mathbf{X}_{B_i})$, we can solve the eigenproblems related to $\mathbf{I}_{i+1}$ and $\mathbf{I}_i$, respectively, defined as it follows:

$$
\begin{aligned}
\mathbf{I}_{i+1}\,\mathbf{y}_{i+1} &= \lambda_{i+1}\,\mathbf{y}_{i+1}, & (3.5)\\
\mathbf{I}_i\,\mathbf{y}_i &= \lambda_i\,\mathbf{y}_i, & (3.6)
\end{aligned}
$$

where the eigenvectors obtained as solution of the eigenproblem (3.5) represent the orientation of the axes $\mathbf{X}_{B_{i+1}} = [\mathbf{x}_{B_{i+1}}^T, \mathbf{y}_{B_{i+1}}^T, \mathbf{z}_{B_{i+1}}^T]$, while the others obtained from (3.6) represent the orientation of $\mathbf{X}_{B_i} = [\mathbf{x}_{B_i}^T, \mathbf{y}_{B_i}^T, \mathbf{z}_{B_i}^T]$. In order to solve this problem we have employed the LAPACK-DSYEV routine of the Teuchos Trilinos package [58], that is generally used to compute eigenvalues and eigenvectors for symmetrical matrices, by implementing a QR based factorization. It is possible, at this point, to establish through the introduction of a rotation matrix $\mathbf{R}(\boldsymbol{\theta})$, one relation between each axes pair (e.g. between $\mathbf{x}_{B_{i+1}}$ and $\mathbf{x}_{B_i}$). In detail, the orthogonal rotation matrix corresponding to a clockwise left-handed rotation, with Euler angles θ_x, θ_y, θ_z and **x-y-z** convention, is given by:

$$
\mathbf{R}(\boldsymbol{\theta})_{xyz} = \begin{bmatrix} \cos\theta_y\cos\theta_z & -\cos\theta_x\sin\theta_z + \sin\theta_x\sin\theta_y\cos\theta_z & \sin\theta_x\sin\theta_z + \cos\theta_x\sin\theta_y\cos\theta_z \\ \cos\theta_y\sin\theta_z & \cos\theta_x\cos\theta_z + \sin\theta_x\sin\theta_y\sin\theta_z & -\sin\theta_x\cos\theta_z + \cos\theta_x\sin\theta_y\sin\theta_z \\ -\sin\theta_y & \sin\theta_x\cos\theta_y & \cos\theta_x\cos\theta_y \end{bmatrix}.
$$

Thanks to the hypothesis of small angles (in detail, $-10° \leq \alpha \leq 15°$, that leads to $\cos\alpha \simeq 1$, $\sin\alpha \simeq \alpha$), we get the expression of the linearized rotation matrix $\tilde{\mathbf{R}}(\boldsymbol{\theta})$, defined as it follows:

$$
\tilde{\mathbf{R}}(\boldsymbol{\theta}) = \begin{bmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix}, \tag{3.7}
$$

where we need no more to indicate the dependency related to the order of the applied rotations, due to the hypothesis of small angles adopted. Now we can write three relations, one for each axes pair, in order to find the rigid rotation angles occurred between the FSI simulation time-steps t_i and t_{i+1}. Thanks to the definition of the linearized rotation matrix is it possible to write:

$$
\begin{Bmatrix} x_{B\,i+1_{(1)}} \\ x_{B\,i+1_{(2)}} \\ x_{B\,i+1_{(3)}} \end{Bmatrix} = \begin{bmatrix} 1 & -\theta_z & \theta_y \\ \theta_z & 1 & -\theta_x \\ -\theta_y & \theta_x & 1 \end{bmatrix} \begin{Bmatrix} x_{B\,i_{(1)}} \\ x_{B\,i_{(2)}} \\ x_{B\,i_{(3)}} \end{Bmatrix}, \tag{3.8}
$$

where $x_{B\,i_{(1)}}$, $x_{B\,i_{(2)}}$ and $x_{B\,i_{(3)}}$ represent the components of the computed eigenvector $\mathbf{x}_{B,i}$ projected on the global reference coordinate system $\mathcal{T}(\mathbf{X})$. By rewriting (3.8) as it follows:

$$\begin{Bmatrix} x_{B\,i+1_{(1)}} \\ x_{B\,i+1_{(2)}} \\ x_{B\,i+1_{(3)}} \end{Bmatrix} = \underbrace{\begin{bmatrix} 0 & x_{B\,i_{(3)}} & -x_{B\,i_{(2)}} \\ -x_{B\,i_{(3)}} & 0 & x_{B\,i_{(1)}} \\ x_{B\,i_{(2)}} & -x_{B\,i_{(1)}} & 0 \end{bmatrix}}_{\mathbf{A}} \begin{Bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{Bmatrix} + \begin{Bmatrix} x_{B\,i_{(1)}} \\ x_{B\,i_{(2)}} \\ x_{B\,i_{(3)}} \end{Bmatrix}, \quad (3.9)$$

is possible to observe easily that, in order to get the unknown angles of rotations, we need to invert the matrix $\mathbf{A}$, which is clearly singular, since $\det(\mathbf{A}) = 0$. Therefore, we overcome such a problem by proceeding as it follows: since the global reference coordinate system $\mathcal{T}(\mathbf{X})$ is defined by:

$$\mathbf{X} = \begin{bmatrix} \mathbf{x}^T, \mathbf{y}^T, \mathbf{z}^T \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.10)$$

we can evaluate, by employing (3.8) applied to each axes pair the rotation angles, $\boldsymbol{\theta}_{i+1}$, between $\mathcal{T}(\mathbf{X}_{B_{i+1}})$ and $\mathcal{T}(\mathbf{X})$:

$$\mathbf{X}_{\mathbf{B_{i+1}}} = \begin{bmatrix} 1 & -\theta_{z\,i+1} & \theta_{y\,i+1} \\ \theta_{z\,i+1} & 1 & -\theta_{x\,i+1} \\ -\theta_{y\,i+1} & \theta_{x\,i+1} & 1 \end{bmatrix} \mathbf{X} \quad (3.11)$$

and, in the same way, we can find $\boldsymbol{\theta}_i$, between $\mathcal{T}(\mathbf{X}_{B_i})$ and $\mathcal{T}(\mathbf{X})$:

$$\mathbf{X}_{\mathbf{B_i}} = \begin{bmatrix} 1 & -\theta_{z\,i} & \theta_{y\,i} \\ \theta_{z\,i} & 1 & -\theta_{x\,i} \\ -\theta_{y\,i} & \theta_{x\,i} & 1 \end{bmatrix}.\mathbf{X} \quad (3.12)$$

In both (3.11) and (3.12) the angle values are given through simple algebraic relations, that, in detail, for the case related to $\boldsymbol{\theta}_i$, are given by:

$$\begin{aligned} \theta_{x\,i} &= & z_{B\,i_{(2)}} &= & -y_{B\,i_{(3)}}, \\ \theta_{y\,i} &= & -z_{B\,i_{(1)}} &= & x_{B\,i_{(3)}}, \\ \theta_{z\,i} &= & y_{B\,i_{(1)}} &= & -x_{B\,i_{(2)}}, \end{aligned} \quad (3.13)$$

At this point, once we have found both $\boldsymbol{\theta}_{i+1}$ and $\boldsymbol{\theta}_i$, it is possible to compute the values of rigid rotations $\boldsymbol{\theta}$, between $\mathcal{T}(\mathbf{X}_{B_{i+1}})$ and $\mathcal{T}(\mathbf{X}_{B_i})$, as it follows:

$$\boldsymbol{\theta} = \boldsymbol{\theta}_{i+1} - \boldsymbol{\theta}_i \quad (3.14)$$

In order to better assess both the accuracy and the limits of applications of the

method, we performed several tests by considering, as structure, a commercial airplane. In detail, we started considering the aircraft whose barycenter is centered in $\mathbf{x}_{B_i} = \{0, 0, 0\}^T$, and we imposed different combinations of both translations and rotations, to be recovered by the algorithm proposed, and simulated aeroelastic deformations (as shown in Figure 3.3). Our goal consists in recovering, as best as possible, the rigid components of the total applied displacement. With respect to Figure 3.6, it is possible to appreciate the main results obtained:

(a) Translations to be recovered.   (b) Rotations to be recovered.
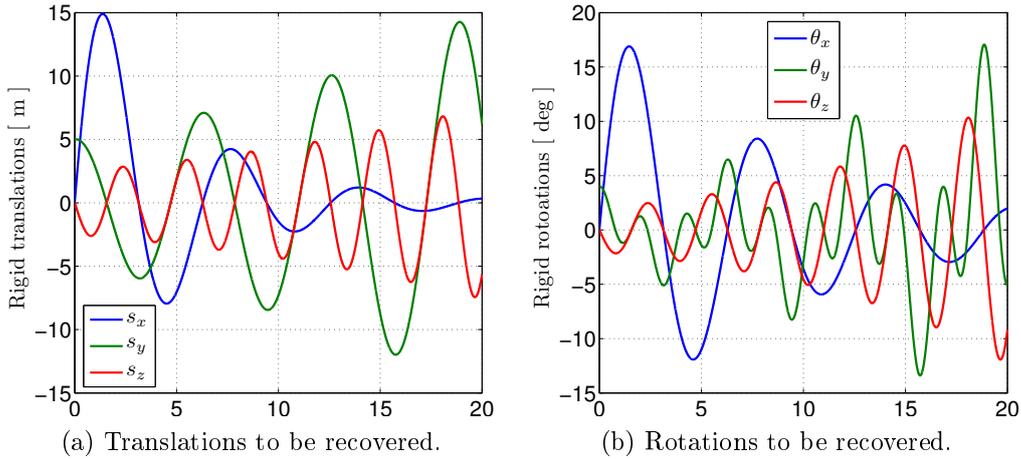
Figure 3.4: Representation of both the translations and rotations imposed simultaneously.

(a) Translations recovered.   (b) Error.
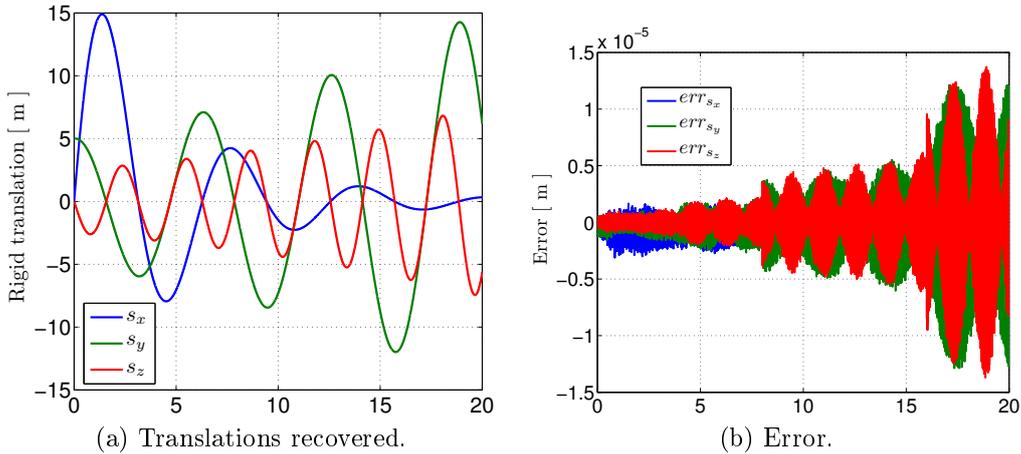
Figure 3.5: Visualization of the gained results related to the translations recovery.

Figures 3.5 and 3.6 demonstrate the quality of the strategy proposed: for all the possible combinations of rigid translations the estimation of the error is absolutely trivial while, concerning rotations, the results are in agreement with

theory due to the great results obtained with the exception of large angles (even if the errors are of the order of one degree).

(a) Comparison of θ_x.

(b) Error.

(c) Comparison of θ_y.

(d) Error.

(e) Comparison of θ_z.
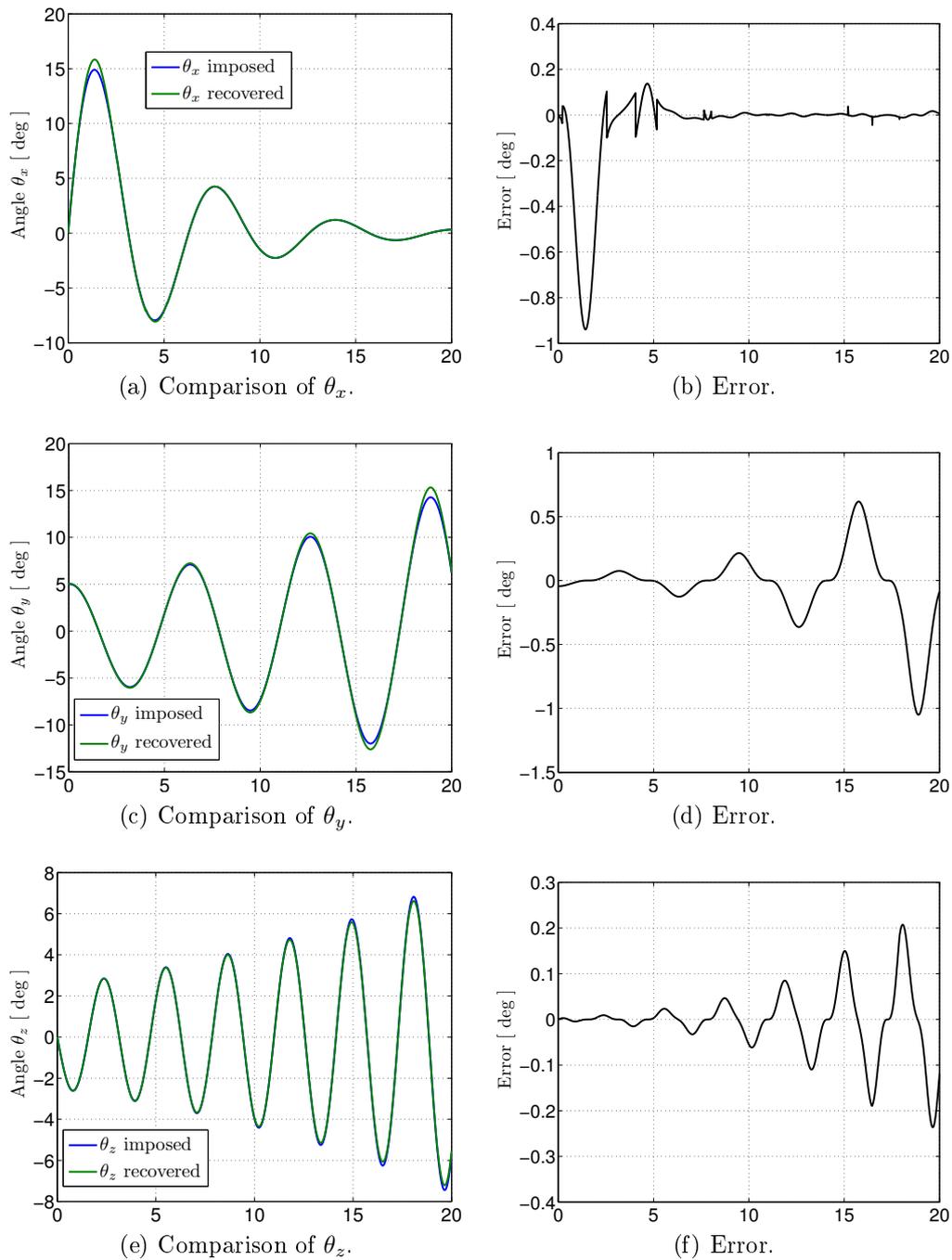
(f) Error.

Figure 3.6: Results obtained within rotations recovery: on the left-hand-side we make a comparison between the angles imposed and the ones estimated, while, on the right-hand-side, we report their corresponding errors.

3.2   Adaptive selection of RBF interpolation sites

In this section we focus our attention on the choice of the interpolation sites, within RBF method, to handle moving mesh problems for FSI. As introduced in Chapter 2, RBF technique relies on the use of a (possible small) set of control points whose displacements induce the shape deformation. Here, we aim at proposing a standard methodology dealing with the identification of the best set of parameters to be taken into account, in order to employ RBF strategy within the dynamic mesh handling. Since now, we have assumed that both the number, and locations, of control points should be fixed a priori by the user, in a problem-dependent way: in this Section we propose an innovative methodology that could be employed in order to overcome such a limitation. We have seen, with respect to Section 2.3, that the RBF accuracy and computational costs, show a quite significant sensitivity related to the number of parameter adopted for the analysis: in such a situation, when any physical criteria is adopted to choose the control points setting, an incorrect choice of both the number of parameters and their sites, would drive to both a loss in computational time, when we adopt excessive number of control points, and, suddenly, inaccurate results, when the parameters employed are not enough to describe the shape deformations. Therefore, in the FSI framework, the main questions consist in how to represent shapes, and their deformations, using shape parametrization strategies, like RBF. In literature a wide range of studies is present [113], related to the identification of the best control points setting to be adopted: they are based on the idea of finding a possible best set by mean of several screening tests and sensitivity analysis performed considering a progressive growing number of parameters, and, in a trivial way, finally they consider the one showing best results in terms of accuracy of the solution computed. Such an approach, as stated before, is clearly characterized by a certain lack in precision and uncertainty, since the parameters pattern is established without any idea of the real (physical) shape deformations that will occur during the FSI simulation process. Moreover, in that way, is not possible to generalize the whole process, since the control points setting would change in a very problem-dependent way. On the basis of this introduced background, here, we propose a high-fidelity approach designed to better assess the choice of both the number of control points as well as their locations, through a physical approach suited for FSI problems. The first issue to be faced, in order to outline such a procedure, deals with the estimation related to the shape deformations (structural displacements) that will occur during the FSI simulation process. If we will be able to find a set of likelihood shapes to be captured by the RBF method, the problem of finding the best parameters set, would be solved through an optimization algorithm, by placing control points in order to recover the target shapes. Thus, our main idea relies on estimating these target shapes through the computation of the *structural eigenmodes*: such an approach lies on the physical principle that, during the FSI simulation, the structural behavior could be well-

represented, and approximated, by the superposition of its normal modes. In fact, in structural dynamics or aeroelasticity problems, such an approach is employed when we are interested in approximating the spatial displacement $\varsigma(\mathbf{x}, t)$ of the structure using much fewer degrees of freedom than is necessary to solve the discretized equations of motion resulting from a finite-element or finite-difference approximation. Intuitively, the first N_m eigenmodes represent good candidates for a general basis to approximate the displacement since, with respect to (3.15), the exact displacement can *always* be obtained as a linear combination of these eigenmodes:

$$\varsigma(\mathbf{x}, t) = \sum_{i=1}^{N_m} \mathbf{N}_i(\mathbf{x}) \, \mathbf{q}_i(t), \tag{3.15}$$

where $\mathbf{q}_i(t)$, $i = 1, \ldots, N_m$, represent the time-harmonic part of the solution and $\mathbf{N}_i(\mathbf{x})$ are the so-called eigenmodes that determine completely the spatial dependency of the displacement. Indeed, by introducing a finite-element discretization for the structural continuum, in order to obtain the eigenmodes $\mathbf{N}_i(\mathbf{x})$, we have to solve the following generalized eigenproblem:

$$\omega^2 \, \mathbf{M} \, \varsigma = \mathbf{K} \, \varsigma, \tag{3.16}$$

where $\mathbf{M} \in \mathbb{R}^{n_s \times n_s}$ and $\mathbf{K} \in \mathbb{R}^{n_s \times n_s}$ represent the mass and stiffness matrices of the structure, while ω indicate the eigenvalues of the structural system, whose corresponding eigenmodes are collected within $\mathbf{N}_i(\mathbf{x})$. In the framework of our FSI mesh motion problem, one of the issue that could arise is related to the number of structural eigenmodes to be taken into account for our procedure: the choice of the bandwidth N_m is highly problem-dependent and, in detail, is related to the purpose of our FSI analysis. Generally, when we are interested in the global behavior of rigid structural system, small values of N_m could be taken into account, while, when dealing with FSI problems high-flexible systems, N_m should increase. In fact, as illustrated in Figure 3.7, it is possible to rank eigenmodes with respect to the frequency ω: the more frequency increases, the more oscillating (in space) eigenmodes would be evaluated. A common adopted technique, within that issue, relies in the computation of an initial greater number of N_m eigenmodes in order to perform, then, a sensitivity analysis evaluating the contribution of each of them to the global response. In that way it will be possible to truncate our basis, identifying the bandwidth of interest. Once we have introduced the way adopted to find the target shapes to be described, we can proceed detailing the strategy employed to find both the number of control points and their locations. For that purpose we employed an optimization method, based once again on a Greedy algorithm [97], that has been previously introduced in Section 2.1.1. After we have identified the N_m target shapes, the aim is to find the best set of control points able to describe all these configurations (eigenmodes). Generally, as stated before, higher shape complexities are related to the high frequency eigenmodes, thus, within our procedure, we start identifying the

parameters pattern able to describe the shape related to the N_m-th eigenmode. In this way, we can formalize the problem for the 3D case as it follows: we denote by $\hat{\Omega}_0$ the initial structural discretized domain (e.g. Figure 3.7a), and by $\mathcal{V}$ the space of the admissible structural configurations, whose elements are represented by the eigenmodes $\{\hat{\Omega}_k\}_{k=1}^{N_m}$, such that $\dim(\mathcal{V}) = N_m$. Our goal consists in finding the best set of parameters, $\mathbf{\Pi}(\mathbf{x}) \in \mathbb{R}^{N_c \times 3}$, able to accurately capture all the shape configurations belonging to $\mathcal{V}$.

(a) Plate.

(b) First mode, $\omega = 4.21E + 5$ Hz.

(c) Second mode, $\omega = 1.05E + 6$ Hz.

(d) Third mode, $\omega = 1.75E + 6$ Hz.

(e) Fourth mode, $\omega = 2.44E + 6$ Hz.

(f) Fifth mode, $\omega = 3.07E + 6$ Hz.
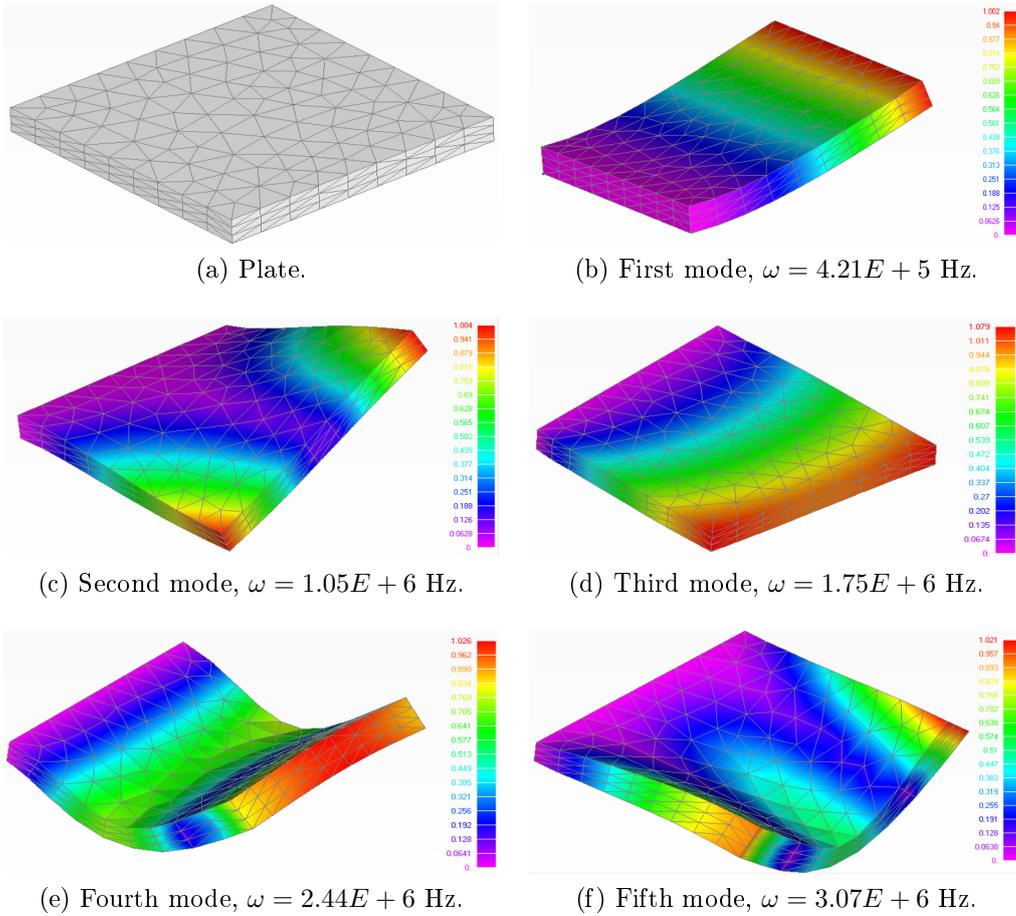
Figure 3.7: Eigenmodes and generalized displacements computed for a one edge clamped steel plate ($1\,m \times 1\,m \times 0.1\,m$). Analysis performed using *Msc-Nastran*, a commercial Finite Element code.

In this way, we can start identifying the set of control points able to recover firstly the most complex shape (as an example see Figure 3.10), in the space $\mathcal{V}$, and then, to verify if it is sufficient to describe the others. In detail, our identification procedure will take into account as first target shape $\hat{\Omega}_{N_m}$, and, then, the others $\{\hat{\Omega}_k\}_{k=N_m-1}^{1}$. We can focus now on the use of the Greedy algorithm [97] in order to find $\mathbf{\Pi}(\mathbf{x})$. As we have introduced, we have two iteration loops: the

outer that considers the eigenmodes computed in a decreasing order (from $\hat{\Omega}_{N_m}$ to $\hat{\Omega}_1$), and an inner one, that for a fixed target shape to be captured finds both the number of control points and their sites. We can now deeply analyze the inner loop devoted to the identification of $\mathbf{\Pi}(\mathbf{x})$. At each iteration of the inner loop, namely for each i_k, we firstly compute the nodal values of the difference between $\hat{\Omega}_k$ and $\hat{\Omega}_0^{(i_k)}$, then, we identify the locations, $\tilde{\mathbf{X}} = \{\tilde{\mathbf{x}}^T, \tilde{\mathbf{y}}^T, \tilde{\mathbf{z}}^T\}$ with $\tilde{\mathbf{X}} \in \mathbb{R}^{6 \times 3}$, of the nodes where the differences in the x, y and z directions reach their positive maximum and negative minimum values. At that point, thanks to the interpolatory behavior of RBF [20], it will be possible to locate, there, a maximum of six control points: in such a way we can control the local error between $\hat{\Omega}_k$ and $\hat{\Omega}_0^{(i_k)}$. We have indicated with $\hat{\Omega}_0^{(i_k)}$ the modified initial structural configuration, which is obtained thanks to control points displacements imposed in order to match $\hat{\Omega}_0^{(i_k)}(\tilde{x}, \tilde{y}, \tilde{z})$ with $\hat{\Omega}_k(\tilde{x}, \tilde{y}, \tilde{z})$. At each iteration, once we add few control points, we can evaluate the cost functional $\mathcal{J}^{(i_k)}$, that is defined as it follows:

$$\mathcal{J}^{(i_k)} = \left\| \frac{\hat{\Omega}_0^{(i_k)} - \hat{\Omega}_k}{\hat{\Omega}_0^{(1)} - \hat{\Omega}_k} \right\|_{L_2}. \tag{3.17}$$

Such a functional, at each iteration i_k, represents a non-dimensional measure of how much the error between the shape recovered $\hat{\Omega}_0^{(i_k)}$ and $\hat{\Omega}_k$ has been reduced with respect to its initial value, namely between $\hat{\Omega}_0^{(1)}$ and $\hat{\Omega}_k$. We considered as stopping criteria, for the inner loop, a minimum value, TOL, to be reached by the cost functional, such that $\mathcal{J}^{(i_k)} < TOL$. At this point, since we have reached convergence, we have a set of identified parameters, $\mathbf{\Pi}(\mathbf{x}) \in \mathbb{R}^{N_c \times 3}$, that will be used as starting control points set for the outer loop, considering the others eigenmodes. Generally, since the first eigenmode considered, whose corresponding eigenvalue is the highest in the bandwidth selected, is the most complex from a geometrical point of view, the set of control points found for $k = N_m$ should be able to describe, also, the others. Nevertheless, if necessary, when iterating on the outer loop, it will be possible to enrich the set of parameters previously found, in order to refine the accuracy of our basis made up of control points. With respect to Figure 3.8, a schematic outline of the strategy proposed is represented, applied to the example of the plate that we have previously introduced. It is important to remark the fact that such a strategy should be employed *offline*, namely before running the FSI simulation: in fact, to start the algorithm proposed, we need to compute the structural eigenmodes within a bandwidth of interest; as result, the algorithm provides a set of control points to be exploited for the dynamic mesh handling within the FSI simulation, namely *online*. The approach proposed let us overcome the curse of dimensionality and the arbitrary related to the choice of parameters since we do not consider, as RBF control points, all the nodes of the fluid-structure interface, but only a small part of them.
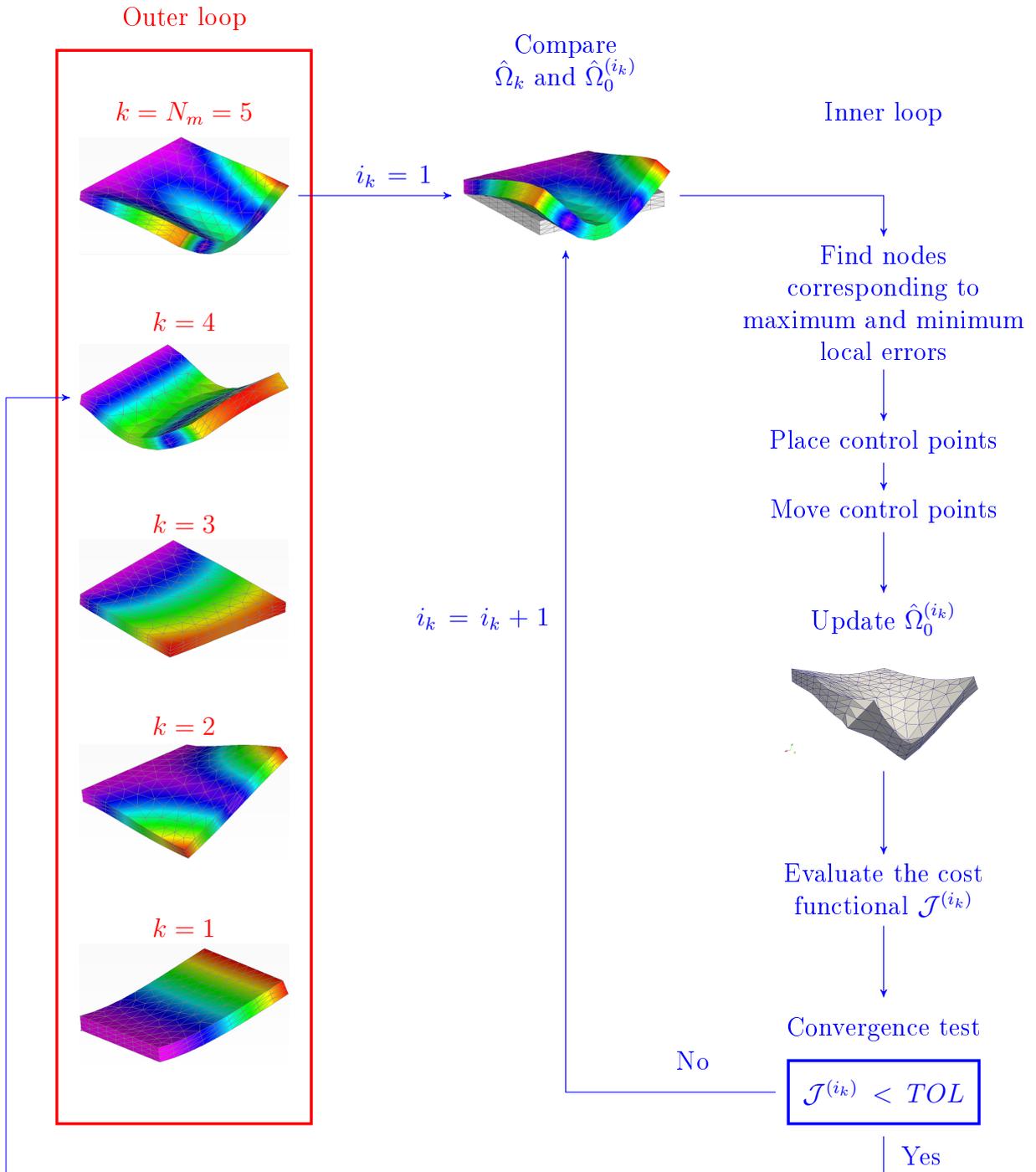
Figure 3.8: Schematic outline of the algorithm proposed: we start recovering the N_m-th eigenmode within the bandwidth selected, namely $\hat{\Omega}_{N_m}$ such that $k = N_m$, and, when convergence has been reached, we switch to $\hat{\Omega}_{N_m-1}$ and we repeat the inner loop.

3.2.1   Some applications of 3D examples

In this section we propose two examples that show the application of the algorithm proposed, suited for RBF technique, to find both the number of control points and their locations in order to describe accurately a family of different shapes (structural eigenmodes). In detail, we will perform these test applied to both the case of the plate introduced in Section 3.2 and of a beam model of the aircraft Fokker F28 [50]. Concerning the first one, we have identified in our bandwidth of interest (chosen to test only the behavior of the algorithm) the first fifth eigenmodes, depicted in Figure 3.7. As stated before, within our procedure, we start considering the fifth mode, and, then, we will shift to the lower ones: in Figure 3.9 it is possible to understand the behavior of the cost functional $\mathcal{J}^{(i_5)}$ versus the iterations number, within the recovery of the fifth eigenmode. In this case, in order to show the consistence of the method, we did not fixed any tolerance, and, for that reason, the lowest value of $\mathcal{J}^{(i_5)}$ is reached when the number of control points introduced by the algorithm equals the number of structural grid nodes ($n_s = 444$).

(a) Cost functional.   (b) Control points number
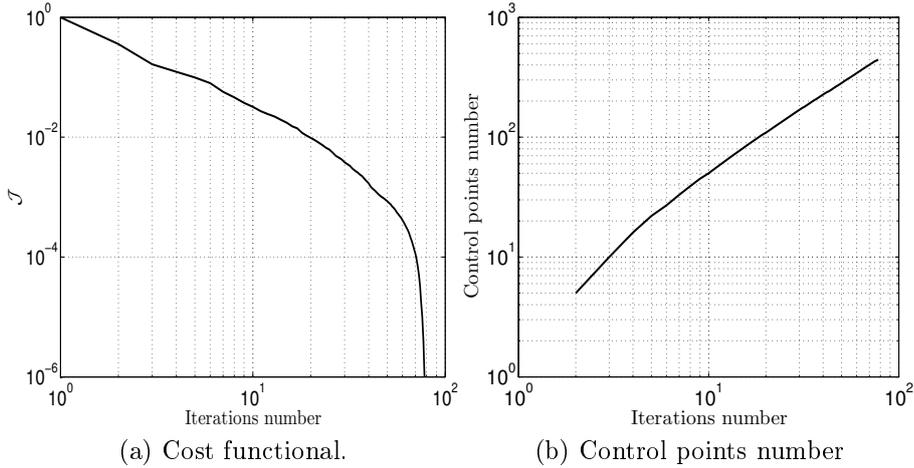
Figure 3.9: Results of convergence gained for the recovery of the fifth eigenmode.

Furthermore, once we have fixed the tolerance to $TOL = 0.05$ we obtained, as result, a number of control points $N_C = 49$. Such result has been obtained by forcing control points to be placed over the fluid-structure interface. With respect to Figure 3.10, it is possible to appreciate the control points sites over the initial undeformed shape and the corresponding achievable deformations, related to different inner loop iteration numbers i_5 (within the recovery of the fifth eigenmode). As outlined in Figure 3.8, with the identified set of parameters $\mathbf{\Pi}(\mathbf{x}) \in \mathbb{R}^{N_C \times 3}$ it is possible to iterate on the other lower eigenmodes: we have to check if such set $\mathbf{\Pi}(\mathbf{x})$ is still able to describe $\{\hat{\Omega}_k\}_{k=N_m-1}^1$. Within Table 3.1 we report the results obtained:

Eigenmode	Number of iterations i_k	$\mathcal{J}^{(i_k)}$ final value
$\hat{\Omega}_4$	1	0.0471
$\hat{\Omega}_3$	1	0.0114
$\hat{\Omega}_2$	1	0.0239
$\hat{\Omega}_1$	1	0.0100

Table 3.1: Results obtained applying the algorithm to $\{\hat{\Omega}_k\}^1_{k=N_m-1}$.

(a) $i_5 = 4$, $N_C^{(4)} = 21$.

(b) Deformed structural configuration $\hat{\Omega}_0^{(4)}$.

(c) $i_5 = 8$, $N_C^{(8)} = 44$.

(d) Deformed structural configuration $\hat{\Omega}_0^{(8)}$.

(e) $i_5 = 10$, $N_C^{(10)} = 49$.

(f) Deformed structural configuration $\hat{\Omega}_0^{(10)}$.
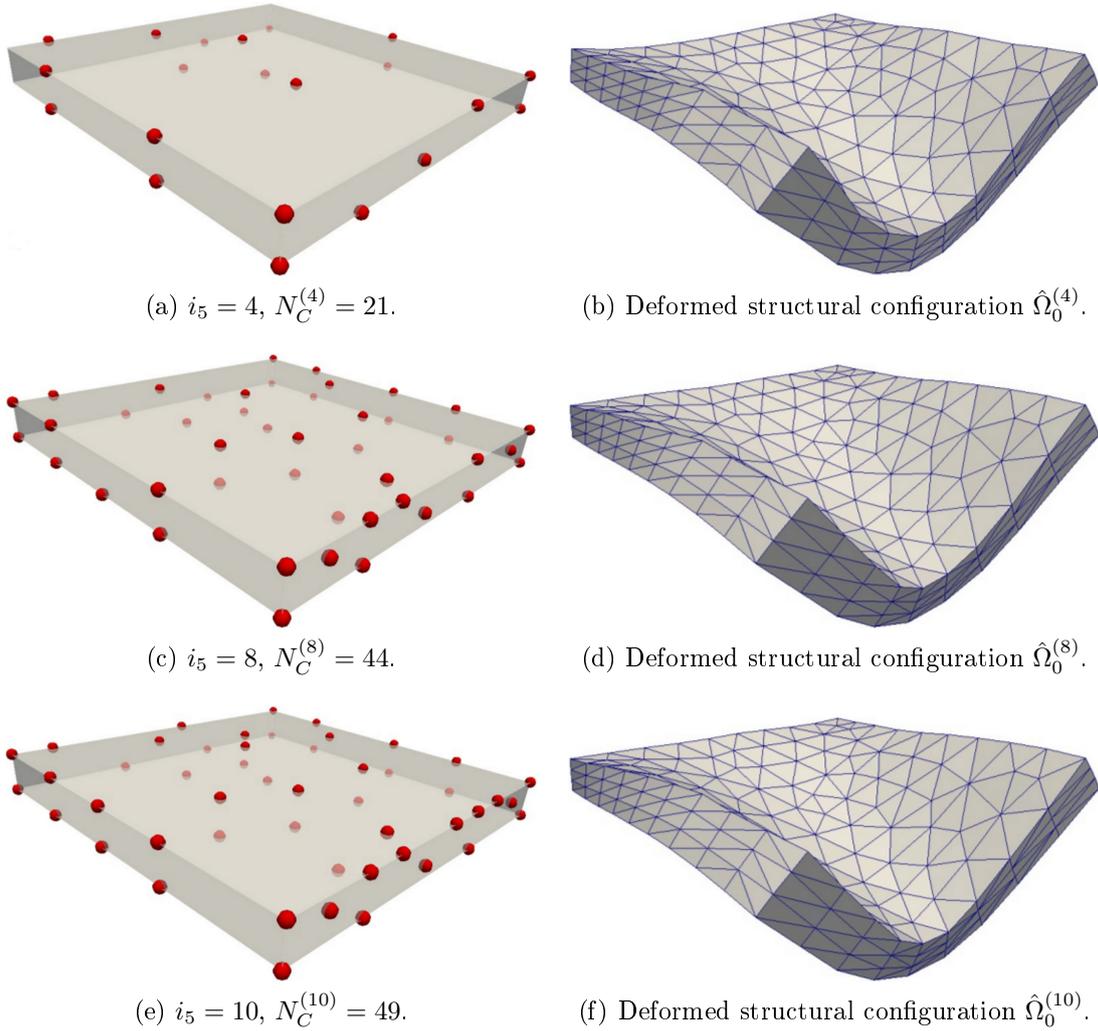
Figure 3.10: Snapshots taken from the recovery of the 5-th eigenmode: interpolation sites (left-hand-side) and achievable deformations (right-hand-side).

Table 3.1 shows that the initial set of control points, built to describe the fifth eigenmode, has also been able to represent accurately the lower ones: in that way

no further parameters needed to be adopted. At the basis of such a result, with respect to Table 3.1, we have a number of iterations $i_k = 1$ for each eigenmode considered (for $k = 4, \ldots, 1$), since the set of parameters gained to recover $\hat{\Omega}_5$ was also sufficient to capture the others $\{\hat{\Omega}_k\}_{k=4}^{1}$. As second example, we have tested the behavior of the algorithm proposed in relation to the eigenmodes computed for a commercial aircraft depicted in Figure 3.11, namely the Fokker F28.

(a) Fokker 28.   (b) Beam model.
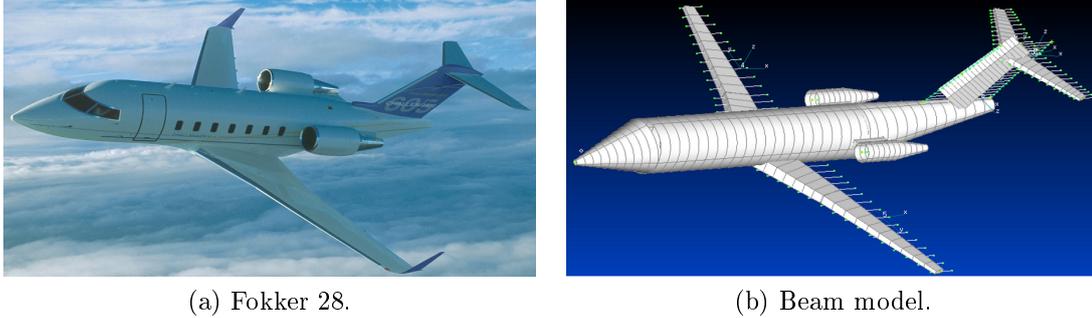
Figure 3.11: Representation of the aircraft considered and its corresponding beam model.

In order to set up the beam model, the main prerequisites that we have followed consist in a correct representation of the mass distribution, of the inertial properties and of the dimensions of the real aircraft [50]. On the basis of these considerations, all the structural and nonstructural mass contributions have been taken into account: in order to appreciate the high-fidelity of the model, with respect to Table 3.2, the comparison between the weights, of both each structural subcomponent belonging to the beam model and the real aircraft, is reported.

Component	Beam model [kg]	Real aircraft [kg]
Wing	2576.96	2579.09
Horizontal tail	407.424	407.5
Vertical tail	393.095	392.81
Fuselage	3927.85	3927.81
Nacelle	569.40	569.40

Nonstructural mass	Weight [kg]
Fuel system	600.04
Hydraulic system	80.57
Electrical system	875.90
Environmental control system	445.10
Avionics	764.62
Flight control system	501.98
Furniture	1940.76
Oils	169.26

Table 3.2: Distribution of nonstructural masses and weights validation.

At that point, in order to apply the algorithm developed within the choice of RBF interpolation sites, we computed the first 15 eigenmodes of the aircraft: within such bandwidth selected we will start recovering the 15-th mode and, then, we will shift to lower ones.

(a) Undeformed configuration.   (b) 7-th eigenmode, $\omega = 3.68\,Hz$.

(c) 12-th eigenmode, $\omega = 4.74\,Hz$.   (d) 15-th eigenmode, $\omega = 7.28\,Hz$.
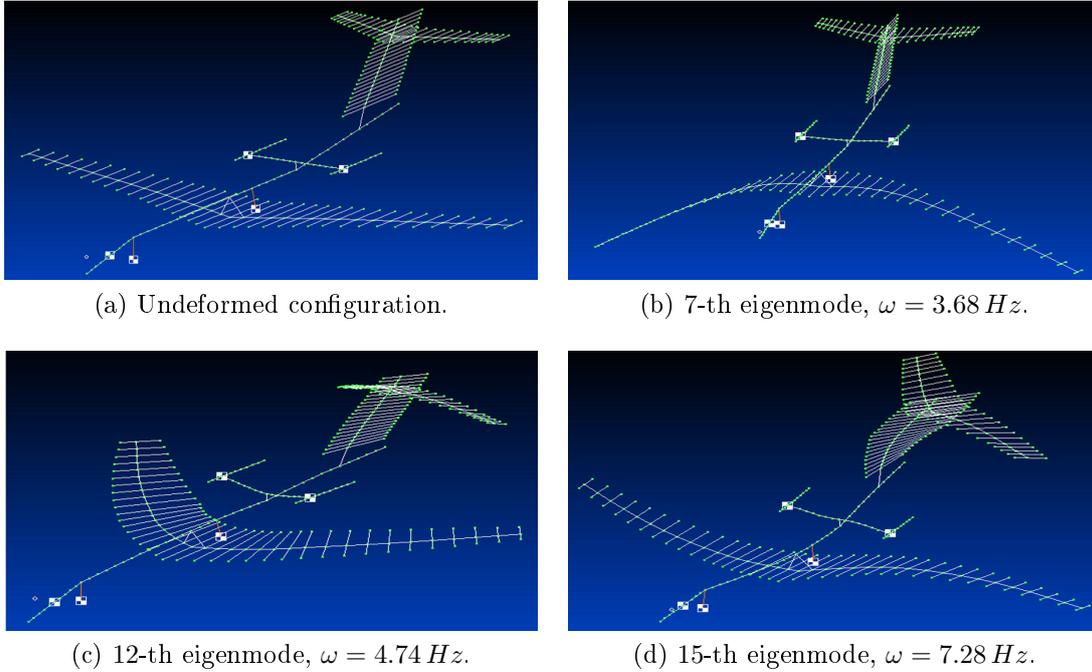
Figure 3.12: Representation of some eigenmodes, computed with *Msc-Nastran*.

(a) Cost functional.   (b) Control points number.
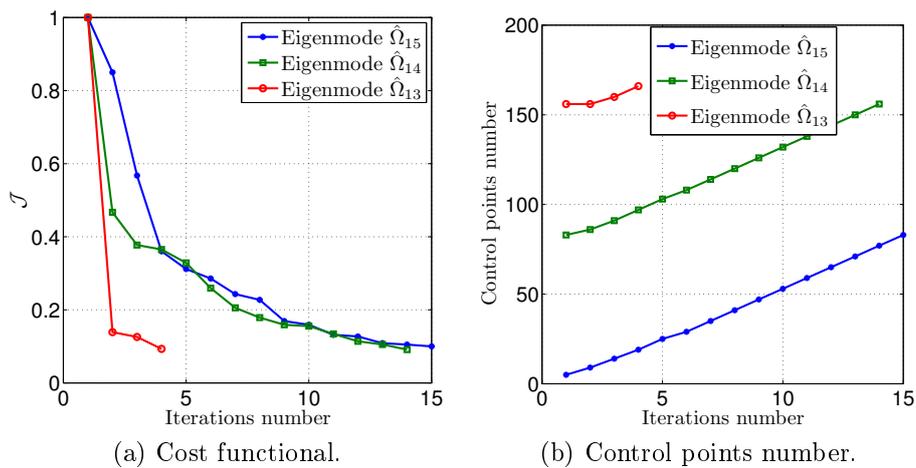
Figure 3.13: Convergence properties within the description of the first 15-th, 14-th and 13-th aircraft eigenmodes.

For this example, unlike the previous one concerning the plate, the set of control points adopted to recover the 15-th eigenmode was not sufficient, as it is shown in Figure 3.13, to describe the others. In fact, since we fixed the tolerance to $TOL = 1E-1$, the control points found for $k = 15$, was not able to capture the 14-th mode, resulting in a necessary enrichment of the space of parameters $\mathbf{\Pi}(\mathbf{x})$. In Figure 3.13b is it possible to appreciate the growing number of control points related to both the iterations of the inner loop (on a fixed curve) and to the outer one (in that case we refer to different curves). Thus, as shown in Figure 3.14, we illustrate the growing number of parameters, as well as their locations, across different iterations considering the different eigenmodes $\{\hat{\Omega}_k\}_{k=15}^{13}$: the control points found for $k = 15$ are illustrated with red bullets, the ones added when dealing with the 14-th eigenmode are represented in green, while the last ones, identified for $k = 13$, in blue.

(a) $k = 15$.

(b) $k = 14$.

(c) $k = 13$.
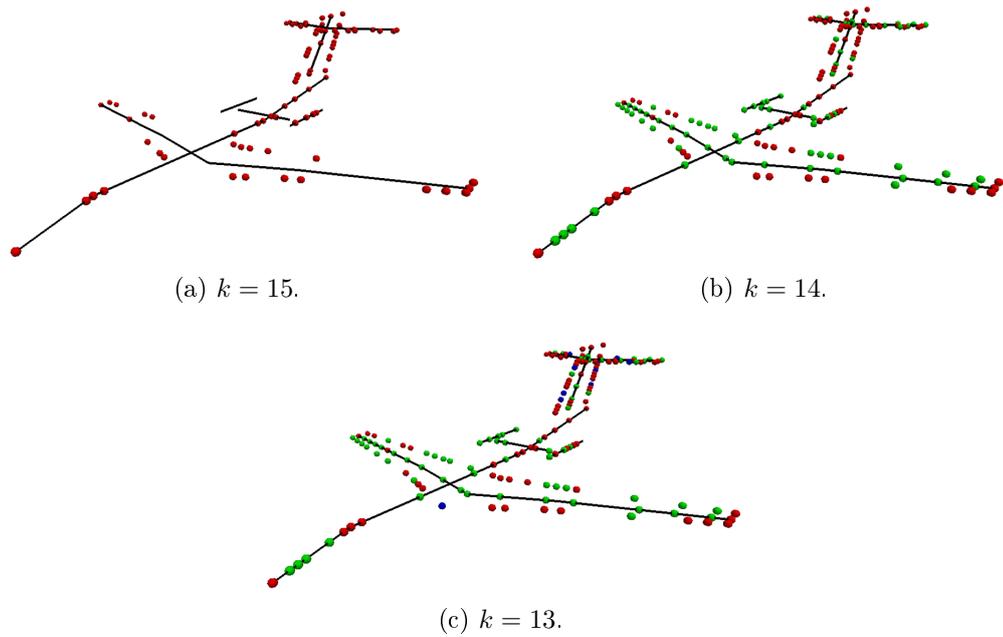
Figure 3.14: Visualization of the growing number of control points over the frame of the aircraft configuration.

Once we have iterated on the 13-th eigenmode, with the identified set of $N_c = 161$ control points, we were able to describe accurately the lower first 12 ones since, at their corresponding first inner loop iteration, the cost functional had values which were lower than the fixed prescribed tolerance.

3.3 Domain-Decomposition approach for free-form deformations

In this Section we focus our attention on free-form deformation techniques, and, in detail, we will propose a numerical strategy that will allow us to overcome some of its limitations. As we stated throughout Section 2.1, the most severe restriction related to its use within dynamic mesh handling lies in the impossibility of performing local control points refinements since parameters must be located into a regular lattice embedding the solid object to be represented. The main idea, in order to tackle such a problem, relies in considering the whole fluid domain as composed by different elementary patches: thanks to such decomposition we can build for each patch, or some of them, different localized FFD maps in order to manage mesh deformations within each single patch. To better assess the method introduced and to demonstrate its efficiency, we will test this strategy within the 2D problem introduced within Section 2.5:

(a) Initial configuration.   (b) Final configuration.
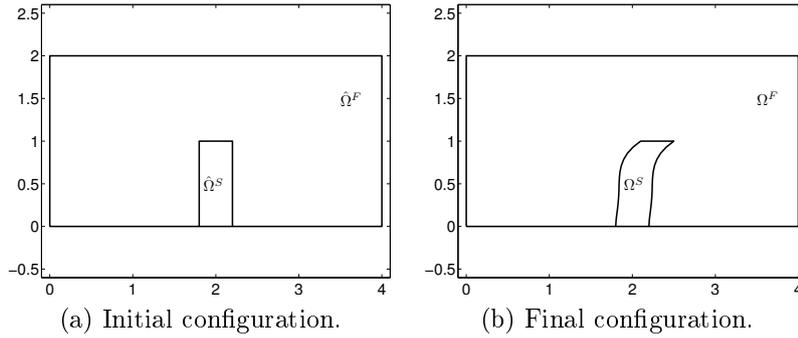
Figure 3.15: Representation of the 2D problem setting.

In order to describe such a structural deformation, we firstly decomposed the undeformed fluid domain in four different patches, $\hat{\Omega}^{F(i)}$, with $i = 1, \dots, 4$, such that $\hat{\Omega}^{F(1)} \cup \hat{\Omega}^{F(2)} \cup \hat{\Omega}^{F(3)} \cup \hat{\Omega}^{F(4)} = \hat{\Omega}^{F}$, as illustrated in Figure 3.16, and, then, we built a local FFD map only for $\hat{\Omega}^{F(3)}$.
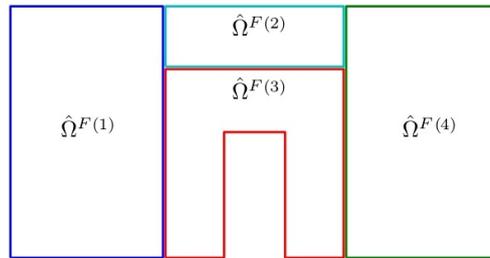
Figure 3.16: Domain decomposition adopted for the example to be solved.

In such a way, the problem consists in applying the Algorithm 2.1 to the patch selected. Within this approach, since we mapped through FFD technique only a portion of the global domain, we are now allowed to adopt a smaller number of control points thanks to the fact that they are located in crucial sites, in the nearness of the structural domain. Thus, we show the results obtained by means of the patch-approach proposed, and, also, we compare them with respect to the ones gained in Section 2.5, where we considered the FFD map extended to the whole fluid domain $\hat{\Omega}^F$.
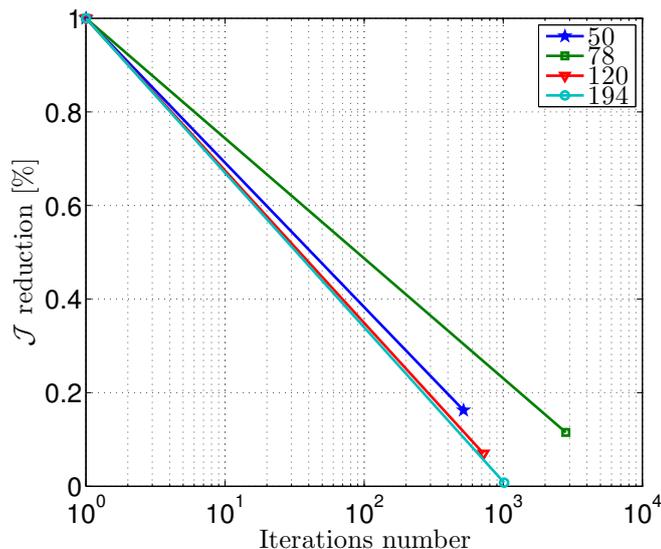
Figure 3.17: Results obtained considering different numbers of control points placed within the patch $\hat{\Omega}^{F\,(3)}$.

With respect to Figure 2.12a, where the results gained considering the FFD map extended to the whole domain are reported, it is possible to appreciate the accuracy of the approach, achieved by means of smaller control points number. In this way it will be possible on one hand to reduce the number of parameters involved within the FFD strategy, on the other we improve the quality of the fluid mesh deformed. Nevertheless, it is necessary to remark the fact that in order to achieve satisfactory results related to the grid quality, the user-defined patch adopted, namely $\hat{\Omega}^{F\,(3)}$, should not be excessively small, in the sense that it should not limit the space where the mesh will get deformed. Concerning the measure of the mesh quality, within Figure 3.18 we show the scaled Jacobian values of the deformed fluid mesh as well as its related histogram. Generally an element is said to be "inverted" (or not valid) [127] if its related scaled Jacobian is negative or equal to zero. On the basis of the achieved results it is possible to state the good quality of the deformed mesh (corresponding to high scaled Jacobian values) since this measure keeps values within the acceptable range $[\sim 0.4,\ 1]$. Furthermore, as indicated by the histogram, only a small number of elements has

a quality indicator that equals the lower bound 0.4.
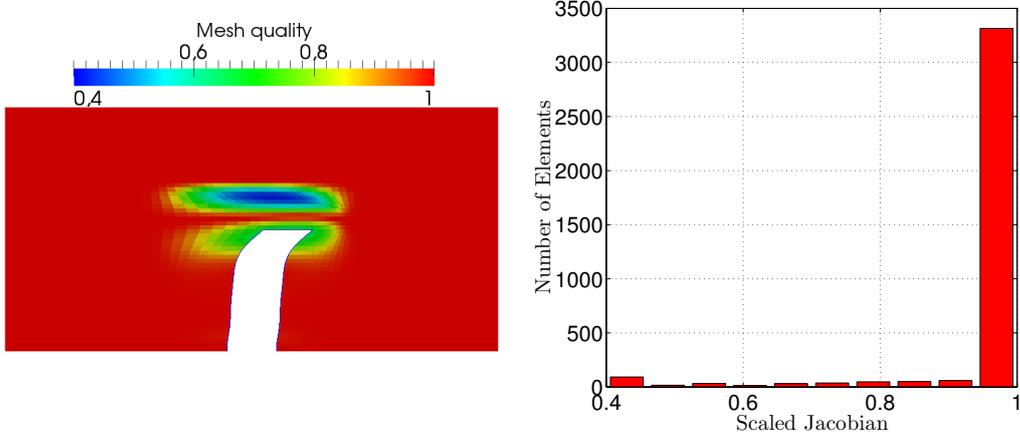
(a) Scaled Jacobian of the deformed fluid mesh, gained with a number of control points $N_c = 194$.

(b) Histogram representing the occurrences (number of elements) within each mesh quality bin.

Figure 3.18: Mesh quality measure (scaled Jacobian) obtained adopting the patch-approach proposed.

Finally, it is possible to make a comparison of the computational effort between the results achieved thanks to the patch-approach proposed and the ones related to the classical FFD strategy (which consider the map extended to the whole fluid domain). With respect to Table 3.3, the main results towards such a comparison are illustrated: in detail it is possible to highlight the reduction, gained thanks to the approach proposed, of the total time needed to perform the fluid mesh deformation:

FFD on $\hat{\Omega}^F$

N_C	$\mathcal{J}$ reduction [%]	Computational time [s]
25	59.3	21.2
126	87.1	52.0
234	89.83	16
456	94.52	45
588	96.70	100

FFD on $\hat{\Omega}^{F\,(3)}$

N_C	$\mathcal{J}$ reduction [%]	Computational time [s]
25	65.4	0.38
50	84.1	0.51
78	88.5	3.38
120	93.2	0.86
194	99.3	1.20

Table 3.3: Comparison of the accuracy and computational costs between the FFD map applications extended to the whole fluid domain (left-hand-side), and only to one of its patch (right-hand-side).

On the basis of the results obtained, we could review the importance of FFD techniques used to handle mesh motion problems thanks to the new proposed patch-approach, based on domain decomposition, since it drives, as shown in Table 3.3, to clear reductions in the computational effort and it allows the possibility of making control points refinements.

3.4 Outline of the hierarchical pattern

In this section we detail the hierarchical strategy developed, within the FSI framework, in order to deal with both the fluid-structure interface and the mesh motion problems. With respect to the previously introduced background, and in particular, on the basis of the several tests performed, we outline a standard path suitable for shape parametrization techniques. Within sections 2.4 and 2.5 we highlighted both the limitations and the advantages of IDW, RBF and FFD strategies: in detail, we tried to identify for each of them, the application field best suited for their use. In fact, it could be possible to highlight, concerning IDW, its great potentiality and flexibility related to the handling of the mesh motion problem, instead of the fluid-structure interface one. RBF has demonstrated to be globally robust in dealing with both these problems, while FFD, even if it represents an efficient tool within shape parametrization to handle the mesh motion problem, its adoption would lead to high computational costs. Nevertheless, we have shown that, thanks to the patch-approach within the free-form deformation strategy, we can both reduce its computational effort and we make FFD able to describe local deformations, even if actually it has to be used in a problem-dependent way, since the domain decomposition has not been automatized for treating arbitrary domain configurations yet.
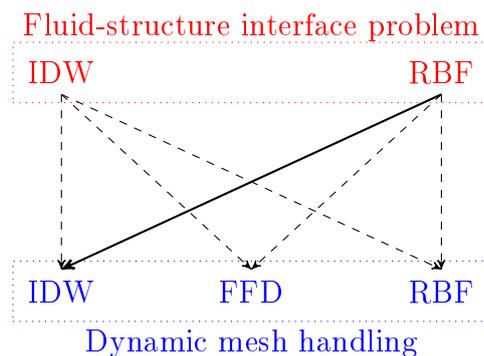
Figure 3.19: Possible combinations of solution strategies to be employed. The bold arrow indicates the combination adopted: IDW for the dynamic mesh handling and RBF to face the fluid-structure interface problem.

The mesh tool developed, written in C++ code, has been organized to be fully

flexible, thanks also to the adoption of an object-oriented framework to guarantee the possibility to choose the solution strategy desired, among the ones illustrated in Figure 3.19.
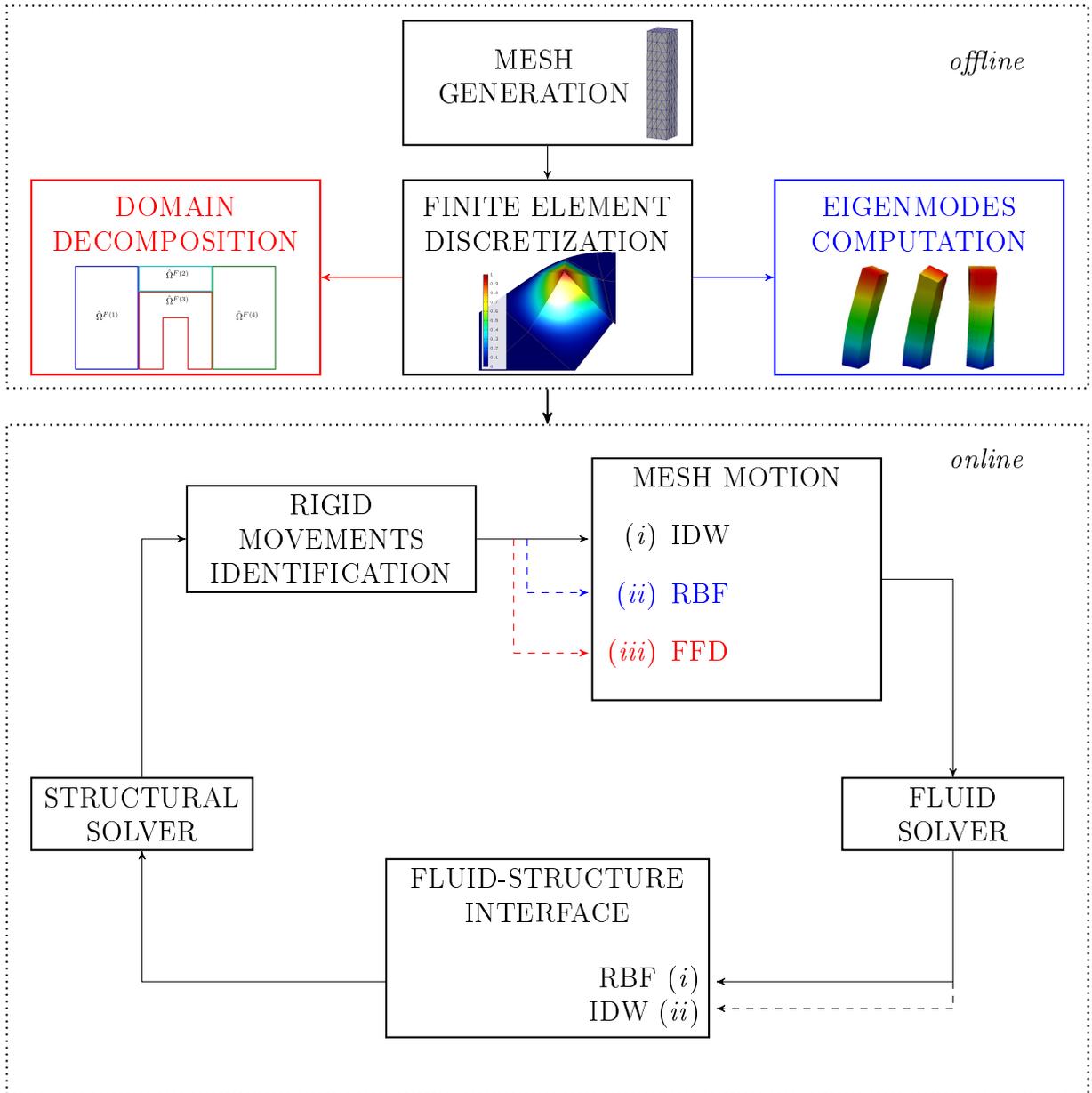
Figure 3.20: Framework of the mesh toolbox developed: we have highlighted with two different blocks those operations to be performed online and offline. Moreover, by different colors we have clarified the aim of both the eigenmodes computation and the domain decomposition techniques, related either to the adaptive choice of interpolation sites within RBF, and to the FFD strategy, respectively.

The use of both the RBF technique, to deal with fluid-structure interface problems, and the IDW, for the dynamic mesh handling, has proven to represent the most efficient and robust choice to be adopted. With respect to Figure 3.20, it is possible to appreciate the design of the approach proposed where we have split in two different stages the whole FSI simulation process. We have highlighted the offline stage, related to all these actions to be performed before running the FSI simulation, and the online one, dealing with all these procedures to be adopted when the simulation takes place (iteratively). Concerning the offline stage, once generated the computational grids for both the fluid and the structure domains, and their Finite Element discretizations, we underline the possibility of adopting, either the adaptive selection of RBF interpolation sites, when this method would be employed to handle the mesh motion problem, or the domain decomposition technique, related to the patch-approach, when we employ the FFD method.
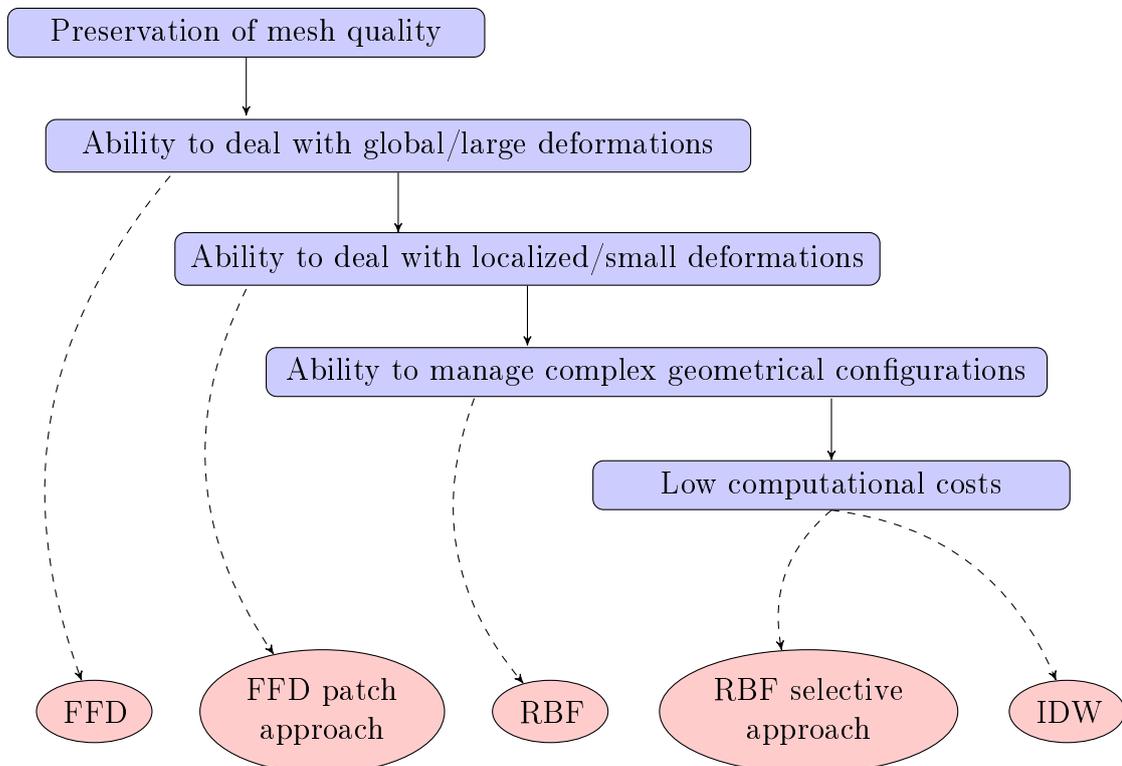
Figure 3.21: Flux diagram which summarizes the main selection criteria (blue rectangles) followed to choose the technique to be adopted within the dynamic mesh handling.

The default path followed by our hierarchical approach is the one characterized by the sub-sequential blocks linked by black arrows: as default choice for the fluid-structure interface the RBF method is adopted, while, for the mesh motion, the IDW technique, respectively. Nevertheless, if desired, it will be possible to

switch easily from one strategy to the other one, as illustrated in Figure 3.20 by the dashed colored lines. Concerning the dynamic mesh handling, with respect to the flux diagram illustrated in Figure 3.21, since IDW has shown a good compromise between the accuracy of the achievable results and its computational costs, this multivariate interpolation strategy has been chosen to be adopted in order to move and update the fluid mesh configuration after structural deformations. Indeed, although FFD and RBF have demonstrated to represent a valid alternative to IDW, they still suffer from some limitations. Concerning FFD techniques, as we have detailed in Section 2.1.1, it would not be adopted in those cases involving both localized deformations as well as complex domains configurations since it is not interpolatory, leading, during the mesh motion process, to an unacceptable presence of gaps and overlaps over the fluid-structure interface (as shown in Figure 2.11). On the other hand, when the FSI problems involves localized deformations related to simple geometrical domains, the patch-approach developed for FFD would represent a valid choice to be taken into account. Regarding the RBF strategy, although it actually represent a robust and powerful choice (in terms of accuracy of the achievable mesh quality), we have demonstrated that, with respect to IDW technique, it is more expensive from a computational costs point of view. Nevertheless, by using a selective approach for the identification of both the number of control points and their sites, the costs would get slightly reduced. In order to treat the data transfer across non-matching grids, the RBF strategy, based on the adoption of multi quadratic biharmonic splines has been privileged with respect to IDW, since its use would guarantee an high-quality and robustness of the interpolation process. At the basis of the possibility of dealing within FSI problems with free-bodies we have included, as shown in Figure 3.20, a block dealing with the estimation of both the rigid translations and rotations as output from the structural solver, in order to be able to update accurately the fluid mesh configuration with respect to such movements too.

Chapter 4

Fluid-Structure Interaction modeling

In this Chapter we focus on the mathematical modeling of the FSI problem. We introduce the equations that describe both the fluid and the structure physics, as well as the numerical strategies adopted to solve the coupled FSI problem in a monolithic way [35]. In Section 4.1 we introduce the reference frameworks that will be employed in order to derive the FSI model, in detail the Eulerian, the Lagrangian and the Arbitrary Lagrangian Eulerian (ALE) formulations. Then, through Sections 4.2 and 4.3, the equations governing the physics of both the fluid and the structure are derived. At the basis of the mathematical model achieved, it will be possible to further proceed by introducing, in Section 4.4, the time discretization of both the fluid and structure problems by means of the Geometric-Convective Explicit scheme, that will be fully detailed. Throughout Section 4.5 the weak formulation of the FSI problem, as well as its space discretization, are proposed. Moreover we will derive the algebraic system describing the fully-discrete FSI problem as well as its solution strategy, with respect to Section 4.6, adopted within the LifeV environment. We anticipate that for the numerical solution of the fluid-structure problem we adopted LifeV, that is a state of the art finite element library providing many solution schemes for the FSI problem. Moreover, concerning the time discretization and the discrete form of the coupled FSI problem we will follow the approach proposed in [28, 86], while for the algorithm part (Section 4.7) we refer to [29].

4.1   Eulerian, Lagrangian and ALE formulations

The choice of an appropriate reference system represents a key-aspect when dealing with the mathematical modeling and the numerical simulation of continuum mechanics problems characterized by moving boundaries. In general, one of the most commonly adopted formulation within Fluid-Structure Interaction

problems makes use of two classical descriptions of motion: the so called *Arbitrary Lagrangian-Eulerian* (ALE) formulation relying on both the *Lagrangian* and *Eulerian* kinematics descriptions. Such a formulation, introduced by [87], was developed in an attempt to combine the advantages of the above classical kinematical descriptions, and to minimize their respective drawbacks as far as possible. At this point, before introducing the main aspects that characterize the ALE frame, we briefly introduce both the Lagrangian and the Eulerian descriptions of motion. Lagrangian formulation is commonly adopted within structural mechanics, and its description is based on the fact that each individual node of the computational grid follows the material particle during motion, such that it allows an easy tracking of free surfaces and interfaces between different materials (as shown in Figure 4.1). Moreover, it has also been extensively used within the treatment of materials with history-dependent constitutive relations [23]. On the other hand, purely Lagrangian methods typically result in severe mesh distortion and the consequence is ill conditioning of the element stiffness matrix leading to mesh lockup or entanglement, such that the need for expensive re-meshing arises. In the framework of Eulerian description, which is the most commonly used strategy within fluid dynamics, the computational mesh is fixed while the continuum moves with respect to the grid. Such a formulation allows the possibility of handling with relative ease large distortion, while its weakness relies at the expense of a precise definition of the interface, as well as the resolution of flow details. At the basis of the drawbacks of purely Lagrangian and purely Eulerian descriptions, the ALE formulation has been developed in order to combine the best features of both classical formulations. Within such a description the nodes of the mesh can be either moved to follow the continuum in a Lagrangian manner, or be held fixed as in the Eulerian description, or even be moved in in some arbitrarily specified way to give a continuous rezoning capability. At the basis of this freedom in moving the computational mesh, the possibility of handling great distortions of the continuum relies, such that we overcome both the limitations of purely Lagrangian formulation and we improve the resolution afforded by a purely Eulerian approach. In Figure 4.1, we depict a schematic representation of the Eulerian, Lagrangian and ALE descriptions. In order to introduce the mathematical models lying at the basis of such formulations, we need to introduce the notation that we will adopt hereafter in this chapter. The position of a material point in the reference (Lagrangian) configuration is denoted by $\hat{\cdot}$, while if we refer to the current (Eulerian) configuration we will omit $\hat{\cdot}$. Furthermore, as previously introduced, we assume that the reference configuration is at its natural state, that is when the Cauchy stresses are zero everywhere. Thus, let us denote by $\hat{\Omega} \subset \mathbb{R}^3$ and $\Omega_t \subset \mathbb{R}^3$ the reference and current domains configurations, respectively, and by $\hat{\mathbf{x}} \in \hat{\Omega}$ and $\mathbf{x} \in \Omega_t$ the coordinates of two points belonging to the reference and current configurations.
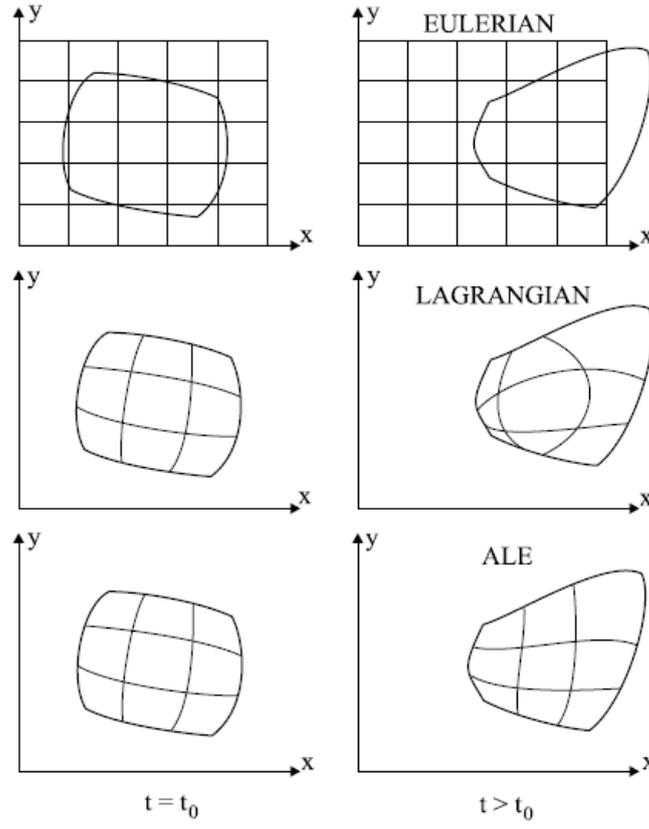
Figure 4.1: Representation of Eulerian, Lagrangian and ALE formulations [19], respectively.

In order to describe the kinematics of the continuum media, we adopt a frame of reference with respect to the $\hat{\mathbf{x}}$ coordinate if we refer to a purely Lagrangian formulation, on the other hand, by adopting $\mathbf{x}$, we rely on an Eulerian description. We can further define a function ϕ representing the motion, being $\phi : \hat{\Omega} \times \mathbb{R}^+ \rightarrow \Omega_t \subset \mathbb{R}^3$, such that $\phi(\hat{\mathbf{x}}, t)$ represents a deformation evolving in time. At the basis of such a definition, the relations between Lagrangian and Eulerian frames of reference read:

$$
\begin{aligned}
\mathbf{x} &= \phi(\hat{\mathbf{x}}, t), \\
\hat{\mathbf{x}} &= \phi^{-1}(\mathbf{x}, t),
\end{aligned}
\tag{4.1}
$$

where we assume that ϕ^{-1} always exists due to the hypothesis of dealing with continuum media, such that during motion the domains would not be lacerated. The Jacobian of such a transformation $J = (\partial \mathbf{x}/\partial \hat{\mathbf{x}})$ could also be interpreted as the relative volume increment between current and reference configurations, namely: $Jd\hat{\Omega} = d\Omega$. Moreover, we can define the so-called deformation gradient

F, whose expression reads as:

$$\mathbf{F} = \hat{\nabla}\phi, \quad \text{component-wise} \quad \mathbf{F}_{i,j} = \frac{\partial \phi_i}{\partial x_j}. \tag{4.2}$$

As previously introduced, the description of the mechanics of continuum media with respect to $\hat{\mathbf{x}}$, or *material point*, is called Lagrangian: within such a formulation a scalar or vectorial field $\hat{V}$ is defined Lagrangian as well, since it is defined in $\hat{\Omega}$. By adopting an Eulerian frame of reference we need to define a fixed control volume V_C, such that $V_C \subseteq \Omega_t$, $\forall t \in (0,T) \subset \mathbb{R}$. Furthermore, a scalar or vectorial field V is called Eulerian since it is defined in Ω_t; thus, the Eulerian counterpart of the Lagrangian vector field is $V(\mathbf{x}, t)$, for $\mathbf{x} \in V_C \subset \Omega_t$. Within an Eulerian approach, since $\mathbf{x} = \phi(\hat{\mathbf{x}}, t)$, to express the total time derivative of $V(\mathbf{x}, t)$ we need to use the chain rule:

$$D_t V(\mathbf{x}, t) = \frac{dV(\mathbf{x}, t)}{dt} = \frac{\partial V}{\partial \mathbf{x}} \frac{\partial \phi}{\partial t}(\phi^{-1}(\mathbf{x}, t), t) + \frac{\partial V}{\partial t}. \tag{4.3}$$

On the other hand, the total time derivative within a Lagrangian formulation coincides with the partial derivative, since $\hat{\mathbf{x}}$ is expressed in the reference configuration:

$$\frac{d\hat{V}(\hat{\mathbf{x}}, t)}{dt} = \frac{\partial \hat{V}(\hat{\mathbf{x}}, t)}{\partial t}. \tag{4.4}$$

Thus, we can define the velocity of the material point $\mathbf{u}$ with respect to a Lagrangian and an Eulerian formulations as it follows:

- Lagrangian: $\hat{\mathbf{u}} = D_t \phi(\hat{\mathbf{x}}, t)$;

- Eulerian: $\mathbf{u} = \partial_t \phi(\phi^{-1}(\mathbf{x}, t), t)$.

A this point, in order to account for the displacement of the fluid domain we need to introduce the ALE description, which is obtained by modifying the Eulerian one in such a manner that the fixed control volume is no longer constant but it follows the material particles of the moving boundaries. On this idea the ALE formulation relies. To this end we define a reference control volume $\hat{\Omega}_{\mathcal{A}} \subset \mathbb{R}^3$, such that the ALE map reads:

$$\mathcal{A} : \hat{\Omega}_{\mathcal{A}} \times \mathbb{R}^+ \to \hat{\Omega}_{\mathcal{A}} \subset \Omega_t. \tag{4.5}$$

At the basis of the above definition we have that, for each $t > 0$, $\mathcal{A}$ maps the reference control volume to the arbitrary domain $\Omega_{\mathcal{A}}$ in the deformed configuration. Thus, for a given function which is defined in the current configuration,

$g : \Omega_t \times (0, T) \to \mathbb{R}$, we indicate by $\hat{g} = g \circ \mathcal{A}^{-1}$ its counterpart within the reference one $\hat{\Omega}$. Furthermore, it is possible to define the ALE time derivative:

$$\left.\frac{\partial g(\mathbf{x}, t)}{\partial t}\right|_{\hat{\mathbf{x}}} = \frac{\partial \hat{g}(\mathbf{x}, t)}{\partial t} \circ (\mathcal{A}(\mathbf{x}))^{-1}. \tag{4.6}$$

We can also introduce the domain velocity $\mathbf{w}$, whose expression reads as:

$$\mathbf{w} = \left.\frac{d\mathbf{x}}{dt}\right|_{\hat{\mathbf{x}}} = \frac{\partial \mathcal{A}}{\partial t} \circ (\mathcal{A}(\mathbf{x}))^{-1}. \tag{4.7}$$

Finally, it is possible to introduce one of the key-ingredients that we will further adopt in order to derive the fluid equations, namely the Reynolds' transport theorem (also known as the Leibniz-Reynolds' transport theorem) [98] that yields the following relationship between Lagrangian and Eulerian perspectives:

$$\frac{d}{dt}\int_{\Omega_t} f \, d\Omega_t = \int_{\hat{\Omega}} \left[\frac{\partial f}{\partial t} + \nabla \cdot (f\mathbf{u})\right] d\hat{\Omega} = \int_{\hat{\Omega}} \frac{\partial f}{\partial t} \, d\hat{\Omega} + \int_{\partial\hat{\Omega}} f \, \mathbf{u} \cdot \mathbf{n} \, d\gamma, \tag{4.8}$$

where we indicated with $\partial\hat{\Omega}$ the boundary delimiting the reference domain configuration.

4.2 Derivation of the fluid equations

In this Section, following the approach of [28], we report the conservation laws for mass and momentum with respect to the Eulerian and ALE frames of reference. In detail we will deal with an incompressible Newtonian fluid model, such that the conservation laws that we will derive will describe the Navier-Stokes (NS) equations as well. In order to derive the model that describes the physics of the fluid, we prefer to firstly introduce a general formulation for the conservation of both a scalar, $\alpha(x, t)$, and vectorial, $\boldsymbol{\alpha}(x, t)$, fields within an Eulerian frame of reference. At the basis of the equations introduced, in detail of (4.11) and (4.12), later it will be possible to derive easily the other ones suited for the fluid domain. By indicating with $V \subset \mathbb{R}^3$ an arbitrary control volume in the current configuration, and being ∂V its surface boundary, the momentum conservation, in the Eulerian form, for scalar and vectorial fields read:

$$\frac{D}{Dt}\int_V \alpha(x, t) \, d\Omega_t = \int_{\partial V} \boldsymbol{\phi}(x, t) \cdot \mathbf{n} \, d\gamma + \int_V b(x, t) \, d\Omega_t, \tag{4.9}$$

$$\frac{D}{Dt}\int_V \boldsymbol{\alpha}(x, t) \, d\Omega_t = \int_{\partial V} \boldsymbol{\Phi}(x, t) \cdot \mathbf{n} \, d\gamma + \int_V \mathbf{b}(x, t) \, d\Omega_t, \tag{4.10}$$

being $\int_{\partial V} \boldsymbol{\phi}(x, t) \cdot \mathbf{n}\, d\gamma$ the flux of α across ∂V and $\boldsymbol{\phi}$ the Eulerian vector determining the flux; b and $\mathbf{b}$ represent the source/sink terms for the scalar and vectorial case, respectively. At this point, using the Reynolds transport theorem, the conservation law, in the Eulerian form, for a generic scalar field α reads:

$$\int_V \frac{\partial \alpha}{\partial t}\, d\Omega_t = \int_{\partial V} (\boldsymbol{\phi} - \alpha \mathbf{u}) \cdot \mathbf{n}\, d\gamma + \int_V b\, d\Omega_t, \qquad (4.11)$$

while for a vectorial field $\boldsymbol{\alpha}$ we get:

$$\int_V \frac{\partial \boldsymbol{\alpha}}{\partial t}\, d\Omega_t = \int_{\partial V} (\boldsymbol{\Phi} - \boldsymbol{\alpha} \otimes \mathbf{u}) \cdot \mathbf{n}\, d\gamma + \int_V \mathbf{b}\, d\Omega_t. \qquad (4.12)$$

In this way it is possible to further proceed writing the mass and momentum conservation laws for the fluid domain with respect to the Eulerian frame of reference. Concerning the mass conservation, with respect to (4.11), by taking the density of the continuum medium as the scalar field α and by setting to zero both the flux and the source/sink terms, we obtain:

$$\int_V \frac{\partial \rho_F}{\partial t}\, d\Omega_t = \int_{\partial V} -\rho_F \mathbf{u} \cdot \mathbf{n}_F\, d\gamma, \qquad (4.13)$$

Thanks to the divergence theorem the above equation can be rewritten as it follows:

$$\int_V \frac{\partial \rho_F}{\partial t}\, d\Omega_t = \int_V -\rho_F \nabla \cdot \mathbf{u}\, d\Omega_t. \qquad (4.14)$$

At this point, on the basis of the hypothesis of incompressibility of the fluid and employing the localization argument, we get:

$$\nabla \cdot \mathbf{u} = 0 \ \text{ in } \ \Omega_t, \qquad (4.15)$$

which represents the mass conservation for an incompressible fluid in the Eulerian frame of reference. In order to write the momentum conservation law, with respect to (4.11), we consider $\boldsymbol{\alpha}$ to be equal to the vectorial field $\rho_F \mathbf{u}$, Φ represents the flux vector $\sigma_F \cdot \mathbf{n}_F$, while the source/sink term $\mathbf{b}$ indicates the momentum generated by the volume forces $\mathbf{f}_F$ (per unit mass) acting on the fluid. On the basis of this setting we can write the momentum conservation law as it follows:

$$\int_V \frac{\partial \rho_F \mathbf{u}}{\partial t}\, d\Omega_t = \int_{\partial V} (\sigma_f - \rho_F \mathbf{u} \otimes \mathbf{u}) \cdot \mathbf{n}_F\, d\gamma + \int_V \rho_F \mathbf{f}_F\, d\Omega_t. \qquad (4.16)$$

Since the fluid model that we considered is the Newtonian one, through the constitutive equation it is possible to relate the fluid stress tensor $\boldsymbol{\sigma}_F$ to the fluid velocity $\mathbf{u}$. Within such a model the stress tensor depends linearly on the symmetric part of the velocity gradient:

$$\boldsymbol{\sigma}_F = \mu_F(\nabla\mathbf{u} + (\nabla\mathbf{u})^T) - p\mathbf{I}, \qquad (4.17)$$

where p denotes the fluid pressure; the above relation is generally adopted in fluid dynamics to model Newtonian viscous fluids. By substituting the constitutive relation in (4.16), and employing once again the divergence theorem we get:

$$\int_V \frac{\partial \rho_F \mathbf{u}}{\partial t} \, d\Omega_t = \int_V \mu_F \Delta\mathbf{u} - \nabla p - \rho_F(\mathbf{u}\cdot\nabla)\mathbf{u} + \rho_F\mathbf{f}_F \, d\Omega_t. \qquad (4.18)$$

By employing the mass conservation law to simplify the left-hand-side of (4.18), as well as the localization argument for the whole equation, the momentum conservation law can be written pointwise as it follows:

$$\rho_F\frac{\partial\mathbf{u}}{\partial t} + \rho_F(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \mu_F\Delta\mathbf{u} = \rho_F\mathbf{f}_F \ \text{ in } \ \Omega_t. \qquad (4.19)$$

The governing equations, (4.15) and (4.19), gained for the fluid domain are the Navier-Stokes equations, which in the Eulerian frame are the following:

$$\begin{cases} \rho_F\dfrac{\partial\mathbf{u}}{\partial t} + \rho_F(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \mu_F\Delta\mathbf{u} = \rho_F\mathbf{f}_F \\ \nabla\cdot\mathbf{u} = 0 \end{cases} \qquad (4.20)$$

In order to write the above system of equations with respect to the ALE frame of reference, due to motion of the fluid domain Ω_t, we apply the chain rule to the velocity time-derivative:

$$\left.\frac{\partial\mathbf{u}}{\partial t}\right|_{\mathcal{A}} = \frac{\partial\mathbf{u}}{\partial t} + \mathbf{w}\cdot\nabla\mathbf{u}. \qquad (4.21)$$

Thanks to the above relation we obtain the Navier-Stokes equations within the ALE formulation:

$$\begin{cases} \rho_F\left.\dfrac{\partial\mathbf{u}}{\partial t}\right|_{\mathcal{A}} + \rho_F[(\mathbf{u} - \mathbf{w})\cdot\nabla]\mathbf{u} + \nabla p - \mu_F\Delta\mathbf{u} = \rho_F\mathbf{f}_F \\ \nabla\cdot\mathbf{u} = 0 \end{cases} \qquad (4.22)$$

4.3   Derivation of the structure equations

In this Section we will derive, thanks to the momentum conservation law, the equation of motion for the structure with respect to a Lagrangian frame of reference. By indicating with V an arbitrary control volume in the current configuration, and being ∂V its surface boundary, the momentum conservation reads:

$$\frac{D}{Dt} \int_V \rho_S \dot{\mathbf{d}}_S \, dV = \int_V \rho_S \mathbf{f_s} \, dV + \int_{\partial V} \mathbf{p} \, dS. \tag{4.23}$$

where we indicated with $\mathbf{f_s}$ and $\mathbf{p}$ the force terms per unit mass and per unit surface, respectively. Thanks to the Cauchy relation, $p = \boldsymbol{\sigma}\mathbf{n}$, together with the Nanson's formula, which relates the element area between the current and the reference configurations, namely $\mathbf{n}dS = J\mathbf{F}^{-T}\hat{\mathbf{n}}d\hat{S}$, it is possible to write:

$$\frac{D}{Dt} \int_{\hat{V}} J\rho_S \dot{\mathbf{d}}_S \, d\hat{V} = \int_{\hat{V}} J\rho_S \mathbf{f_s} \, d\hat{V} + \int_{\partial \hat{V}} J\boldsymbol{\sigma}\mathbf{F}^{-T}\hat{\mathbf{n}}d\hat{S}, \tag{4.24}$$

where we adopted, also, the relation $dV = Jd\hat{V}$. At this point, in order to achieve the final expression of the momentum conservation law, we introduce the first Piola-Kirchhoff stress tensor:

$$\boldsymbol{\Sigma}_I = J\boldsymbol{\sigma}\mathbf{F}^{-T}. \tag{4.25}$$

At the basis of this definition, and thanks to the divergence theorem as well, it is possible to rewrite (4.24) as it follows:

$$\frac{D}{Dt} \int_{\hat{V}} \hat{\rho}_S \dot{\mathbf{d}}_S \, d\hat{V} = \int_{\hat{V}} \hat{\rho}_S \mathbf{f_s} \, d\hat{V} + \int_{\hat{V}} \hat{\nabla} \cdot \boldsymbol{\Sigma}_I \, d\hat{V}. \tag{4.26}$$

Finally, adopting both the localization argument and the fact that $\hat{V}$ is fixed, we obtain:

$$\hat{\rho}_s \frac{\partial^2 \hat{\mathbf{d}}_S}{\partial t^2} - \hat{\nabla} \cdot \boldsymbol{\Sigma}_I = \hat{\rho}_S \mathbf{f_s}, \tag{4.27}$$

which represents the equation of motion for the structural sub-system. It is possible to proceed by detailing the constitutive law adopted in order to model the mechanical behavior that describes the structure material. In this way, we introduce firstly the second Piola-Kirchhoff stress tensor:

$$\boldsymbol{\Sigma}_{II} = \mathbf{F}^{-1}\boldsymbol{\Sigma}_I, \tag{4.28}$$

this choice is due to the fact that $\boldsymbol{\Sigma}_{II}$ is symmetric, it respects the axiom of frame

indifference as well, and it would be adopted to write the constitutive law, instead of $\boldsymbol{\Sigma}_I$. In particular $\boldsymbol{\Sigma}_{II}$ is independent from rigid motions (frame indifference): we can explain this concept through the introduction of a rotation tensor $\mathbf{R}$, that is orthogonal ($\mathbf{R}\mathbf{R}^T = \mathbf{I}$); by indicating with $\bar{(.)}$ the rotated entities, we can write:

$$\bar{\boldsymbol{\Sigma}}_{II} = J\bar{\mathbf{F}}^{-1}\bar{\boldsymbol{\sigma}}\bar{\mathbf{F}}^{-1} = J(\mathbf{F}^{-1}\mathbf{R}^T)(\mathbf{R}\boldsymbol{\sigma}\mathbf{R}^T)(\mathbf{R}\mathbf{F}^{-T}) = \boldsymbol{\Sigma}_{II}, \qquad (4.29)$$

such that $\bar{\boldsymbol{\Sigma}}_{II} = \boldsymbol{\Sigma}_{II}$. As strain tensor to be adopted in the constitutive law we refer to the Green-Lagrange one:

$$\in = \frac{1}{2}[(\nabla\hat{\mathbf{d}}_s)^T + \nabla\hat{\mathbf{d}}_s + ((\nabla\hat{\mathbf{d}}_s)^T \cdot \nabla\hat{\mathbf{d}}_s)]. \qquad (4.30)$$

If there exists a scalar valued strain energy function W depending on $\in$, such that:

$$\frac{\partial W(\in)}{\partial \in} = \boldsymbol{\Sigma}_{II}(\in), \qquad (4.31)$$

the material is called hyperelastic. As strain energy function we adopted the St. Venant-Kirchhoff one, that is defined as it follows:

$$W(\in) = \frac{L_1}{2}(\mathrm{tr}(\in)^2 + L_2\mathrm{tr}(\in^2)), \qquad (4.32)$$

where L_1 and L_2 represent the Lamé coefficients defining the mechanical characteristics of the material. In this way, taking the derivative of (4.31) we gain:

$$\boldsymbol{\Sigma}_{II} = L_1(\mathrm{tr}(\in))\mathbf{I} + 2L_2\in. \qquad (4.33)$$

Since expression (4.33) is linear, it is possible to rewrite it into a more compact way:

$$\boldsymbol{\Sigma}_{II} = \mathbf{C} : \in \quad (\text{component-wise} \;\; \Sigma_{II_{ik}} = C_{ikrs}\in_{rs}), \qquad (4.34)$$

being $\mathbf{C}$ a fourth-order tensor. Instead of using, as shown in (4.32), the Lamé constants to characterize the St. Venant-Kirchhoff materials, we will refer hereafter to the Young modulus E and the Poisson ratio ν; equation (4.35) expresses the relation held between the two sets of coefficients:

$$E = L_2\frac{3L_1 + 2L_2}{L_1 + L_2}, \quad L_1 = \frac{E\nu}{(1 - 2\nu)(1 + \nu)},$$
$$\nu = \frac{L_1}{2(L_1 + L_2)}, \quad L_2 = \frac{E}{2(1 + \nu)}. \qquad (4.35)$$

We remark that the constitutive law introduced is nonlinear in the displacement $\hat{\mathbf{d}}_S$, since both $\boldsymbol{\Sigma}_{II}$ and $\in$ are nonlinear in $\mathbf{F}$. In order to get the expression of linear elasticity adopted in our formulation, we further proceed by neglecting the terms which are of order higher than one in (4.30), leading to:

$$\in = \frac{1}{2}[(\hat{\nabla}\hat{\mathbf{d}}_s)^T + \hat{\nabla}\hat{\mathbf{d}}_s + ((\hat{\nabla}\hat{\mathbf{d}}_s)^T \cdot \hat{\nabla}\hat{\mathbf{d}}_s)] \approx \frac{1}{2}[(\hat{\nabla}\hat{\mathbf{d}}_s)^T + \hat{\nabla}\hat{\mathbf{d}}_s] = \tilde{\in}, \qquad (4.36)$$

where $\tilde{\in}$ indicates the symmetric part of the displacement gradient and, thus, it best suites those cases involving small deformations. At this point, we further simplify equation (4.33):

$$L_1(\mathrm{tr}(\in))\mathbf{I} + 2L_2\in = \boldsymbol{\Sigma}_{II} = \mathbf{F}^{-1}\boldsymbol{\Sigma}_I \approx \boldsymbol{\Sigma}_I. \qquad (4.37)$$

Thanks to the simplifications introduced, finally we get the equation of the linear elasticity:

$$\frac{\partial^2 \hat{\mathbf{d}}_S}{\partial t^2} - \hat{\nabla} \cdot (L_1(\mathrm{tr}(\in))\mathbf{I} + 2L_2\in) = \hat{\rho}_S\mathbf{f}_S. \qquad (4.38)$$

4.4   Coupling conditions and three fields formulation

Since we have derived, in the previous Sections, the equations that model both the fluid and the structure physics, here we will introduce the coupling conditions adopted to get the coupled FSI problem. With respect to Figure 5.14, we illustrate the notation that will be followed hereafter to describe the boundaries adopted to model the coupled FSI problem. With $\hat{\Gamma}$ we represent the fluid-structure interface in the reference configuration, on which the coupling conditions are expressed as it follows:

- continuity of the velocity;

- continuity of the stress;

- continuity of the domain displacement.

Reference configuration   Current configuration
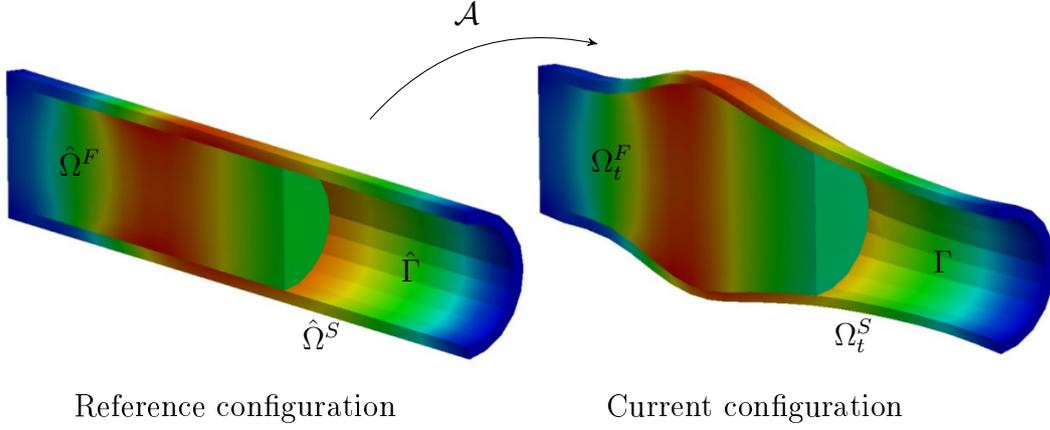
Figure 4.2: Notation adopted for both the domains and the Fluid-Structure interface.

The last coupling condition is due to the adoption of the ALE frame to model the fluid flow: it indicates the geometric adherence between the fluid domain and the material particles of the structure across $\hat{\Gamma}$. On the basis of the previously introduced background, the whole system of equations that holds the FSI problem reads as:

$$\nabla \cdot \mathbf{u} = 0 \qquad \text{in } \Omega_t^F, \quad (4.39)$$

$$\rho_F \frac{\partial \mathbf{u}}{\partial t}\bigg|_{\mathcal{A}} + \rho_F[(\mathbf{u} - \mathbf{w}) \cdot \nabla]\mathbf{u} + \nabla p - \mu_F \Delta \mathbf{u} = \rho_F \mathbf{f}_F \quad \text{in } \Omega_t^F, \quad (4.40)$$

$$\hat{\rho}_s \frac{\partial^2 \hat{\mathbf{d}}_S}{\partial t^2} - \hat{\nabla} \cdot \mathbf{\Sigma}_I = \hat{\rho}_S \mathbf{f_s} \quad \text{in } \Omega_t^S, \quad (4.41)$$

$$\frac{\partial \hat{\mathbf{d}}_S}{\partial t} = \mathbf{u} \circ \mathcal{A}_t \quad \text{on } \hat{\Gamma}, \quad (4.42)$$

$$\hat{\boldsymbol{\sigma}}_F \hat{\mathbf{n}}_F + \mathbf{\Sigma}_I \hat{\mathbf{n}}_S = 0 \qquad \text{on } \hat{\Gamma}, \quad (4.43)$$

$$\frac{\partial \hat{\mathbf{d}}_F}{\partial t} = \mathbf{w} \circ \mathcal{A}_t \quad \text{on } \hat{\Gamma}, \quad (4.44)$$

where equations (4.42), (4.43) and (4.44) indicate the coupling conditions. Moreover, in this work, we decided to set to zero both the terms $\mathbf{f}_F$ and $\mathbf{f}_S$, since we do not account for the gravity forces. Once we have outlined the system of equation governing the fluid-structure interaction problem, we can introduce a more compact notation, that we will adopt in the following Sections, to describe it as being made of three coupled sub-problems. In this way we refer to the fluid problem as:

$$\mathcal{F}(\mathbf{u}_F, \hat{\mathbf{d}}_F, \hat{\mathbf{d}}_S) = 0, \quad (4.45)$$

where $\mathbf{u}_F$ represents the unknown fluid variables $\mathbf{u}_F = (\mathbf{u}, p)$. Such a problem, at every $t > 0$, relies on finding the unknown fluid pair $\mathbf{u}_F$ given the structure displacement $\hat{\mathbf{d}}_S$, or the traction vector $\boldsymbol{\Sigma}_I \hat{\mathbf{n}}_S$ (or a combination of both as well), and the fluid domain displacement $\hat{\mathbf{d}}_F$. We define the structure problem as:

$$\mathcal{S}(\mathbf{u}_F, \hat{\mathbf{d}}_S) = 0, \tag{4.46}$$

representing the equation of motion (4.27) of the structural system. In this case the unknown is the structural displacement $\hat{\mathbf{d}}_S$, to be found for a given fluid velocity $\mathbf{u}$, or traction vector $\boldsymbol{\sigma}_F \hat{\mathbf{n}}_F$ (or a combination of both as well). We introduce the geometric problem, that is defined as it follows:

$$\mathcal{M}(\hat{\mathbf{d}}_F, \hat{\mathbf{d}}_S) = 0, \tag{4.47}$$

where the operator $\mathcal{M}$ represents the mesh motion solver, thus defining the ALE map $\mathcal{A}$. Such a problem is coupled to the geometric adherence condition $\hat{\mathbf{d}}_F = \hat{\mathbf{d}}_S$ on $\hat{\Gamma}$. This work focuses on the comparison of shape parametrization strategies (i.e. RBF and FFD), as well the IDW technique, involved in solving such problem. Once we compute, for a given structural displacement, the new fluid mesh configuration, the ALE mapping can be defined as:

$$
\begin{aligned}
\mathcal{A} : \hat{\Omega}^F &\rightarrow \Omega_t^F \\
\hat{x} &\mapsto \mathcal{A}(\hat{x}) = \hat{x} + \hat{\mathbf{d}}_F(\hat{x}).
\end{aligned}
\tag{4.48}
$$

Before performing the time discretization of the system of equations that holds the FSI problem, it is worthwhile to survey the possible numerical strategies to be adopted in order to solve it. The solution of the FSI problem can be obtained by using either a modular (segregated) or non-modular (monolithic) strategy. The latter approach is followed when both the fluid and the structural problems are solved simultaneously within a single solver, while the first involves the adoptions of pre-existing fluid and structural solvers which are coupled through (4.42), (4.43) and (4.44). In the framework of partitioned strategies we can identify explicit or implicit schemes, the latter requiring sub-iterations among the fluid and structural solvers. Within explicit partitioned procedures the fluid and structure sub-problems are solved once (or just few times) per time step: such an approach is typical in aeroelasticity and rather inconvenient in haemodynamics [61]. In fact, it can be shown that an explicit approach to FSI problems with significant added-mass effect (like haemodynamics where the structure and fluid densities are similar) is instable [24]. Therefore, to avoid fluid-solid sub-iterations, we use

a non-modular or monolithic strategy, where the fluid and the solid are treated as a single problem. In detail, on the basis of the state of the art solvers implemented within the finite element library LifeV, we decided to adopt a monolithic approach to solve the FSI problem using, as time discretization, a Geometric-Convective Explicit scheme (GCE), whose hypothesis and characteristics will be fully detailed in the next Section.

4.5 Geometry-Convective Explicit time discretization

In this Section we focus our attention on the time discretization of the FSI problem. In detail, the numerical scheme that we adopted is the Geometry-Convective Explicit [12] one: such a strategy, as will be clarified in this Section, would give us the possibility of studying, within a monolithic approach, the behavior of different mesh motion strategies like RBF, FFD and IDW, since the geometry problem (4.47) is solved separately once per time step, apart from the coupled system of equations that govern both the fluid and the structure physics. In order to assess the hypothesis of such an approach we need, firstly, to pay attention on the time discretization of the fluid problem. A large variety of time discretization schemes of the incompressible Navier-Stokes (NS) equations on moving domain is present in literature: nevertheless, the adoption of the ALE formulation imposes some constraints. In fact, the way we discretize in time the convective term in (4.19), namely $([(\mathbf{u} - \mathbf{w}) \cdot \nabla]\mathbf{u})$, plays a fundamental role in determining the order of the time advancing scheme, as well as its stability properties. In this way, it is possible to highlight within the coupled FSI problem several levels of nonlinearities, with some of them related to the choice of the ALE frame:

- $i)$ $(\mathbf{u} \cdot \nabla)\mathbf{u}$, which is the convective term of the NS equations in fixed domains;

- $ii)$ $(\mathbf{w} \cdot \nabla)\mathbf{u}$, that represents the advective term due to the ALE formulation;

- $iii)$ $\hat{\mathbf{d}}_F = \hat{\mathbf{d}}_F(\hat{\mathbf{d}}_S)$, the dependence of the fluid domain displacement on the structural one.

At each time-step all these nonlinearities can be solved by means of a Newton, or Fixed Point algorithm [94]. Within this choice relies the Fully Implicit (FI) time discretization which leads to an accurate and robust time discretization, although it represents an expensive approach. By modifying the way the nonlinear terms are discretized in time, we can obtain different approaches. By indicating with the superscript $n + 1$ the current time-step, the nonlinearity due to $(\mathbf{u} \cdot \nabla)\mathbf{u}$ can be discretized in time, given an extrapolation $\mathbf{u}^*$ of the fluid velocity from the

previous time-step, as it follows:

$$(\mathbf{u}^* \cdot \nabla)\mathbf{u}^{n+1}. \tag{4.49}$$

Thanks to this choice the convective term $(\mathbf{u} \cdot \nabla)\mathbf{u}$ becomes linear. A good candidate to approximate $\mathbf{u}^*$ corresponds to $\mathbf{u}^n$, such that $\mathbf{u}^* = \mathbf{u}^n$. Such an approximation is suitable when the characteristic Reynolds number of the fluid flow is not very high (condition fulfilled when dealing with fluid flows in laminar regimes). Moreover this approximation for the convective term is used, e.g. in [36, 47]. In order to deal with the second nonlinearity, namely the advective term $(\mathbf{w} \cdot \nabla)\mathbf{u}$, it can be discretized in time as it follows:

$$(\mathbf{w}^* \cdot \nabla)\mathbf{u}^{n+1}, \tag{4.50}$$

being $\mathbf{w}^*$ an explicit extrapolation of the fluid domain velocity. The way we approximate this advective term, together with (4.49) leads to a linearization of the convective term appearing in (4.19). Concerning the third nonlinearity, due to the moving fluid domain, is it possible to handle it by considering the fluid domain at the previous time-step n, as a suitable approximation of the current one, on the basis of the hypothesis of small displacements. Its explicit treatment, together with (4.49) and (4.50), leads to the *Geometric-Convective Explicit* scheme [29, 12], in which the fluid problem is *linear*. By following such an approach, the coupled time-discrete system of equations written in a compact way, at the time-step t^{n+1}, reads:

$$
\begin{aligned}
\mathcal{F}^n(\mathbf{u}_F^{n+1}, \hat{\mathbf{d}}_F^{n+1}, \hat{\mathbf{d}}_S^{n+1}) &= 0, \\
\mathcal{S}^n(\mathbf{u}_F^{n+1}, \hat{\mathbf{d}}_S^{n+1}) &= 0, \\
\mathcal{M}^n(\hat{\mathbf{d}}_F^{n+1}, \hat{\mathbf{d}}_S^n) &= 0.
\end{aligned}
$$

In this way it is possible to write the time discretized equations, in strong form (with the fluid momentum equation written in non-conservative form), that define the GCE scheme:

$$
\begin{aligned}
\rho_F \delta_t \mathbf{u}^{n+1} + \rho_F((\mathbf{u}^n - \mathbf{w}^{n+1}) \cdot \nabla)\mathbf{u}^{n+1} - \nabla \cdot \boldsymbol{\sigma}_F^{n+1} &= \rho_F \mathbf{f}_F^{n+1} & \text{in } \Omega_{t_{n+1}}^F, \\
\nabla \cdot \mathbf{u}^{n+1} &= 0 & \text{in } \Omega_{t_{n+1}}^F, \\
\hat{\rho}_S \delta_{tt} \hat{\mathbf{d}}_S^{n+1} - \hat{\nabla} \cdot \boldsymbol{\Sigma}_I^{n+1} &= \rho_S \mathbf{f}_S^{n+1} & \text{in } \hat{\Omega}^S, \\
\delta_t \hat{\mathbf{d}}_S^{n+1} - \mathbf{u}^{n+1} \circ \mathcal{A}_{t_{n+1}} &= 0 & \text{on } \hat{\Gamma}, \\
\boldsymbol{\Sigma}_I^{n+1} \hat{\mathbf{n}}_S + \hat{\boldsymbol{\sigma}}_F^{n+1} \cdot \hat{\mathbf{n}}_F &= 0 & \text{on } \hat{\Gamma},
\end{aligned}
$$

where with δ_t we denoted the first order discrete time derivative for the fluid problem, and with δ_{tt} the second order one adopted within the structure equation. As previously introduced, the geometry problem

$$\mathcal{M}^n(\hat{\mathbf{d}}_F^{n+1}, \hat{\mathbf{d}}_S^n) = 0, \tag{4.51}$$

is solved separately once per time step, whose solution defines both the ALE map $\mathcal{A}_{t_{n+1}}$, and the new fluid domain configuration $\Omega_{t_{n+1}}^F = \mathcal{A}_{t_{n+1}}(\hat{\Omega}^F)$. Concerning the fluid domain velocity $\mathbf{w}^{n+1}$, it is computed as it follows:

$$\mathbf{w}^{n+1} = \delta_t \hat{\mathbf{d}}_F^{n+1} \circ \mathcal{A}_{t_{n+1}}^{-1}. \tag{4.52}$$

By adopting this approach we gain also the advantage of dealing with a linear FSI problem, since the structural model considered relies on linear elasticity as well. As already mentioned, we remark that the choice related to the GCE approach is due to the flexibility given by such formulation: in fact, it isolates the geometry (or mesh motion) problem allowing the possibility to use, and compare, all the strategies that have been introduced in Chapters 2 and 3.

4.6 Weak formulation of the FSI problem

In this Section we will firstly derive the weak formulation of the fluid-structure interaction problem [36, 86] and, then, we will report, within the GCE time discretization scheme, the linear system that holds the coupled fluid-structure problem [29]. To gain the weak form of the FSI system it is previously necessary to define the following functional spaces:

$$\begin{cases} U_t^F & := \{\mathbf{v} : \Omega_t^F \to \mathbb{R}^d, \, \mathbf{v} = \hat{\mathbf{v}} \circ \mathcal{A}_t^{-1} | \, \hat{\mathbf{v}} \in H^1(\hat{\Omega}^F)^d\}, \\ U_t^S & := \{\mathbf{v} : \hat{\Omega}^S \to \mathbb{R}^d | \, \mathbf{v} \in H^1(\hat{\Omega}^S)^d\}, \\ U^{\hat{\Gamma}} & := H^{1/2}(\hat{\Gamma})^d, \\ Q_t & := \{q : \Omega_t^F \to \mathbb{R}, \, q = \hat{q} \circ \mathcal{A}_t^{-1} | \, \hat{q} \in L^2(\hat{\Omega}^F)\}, \\ V_t^F & := \{\mathbf{v} : \Omega_t^F \to \mathbb{R}^d, \, \mathbf{v} = \hat{\mathbf{v}} \circ \mathcal{A}_t^{-1} | \, \hat{\mathbf{v}} \in H^1(\hat{\Omega}^F)^d, \hat{\mathbf{v}}|_{\Gamma_D^F} = 0\}, \\ V_t^S & := \{\mathbf{v} : \hat{\Omega}^S \to \mathbb{R}^d, \, \mathbf{v} \in H^1(\hat{\Omega}^S)^d, \, \mathbf{v}|_{\Gamma_D^S} = 0\}, \end{cases} \tag{4.53}$$

where with L^p and H^q we indicated the Banach and the Sobolev spaces, respectively [94]. Moreover, d represents the space dimensions (2 for the bi-dimensional case, 3 for the three-dimensional one), while the subscripts Γ_D^S and Γ_D^F indicate the Dirichlet boundaries (without the interface Γ) of $\hat{\Omega}^S$ and Ω_t^F. Before deriving the weak formulation of the coupled fluid-structure problem, we prefer to better

asses the way through the coupling conditions are enforced: concerning the velocity continuity, it is imposed in its strong way, while the continuity of stresses is imposed weakly, by equating the variational residual of the two momentum equations (of both the fluid and the structure) restricted to the interface. To clarify the latter coupling condition we can write, firstly, the variational formulation of the fluid momentum equation for an arbitrary $\mathbf{v}_F \in V_t^F$:

$$
\int_{\Omega_t^F} \left(\rho_F \frac{\partial \mathbf{u}}{\partial t} \bigg|_{\mathcal{A}} \cdot \mathbf{v}_F + \rho_F [(\mathbf{u} - \mathbf{w}) \cdot \nabla] \mathbf{u} \cdot \mathbf{v}_F + \boldsymbol{\sigma}_F : \nabla \mathbf{v}_F - \rho_F \mathbf{f}_F \mathbf{v}_F \right) d\Omega_t^F +
$$

$$
- \int_{\Gamma_N^F} \mathbf{g}_F^N \cdot \mathrm{tr}_\Gamma(\mathbf{v}_F) d\gamma = \int_\Gamma \boldsymbol{\sigma}_F \mathbf{n}_F \cdot \mathrm{tr}_\Gamma(\mathbf{v}_F) d\gamma. \quad (4.54)
$$

By operating in the same way for the structure problem, choosing as test function $\mathbf{v}_S \in V_t^S$:

$$
\int_{\hat{\Omega}^S} \left(\hat{\rho}_s \frac{\partial^2 \mathbf{d}_S}{\partial t^2} \cdot \mathbf{v}_s - (\hat{\nabla} \cdot \boldsymbol{\Sigma}_I) \cdot \mathbf{v}_s - \hat{\rho}_S \mathbf{f}_\mathbf{s} \cdot \mathbf{v}_s \right) d\hat{\Omega}^S - \int_{\Gamma_N^S} \mathbf{g}_S^N \cdot \mathrm{tr}_{\hat{\Gamma}}(\mathbf{v}_S) d\gamma =
$$

$$
= \int_{\hat{\Gamma}} \boldsymbol{\Sigma}_I \hat{\mathbf{n}}_S \cdot \mathrm{tr}_{\hat{\Gamma}}(\mathbf{v}_S) d\hat{\gamma}. \quad (4.55)
$$

With Γ_N^S and Γ_N^F we indicated the Neumann boundaries (without the interface Γ) of $\hat{\Omega}^S$ and Ω_t^F, while $\mathbf{g}_S^N \in (H^{1/2}(\Gamma_N^S))^d$ and $\mathbf{g}_F^N \in (H^{1/2}(\Gamma_N^F))^d$ represent the given boundary data. Once we express the stress tensors in their reference configuration, the stresses continuity across $\hat{\Gamma}$ in its weak form reads:

$$
\int_{\hat{\Gamma}} \boldsymbol{\Sigma}_I \hat{\mathbf{n}}_S \cdot v_{\hat{\Gamma}} \, d\hat{\gamma} + \int_{\hat{\Gamma}} \hat{\boldsymbol{\sigma}}_F \hat{\mathbf{n}}_F \cdot v_{\hat{\Gamma}} \, d\hat{\gamma} = 0 \qquad \forall v_{\hat{\Gamma}} \in U^{\hat{\Gamma}}. \quad (4.56)
$$

By introducing two linear continuous lift operators, namely $\mathcal{I}_F : U^{\hat{\Gamma}} \to V_t^F$ and $\mathcal{I}_S : U^{\hat{\Gamma}} \to V_t^S$, and by considering equations (4.54) and (4.55), it is possible to write:

$$
\int_{\hat{\Omega}^S} \left(\hat{\rho}_s \frac{\partial^2 \mathbf{d}_S}{\partial t^2} \cdot \mathcal{I}_S(v_{\hat{\Gamma}}) \right) d\hat{\Omega} + \int_{\hat{\Omega}^S} (\boldsymbol{\Sigma}_I : \hat{\nabla}\mathcal{I}_S(v_{\hat{\Gamma}}) - \hat{\rho}_S \cdot \mathcal{I}_S(v_{\hat{\Gamma}})) d\hat{\Omega} - \int_{\Gamma_N^S} \mathbf{g}_S^N \cdot \mathcal{I}_S(v_{\hat{\Gamma}}) d\gamma +
$$

$$
+ \int_{\hat{\Omega}^S} \left(\rho_F \frac{\partial \mathbf{u} \circ \mathcal{A}_t^{-1}}{\partial t} \bigg|_{\mathcal{A}} \cdot \mathcal{I}_F(v_{\hat{\Gamma}}) + \rho_F [(\mathbf{u} \circ \mathcal{A}_t^{-1} - \mathbf{w} \circ \mathcal{A}_t^{-1}) \cdot \hat{\nabla}] \mathbf{u} \circ \mathcal{A}_t^{-1} \cdot \mathcal{I}_F(v_{\hat{\Gamma}}) + \right.
$$

$$
\left. + \hat{\boldsymbol{\sigma}}_F : \hat{\nabla} \cdot \mathcal{I}_F(v_{\hat{\Gamma}}) - \hat{\rho}_F \hat{\mathbf{f}}_F \cdot \mathcal{I}_F(v_{\hat{\Gamma}}) \right) d\hat{\Omega} - \int_{\Gamma_N^S} \mathbf{g}_F^N \cdot \mathcal{I}_F(v_{\hat{\Gamma}}) \, d\gamma = 0 \qquad \forall v_{\hat{\Gamma}} \in U^{\hat{\Gamma}}.
$$

$$
(4.57)
$$

Thus, the weak coupling condition on the stresses continuity can be achieved by choosing test functions which, restricted to the fluid-structure interface, are matching. To this end, we can further introduce the functional space

$$S_t := \{(\mathbf{v}_F, \mathbf{v}_S) \in V_t^F \times V_t^S | \mathrm{tr}_{\hat{\Gamma}}(\mathbf{v}_F \circ \mathcal{A}_t^{-1}) = \mathrm{tr}_{\hat{\Gamma}}(\mathbf{v}_S)\}$$

that guarantees automatically, for all the $(\mathbf{v}_F, \mathbf{v}_S) \in S_t$, the stress continuity across the interface. It is possible now, once we clarified the way the coupling conditions on the velocity (in strong form) and the stresses (weakly) are enforced, to introduce the global weak formulation of the FSI problem following [86, 114]. The weak formulation of equations (4.22) and (4.27) reads:

Weak-formulation 3.1 (FSI problem): *for each $t \in (0, T)$, find $\mathbf{u} \in U_t^F$, $p \in Q_t$ and $\hat{\mathbf{d}}_S \in U_t^S$, such that $\mathbf{u} = \mathbf{g}_F^D$ on Γ_D^F and $\hat{\mathbf{d}}_S = \mathbf{g}_S^D$ on Γ_D^S, satisfying:*

$$\int_{\Omega_t^F} \left(\rho_F \frac{\partial \mathbf{u}}{\partial t}\bigg|_{\mathcal{A}} \cdot \mathbf{v}_F + \rho_F[(\mathbf{u} - \mathbf{w}) \cdot \nabla]\mathbf{u} \cdot \mathbf{v}_F + \boldsymbol{\sigma}_F : \nabla \mathbf{v}_F \right) d\Omega_t^F +$$

$$\int_{\hat{\Omega}^S} \left(\hat{\rho}_s \frac{\partial^2 \mathbf{d}_S}{\partial t^2} \cdot \mathbf{v}_s + \boldsymbol{\Sigma}_I : \nabla \mathbf{v}_s \right) d\hat{\Omega}^S = \int_{\Omega_t^F} \rho_F \mathbf{f}_F \cdot \mathbf{v}_F \, d\Omega_t^F +$$

$$+ \int_{\hat{\Omega}^S} \hat{\rho}_S \mathbf{f_s} \cdot \mathbf{v}_s \, d\hat{\Omega}^S - \int_{\Gamma_N^F} \mathbf{g}_F^N \cdot \mathbf{v}_F \, d\gamma - \int_{\Gamma_N^S} \mathbf{g}_S^N \cdot \mathbf{v}_S \, d\hat{\gamma}$$

$$\forall (\mathbf{v}_F, \mathbf{v}_S) \in S_t, \qquad (4.58)$$

$$\int_{\Omega_t^F} \mathbf{u} \cdot \nabla q \, d\Omega_t^F = 0 \quad \forall q \in Q_t. \qquad (4.59)$$

We remark that, assuming matching grids and equal interpolation spaces for the fluid velocity and structure displacement, the Dirichlet boundary conditions (continuity of velocities) are enforced strongly, by discretizing directly

$$\frac{\partial \hat{\mathbf{d}}_S}{\partial t} = \mathbf{u} \circ \mathcal{A}_t \quad \text{on} \quad \hat{\Gamma}, \qquad (4.60)$$

while the Neumann boundary conditions (on stresses) are imposed weakly, given the boundary data $\mathbf{g}_S^N \in (H^{1/2}(\Gamma_N^S))^d$ and $\mathbf{g}_F^N \in (H^{1/2}(\Gamma_N^F))^d$. In order to obtain the Finite Element (FE) discretization of the FSI problem, we consider its variational formulation and we approximate, by means of finite dimensional spaces, the functional spaces introduced through (4.53). In this way, with the label $_h$ we indicate the approximated spaces, spanned by a basis of shape functions defined in the reference domain, $\{\psi_i\}_{i=1}^N$, being $N = \dim(U_{t,h}^F)$. In this way, the fluid

velocity $\mathbf{u}$ is approximated on Ω_t^F as it follows:

$$\mathbf{u}(x,t) \approx \mathbf{u}_h(x,t) = \sum_{i=1}^{N} \mathbf{u}_i(t)\psi_i(\mathcal{A}_{h,t}^{-1}(x)). \qquad (4.61)$$

We aim, now at writing the fluid-structure system yielded by the linearized and fully discretized FSI problem. We start by introducing the unknowns for the fluid problem: $\mathbf{u}_F^{n+1}$ and $\mathbf{u}_{F_\Gamma}^{n+1}$. They represent the arrays of nodal values for the velocity and pressure unknowns of the inner and of the interface nodes, respectively, since we will distinguish, in the resulting algebraic system, between the degrees of freedom lying on the FS interface, and those who are internal to Ω_t^F and $\hat{\Omega}^S$. The structural unknowns are $\hat{\mathbf{d}}_S^{n+1}$ and $\hat{\mathbf{d}}_{S_\Gamma}^{n+1}$, in accord with the same notation that we have previously introduced. Since the geometry (or mesh motion) problem is solved separately once per time-step due to the GCE time discretization, the fluid-structure coupled problem discretized in space leads to the following linear system:

$$\left[\begin{array}{cc|cc|c} \mathbf{F}_{FF} & \mathbf{F}_{F\Gamma} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{F}_{\Gamma F} & \mathbf{F}_{\Gamma\Gamma} & \mathbf{0} & \mathbf{0} & \mathbf{I} \\ \hline \mathbf{0} & \mathbf{0} & \mathbf{S}_{SS} & \mathbf{S}_{S\Gamma} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{S}_{\Gamma S} & \mathbf{S}_{\Gamma\Gamma} & -\mathbf{I} \\ \hline \mathbf{0} & \mathbf{I} & \mathbf{0} & -\mathbf{I}/\Delta t & \mathbf{0} \end{array} \right] \left\{ \begin{array}{c} \mathbf{u}_{F_F}^{n+1} \\ \mathbf{u}_{F_\Gamma}^{n+1} \\ \hline \hat{\mathbf{d}}_{S_S}^{n+1} \\ \hat{\mathbf{d}}_{S_\Gamma}^{n+1} \\ \hline \boldsymbol{\lambda}^{n+1} \end{array} \right\} = \left\{ \begin{array}{c} \mathbf{b}_{FF} \\ \mathbf{b}_{F\Gamma} \\ \hline \mathbf{b}_{SS} \\ \mathbf{b}_{S\Gamma} \\ \hline -\mathbf{I}/\Delta t \mathbf{d}_{S\Gamma}^n \end{array} \right\}, \qquad (4.62)$$

that within a more compact notation reads:

$$\mathbf{A}_{FSI}^{n+1}\mathbf{u}_{FSI}^{n+1} = \mathbf{b}_{FSI}^{n+1}. \qquad (4.63)$$

With respect to system (4.62), we remark the separation within the left hand side matrix $\mathbf{A}_{FSI}^{n+1}$ on two different levels: at the first, within the whole matrix structure, we identify three main blocks which represent the fluid, the structure and the interface sub-problems; furthermore, as previously introduced, within the fluid and structure parts, we separated those degrees of freedom lying on the FS interface, form the others, that are inner to Ω_t^F and $\hat{\Omega}^S$. In detail, the block $(1,1)$ represents the discretized fluid problem equations, the block $(2,2)$ indicates the discretized solid problem, while the off-diagonal terms enforce the coupling conditions between the two fields. Moreover, with respect to (4.62), the right-hand-sides are composed by the volume forces and the terms of the time discretization that depend on the previous time-step, while within the vector of the unknowns $\mathbf{u}_{FSI}^{n+1}$, the Lagrange multiplier $\boldsymbol{\lambda}$ has been introduced to enforce the coupling conditions across the fluid-structure interface. On the basis of the introduced background it is possible, finally, to summarize the main aspects that characterize the solution strategy for the FSI problem within a monolithic GCE approach:

- define an appropriate extrapolation of both the geometric and convective (4.49) terms. Concerning the first, on the hypothesis of small displacements, we can handle it by considering the fluid domain at the time-step t^n, as a suitable approximation of the current one. For the latter, we approximate $(\mathbf{u} \cdot \nabla)\mathbf{u}$ as $(\mathbf{u}^n \cdot \nabla)\mathbf{u}^{n+1}$. In this way we split the geometric part of the problem from the fluid-structure one;

- hence, at each time-step we solve separately the geometric problem. We compute, by means of FFD, IDW and RBF techniques, the fluid displacement $\mathbf{d}_F^{n+1}$ and we move the fluid mesh accordingly;

- by using the resulting fluid mesh velocity $\mathbf{w}_F^{n+1}$, we replace the convective term in the fluid momentum equation by $[(\mathbf{u}^n - \mathbf{w}_F^{n+1}) \cdot \nabla]\mathbf{u}^{n+1}$;

- finally, once we solved the geometry (mesh motion) problem, the unknowns vector is $\mathbf{u}_{FSI} = \{\mathbf{u}_F^T, \hat{\mathbf{d}}_S^T, \boldsymbol{\lambda}^T\}^T$; for a given $\mathbf{u}_{FSI}^n$, the coupled fluid-structure problem reads in finding $\mathbf{u}_{FSI}^{n+1}$ such that:

$$\mathbf{A}_{FSI}^{n+1}\mathbf{u}_{FSI}^{n+1} = \mathbf{b}_{FSI}^{n+1}.$$

4.7   Programming features of the FSI solver

Throughout this section we outline the solution strategy adopted within the FSI solver employed, namely LifeV. In detail we aim at giving an overview, by means of a block scheme, of the global pattern followed by the code in order to numerically solve the FSI problem within a Geometric-Convective Explicit scheme. To this end, as shown in Figures 4.3 and 4.4, we illustrate the whole pattern, as well as the classes involved within such a Finite Element library, that allow the user to solve the FSI problem. We refrain from detailing the implementative aspects of the numerical procedures, since we like better to describe the way through such a code solves the FSI coupled problem within a monolithic CGE scheme. Thus, to better assess the diagrams reported in Figures 4.3 and 4.4, we report below a brief description of the objects involved within the numerical solution of the FSI problem:

- `main`: it is the function used to call the methods of the classes involved in the solution of the FSI problem. Thus it manages the numerical solution of the FSI problems. Moreover, the `BoundaryConditions` and `ud_functions` files have to be set in order to properly handle the boundary conditions of the problem;

- `data`: it is the file that contains all the parameters to be properly set before running the FSI simulation. In detail, we need to specify those parameters that would control and menage the numerical solution of the problem. In

detail, there, we select the mesh motion strategy to be employed. By means of `GetPot` [1] the data chosen are stored within the `FSIData` class;

- `FSISolver`: it is a class that handles FSI iterations. This class contains four main methods, namely `setData`, `setup`, `initialize` and `iterate`. The first of them is used to load the data of the FSI problem, the `setup` method initializes the variables of the problem, while `initialize` sets the initial condition for the time scheme. The method `iterate` menages the temporal loop of the FSI simulation;

- `FSIOperator`: this class menages the definition of the solver for the FSI problem. Within such an class, the main methods are represented by: `partitionMeshes`, used to manage the solid and fluid meshes, `setupFEspace` that handles the FE space, `setupDOF` employed to manage the degrees of freedom of the FSI problem;

- `OseenSolver`: it is a class used to handle the fluid domain physics, by means of the Oseen equations [14].

- `VenantKirchhoffSolver`: it is a class used to handle the structure domain physics within the hypothesis of linear elasticity.

- `FSIMonolithic`: this pure virtual class handles the monolithic solver for FSI problems. The method `updateSystem`, basically, updates the vectors of the previous time steps by shifting on the right their old values, while `solveJac` solves the linearized FSI problem.

- `FSIMonolithicGE`: this class, derived from `FSIMonolithic`, implements the Monolithic Geometry-Convective Explicit solver; in order to solve the geometric sub-problem, as shown in Figure 4.4, the method `evalResidual` recalls the numerical strategies that have been implemented within the class `ALEMeshMotion`. In detail, to face the dynamic mesh handling it is possible to employ either the FFD, or the RBF, or the IDW strategies. Such a choice has to be set within the `data` file.

- `NonLinearRichardson`: this template menages the numerical solution of the linearized coupled FSI problem by means of a Newton algorithm. In detail it firstly calls the method `evalResidual` of the `FSIMonolithicGE` class to solve the geometry problem (4.47) and to compute the initial residual of (4.62); then, it calls the methods `solveJac` and `evalResidual` of the `FSIMonolithic` class in order to reach the convergence for the Newton algorithm. In detail, until Newton algorithm has not reached convergence, at each Newton-iteration the first method computes the Jacobian matrix, while the latter evaluates the residual of the system (4.62).
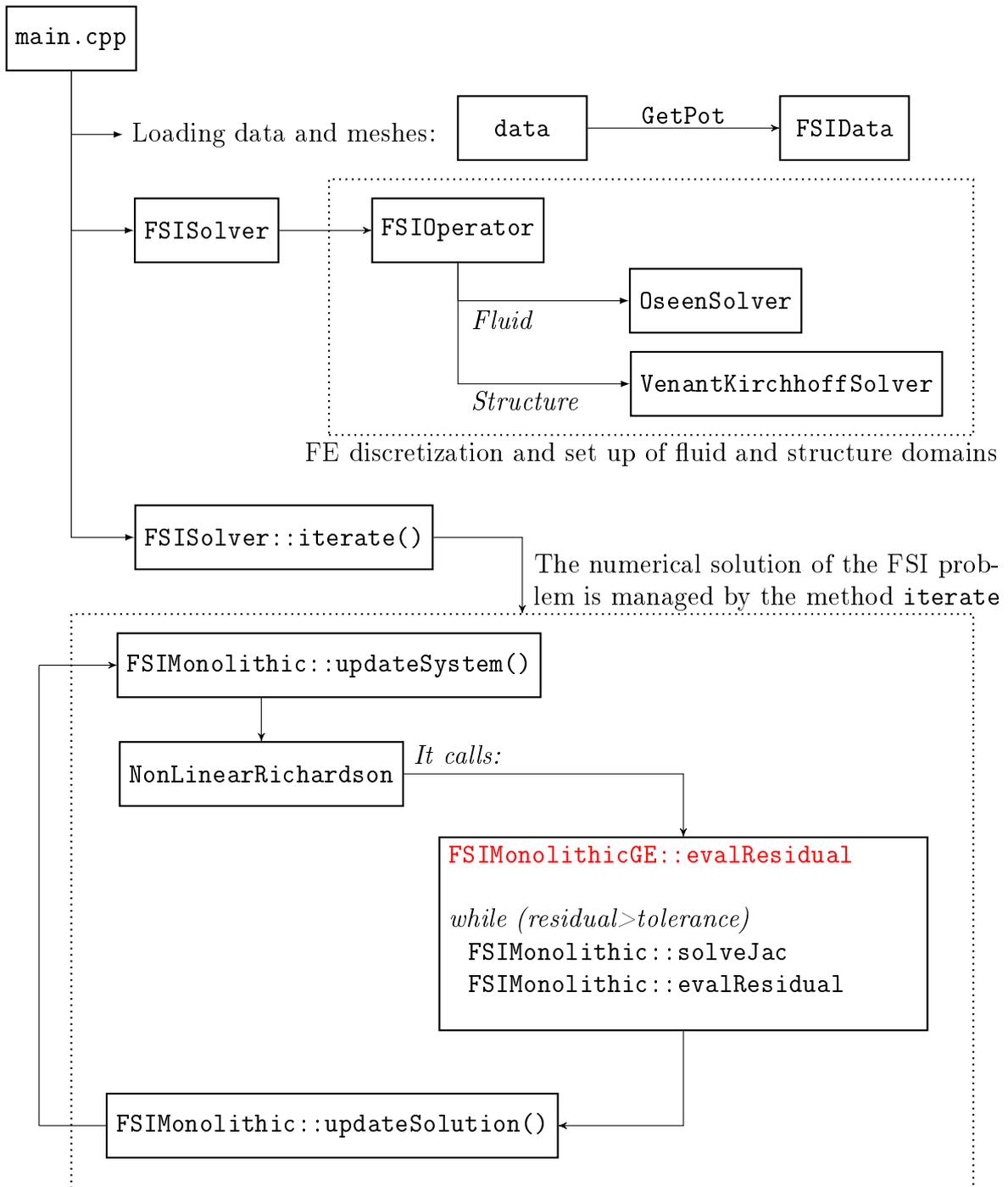
Figure 4.3: Block diagram illustrating the solution strategy of the FSI problem adopted within LifeV. We provide an overview of the whole solution pattern menaged by the `main.cpp` file, thanks to a fully flexible object-oriented code architecture. We indicate, by red color, where the class `ALEMeshMotion` get involved to tackle the dynamic mesh handling.

FSIMonolithicGE::evalResidual

• Solve the geometry (mesh motion) problem (4.47)
by means of FFD, IDW or RBF methods implemented
within the class ALEMeshMotion to compute $\mathbf{d}_F^{n+1}$

• Compute the fluid mesh velocity $\mathbf{w}_F^{n+1}$

• Assemble, with respect to (4.62), the fluid,
$\mathbf{F}$, and the structure, $\mathbf{S}$, blocks of $\mathbf{A}_{FSI}^{n+1}$

• Update, with respect to (4.62), the right-hand-side
due to the applications of boundary conditions

• Solve the system (4.62) and get the first *residual*

Figure 4.4: Block diagram highlighting the numerical procedures employed by the call of the method FSIMonolithicGE::evalResidual; in detail we illustrate schematically the way through we have introduced within LifeV the numerical strategies (by means of the class ALEMeshMotion) that solve the mesh motion problem.

Chapter 5

Simulations and results

In this Chapter we will focus our attention on both the test and the performances evaluation of the numerical strategies proposed to handle the dynamic mesh motion problem within FSI. Here, we aim to validate the mesh handling strategies developed within several examples of FSI problems, dealing with both internal and external fluid flows at low Reynolds numbers; in detail, the purpose of these tests is to verify the applicability of the shape parametrization techniques in such situations involving small and large structural displacements. In this way, by considering fluid flows in laminar regimes and high-elastic structures as well, we will investigate on one hand the robustness of the strategies proposed, while, on the other, we would compare their related computational costs. We remark the fact that, although we will investigate the behavior of shape parametrization methods with respect only to the mesh motion problem, it would be possible to employ the toolbox developed within fluid-structure interface problems too. At the basis of this choice there is the reason that the novelty of this work is mostly related to the application of such parametrization strategies to mesh motion problems. Moreover, since we have carried out, as shown in Chapter 3, several strategies to perform a dimensionality reduction for shape parametrization involved in the mesh motion problem, here we aim to test also those strategies within FSI simulations involving small and large structural displacements. This Chapter has been organized as it follows: in Section 5.1 we will outline the main measures to be adopted and properly monitored to quantify and verify the mesh quality; in Section 5.2 we will propose the main results gained with respect to the FSI simulation of an external laminar fluid flow past an elastic obstacle, and, within this example, it will be possible to highlight the effectiveness of those strategies proposed in Chapter 3, in detail of both the adaptive choice of the RBF interpolation sites, and, concerning the FFD method, its related domain-decomposition approach. In the same way we will propose, in Section 5.3 the results obtained with respect to an FSI simulation dealing with an internal fluid flow.

5.1 Measures of mesh quality

In order to test the mesh handling toolbox developed within FSI simulations, it is worthwhile to survey all the mesh quality measures considered in this work [6, 51, 89]. Concerning the computational grids employed in our several test, we have considered meshes made up of tetrahedral elements. To better assess both the meaning and the definitions of such quality indicators suited for this element type, in Figure 5.1 we depict a simple tetrahedral element:
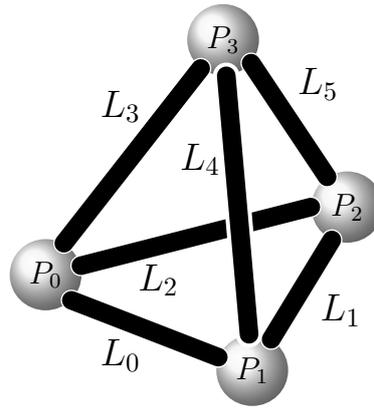
Figure 5.1: Schematic representation of a tetrahedral element.

being $\mathbf{P_i}$ its vertices and $\mathbf{L_i}$ its edges. Furthermore, it is possible to define the following edge vectors:

$$\begin{aligned}
\mathbf{L_0} &= \mathbf{P_1} - \mathbf{P_0}, & \mathbf{L_3} &= \mathbf{P_3} - \mathbf{P_0}, \\
\mathbf{L_1} &= \mathbf{P_2} - \mathbf{P_1}, & \mathbf{L_4} &= \mathbf{P_3} - \mathbf{P_1}, \\
\mathbf{L_2} &= \mathbf{P_0} - \mathbf{P_2}, & \mathbf{L_5} &= \mathbf{P_3} - \mathbf{P_2}.
\end{aligned}$$

In order to outline the mesh quality measures, it is previously necessary to introduce the geometrical entities involved within their evaluations. In this way we can define the tetrahedron edges length as $L_i = \|L_i\|$, whose maximum and minimum values are indicated by $L_{max} = \max(L_i)$ and $L_{min} = \min(L_i)$, respectively. The volume of the element considered can be computed as:

$$V = \frac{(\mathbf{L_2} \times \mathbf{L_0}) \cdot \mathbf{L_3}}{6},$$

while the surface area of such element reads as:

$$A = \frac{1}{2}(\|\mathbf{L_2} \times \mathbf{L_0}\| + \|\mathbf{L_3} \times \mathbf{L_0}\| + \|\mathbf{L_4} \times \mathbf{L_1}\| + \|\mathbf{L_3} \times \mathbf{L_2}\|).$$

Furthermore we can define with R_{in} and R_{out} as the inradius and the circumradius, respectively, of the tetrahedron: in detail they represent the radii of the circumscribed and inscribed spheres of this tetrahedron. Concerning their geometrical definition, they are defined as it follows:

$$
\begin{aligned}
R_{in} &= \frac{3V}{A}, \\
R_{out} &= \frac{\left\| \|\mathbf{L_3}\|^2 \cdot (\mathbf{L_2} \times \mathbf{L_0}) + \|\mathbf{L_2}\|^2 \cdot (\mathbf{L_3} \times \mathbf{L_0}) + \|\mathbf{L_0}\|^2 \cdot (\mathbf{L_3} \times \mathbf{L_2}) \right\|}{12V}.
\end{aligned}
$$

It is possible, now, to outline the mesh quality measures considered in this work. The first is the so-called *edge ratio*, defined as it follows:

$$
\rho = \frac{L_{max}}{L_{min}}. \qquad \text{(Edge Ratio)}
$$

As second measure, we can consider the estimation of the element *distortion*: such a quantity measure would indicate how well-behaved the mapping from parameter space to world coordinates is. In order to perform this test, we can define a master tetrahedron whose vertices are:

$$
\begin{aligned}
\tilde{\mathbf{P}}_0 &= (-1, \frac{-\sqrt{3}}{3}, -\frac{2\sqrt{6}}{9}), \\
\tilde{\mathbf{P}}_1 &= (1, \frac{-\sqrt{3}}{3}, -\frac{2\sqrt{6}}{9}), \\
\tilde{\mathbf{P}}_2 &= (0, \frac{-2\sqrt{3}}{3}, -\frac{2\sqrt{6}}{9}), \\
\tilde{\mathbf{P}}_3 &= (0, 0, \frac{4\sqrt{6}}{9}),
\end{aligned}
$$

and its volume is represented by V_m. In this way, to evaluate the element distortion, we firstly evaluate the Jacobian determinant of the map at the element Gauss quadrature points, $G = \{g_j\}$; thus, by considering its minimum value, we get:

$$
\delta = \frac{\min_j\{\det(J(g_j))\}V_m}{V}. \qquad \text{(Distortion)}
$$

As third mesh quality measure we considered the *radius ratio*, defined as the quotient of the two radii previously introduced, R_{out} and R_{in}, normalized by $\frac{1}{3}$, so that

an equilateral tetrahedron (such that $\{L_i\}_{i=0}^5 = \bar{L}$) has this measure equal to one:

$$r = \frac{R_{out}}{3R_{in}}. \hspace{3cm} \text{(Radius Ratio)}$$

A further way to assess the mesh quality relies in the evaluation of the *minimum dihedral angle*: it is defined, in degrees, as the angle between two faces of the tetrahedron that are adjacent along the edge $\mathbf{L_i}$:

$$\beta_i = \frac{180°}{\pi} \arccos(\mathbf{n_{i_1}} \cdot \mathbf{n_{i_2}}),$$

being $\mathbf{n_{i_1}}$ and $\mathbf{n_{i_2}}$ the normal versors to the two tetrahedron faces that are adjacent to i-th edge. In this way the minimum dihedral angle reads as:

$$\beta = \min_i \beta_i. \hspace{3cm} \text{(Minimum Dihedral Angle)}$$

Another strategy to evaluate the mesh quality, for tetrahedral elements, is related to the measure of the *shape* metric. This quality criterion measures the likeness of the element to the reference one (e.g. an equilateral tetrahedron). Its value is equal to one for a perfect element, while it decreases as the element becomes of worst quality. We remark the fact that, in the case when shape metric reaches negative values, it means that at some points negative Jacobian values has been reached as well. The shape quality criterion is held by the following expression:

$$\sigma = \frac{6\sqrt{2}V}{\sum_{i=0}^{i=5} L_i^3}. \hspace{3cm} \text{(Shape)}$$

Finally we need to check that the elements *volume* do not assume negative values during the simulation process:

$$V > 0. \hspace{3cm} \text{(Volume)}$$

Moreover, we need to evaluate the elements volume reduction, at the basis of the fact that excessive contractions would address to bad mesh quality. Table 5.1 summarizes the main mesh quality measures considered: we indicate also, for each of them, an acceptable range that would represent good elements quality:

Criterion	Values for equilateral tetrahedron	Acceptable range
(Edge Ratio)	1	1 - 3
(Distortion)	0	0.5 - 1
(Radius Ratio)	1	1 - 3
(Minimum Dihedral Angle)	70.53°	40° - 70.53°
(Shape)	1	0.3 - 1
(Volume)	0.1179	positive

Table 5.1: Mesh quality measures considered and their related acceptable ranges.

At the basis of the mesh quality measures introduced, together with the scaled jacobian indicator as well (whose acceptable range is 0.4 - 0.8), throughout Sections 5.2 and 5.3 we will validate all the mesh motion strategies proposed within the solution of many FSI problems.

5.2   External fluid flow past flexible obstacle

In this example we will compare the techniques developed to tackle the mesh motion problem within a FSI simulation of an external fluid flow past a flexible obstacle, whose numerical solution has been obtained by using the FE library LifeV. Such an example illustrates how the fluid flow can deform structures and how to solve the flow in a continuously deforming geometry using the Arbitrary Lagrangian-Eulerian approach. The model geometry consists of a horizontal flow channel in the middle of which there is an obstacle, a narrow vertical structure. The fluid flows from left to right, except where the obstacle forces it into a narrow path in the upper part of the channel, and it imposes a force on the structure's walls resulting from the viscous drag and fluid pressure. The structure, being made of a deformable material, bends under the applied load. Consequently, the fluid flow also follows a new path, so solving the flow in the original geometry would generate incorrect results. The ALE method handles the dynamics of the deforming geometry and the moving boundaries with a moving grid. Thanks to mesh motion strategies developed we compute new mesh coordinates on the channel area based on the movement of the structure's boundaries and mesh smoothing. The Navier-Stokes equations that solve the flow are formulated for these moving coordinates. Concerning the geometry of the problem to be studied, we considered as dimensions for the flow channel a height of 8 mm, a width of 7 mm, and a length of 20 mm. The vertical structure is 1 mm wide (with a square cross section) and 5 mm high. The fluid is a water-like substance with a density $\rho_F = 0.998 \; g \, cm^{-3}$ and dynamic viscosity $\mu = 0.01 \; poise$. In order to study the behavior of the different mesh handling strategies proposed, we assume that the

structure consists of a flexible material with a density $\rho_S = 2.7 \ g \, cm^{-3}$ and a Young modulus $E = 4E + 6 \ dyne \, cm^{-2}$. For the Finite Elements discretization we considered $\mathbb{P}_1$ elements for the structural displacement and, also, for both the fluid pressure and velocity variables, stabilized by Interior Penalty (IP) [21, 22].
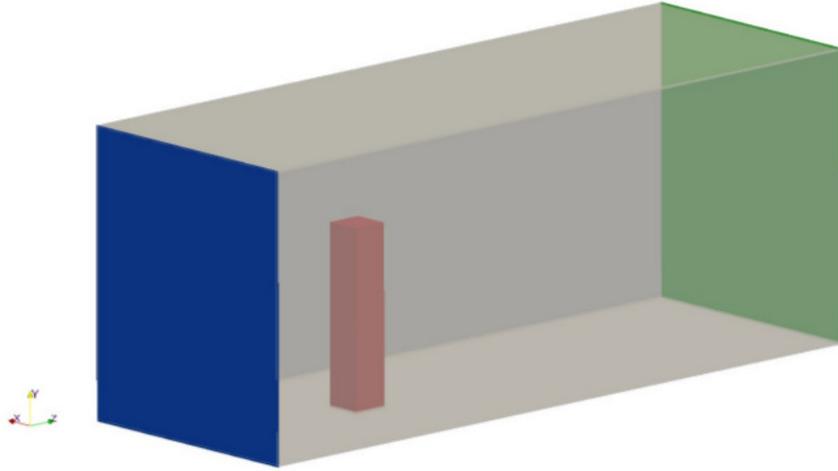
Figure 5.2: Visualization of the domain geometry: we indicated, by different colors, all the different surfaces related to the boundary conditions of the FSI problem.

With respect to Figure 5.2, the boundary conditions applied to the problem read as:

- fluid domain, inlet (blue): parabolic inflow velocity profile;

- fluid domain, outlet (green): stress free;

- fluid domain, walls (grey): no-slip condition, $\mathbf{u} = \mathbf{0}$;

- fluid-structure domains, interfaces (red): coupling conditions of velocities (2.21) and stresses (2.22);

- structure domain, basis: clamped, $\hat{\mathbf{d}}_S = \mathbf{0}$;

with respect to the notation that we have previously introduced (Chapter 4). Regarding the simulation time, we considered $t_0 = 0 \, s$, $t_N = 0.03 \, s$ with a time-step $\Delta t = 0.001 \, s$. Concerning the parabolic inflow velocity profile, by indicating with H and W the flow channel height and width, respectively, its analytical expression is defined as it follows:

$$\tilde{\mathbf{v}}_{\mathbf{F_{inflow}}} = v_{max} \left(1 - \frac{(x - \frac{W}{2})^2 + (y - \frac{H}{2})^2}{(\frac{W}{2})^2 + (\frac{H}{2})^2} \right) \tag{5.1}$$

being $v_{max} = 100 \, cm \, s^{-1}$. In this way, during the simulation process, the inflow velocity profile has been set as:

$$\mathbf{v_{F_{inflow}}} = \begin{cases} \tilde{\mathbf{v}}_{\mathbf{F_{inflow}}} \, 100 \, t, & \text{if } t \le 0.01 \, s \\ \tilde{\mathbf{v}}_{\mathbf{F_{inflow}}}, & \text{if } t > 0.01 \, s \end{cases} \tag{5.2}$$

such that for $t \ge 0.1 \, s$ we have at the inlet surface, for $z = 0 \, mm$, a velocity profile that reaches its maximum value $\mathbf{v_{F_{inflow}}}(\frac{W}{2}, \frac{H}{2}, 0) = v_{max}$. Thanks to the simulation settings introduced, we will deal with a laminar fluid flow, since the Reynolds number is equal to $\mathbb{R}e = 670$, by considering, as velocity, $U = \frac{2}{3}v_{max}$. With respect to Figure 5.3, we illustrate the meshes considered for both the fluid (left) and the structural (right) domains.
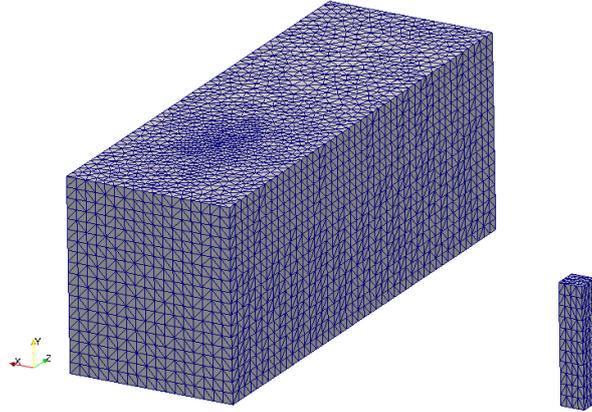
Figure 5.3: Computational meshes realized for the fluid (left) and the structural (right) domains.

In Table 5.2 we provide the details of the elements that compose the meshes:

	Nodes	Triangles	Thetrahedra
Fluid domain	19989	8732	106422
Structure domain	336	428	1254

Table 5.2: Details of the computational grids realized.

With respect to Figure 5.4, we report the main results obtained for this example, in terms of both the structural displacement and the fluid flow velocity. On the basis of the physical results illustrated, obtained using as mesh motion strategy the Inverse Distance Weighting technique, we can report, also, the main

results related to the mesh quality measures, with respect to other mesh motion strategies (RBF and Laplacian).

(a) $t = 0.003 \, s$.

(b) $t = 0.006 \, s$.

(c) $t = 0.010 \, s$.

(d) $t = 0.015 \, s$.

(e) $t = 0.020 \, s$.

(f) $t = 0.025 \, s$.

(g) $t = 0.028 \, s$.

(h) $t = 0.030 \, s$.
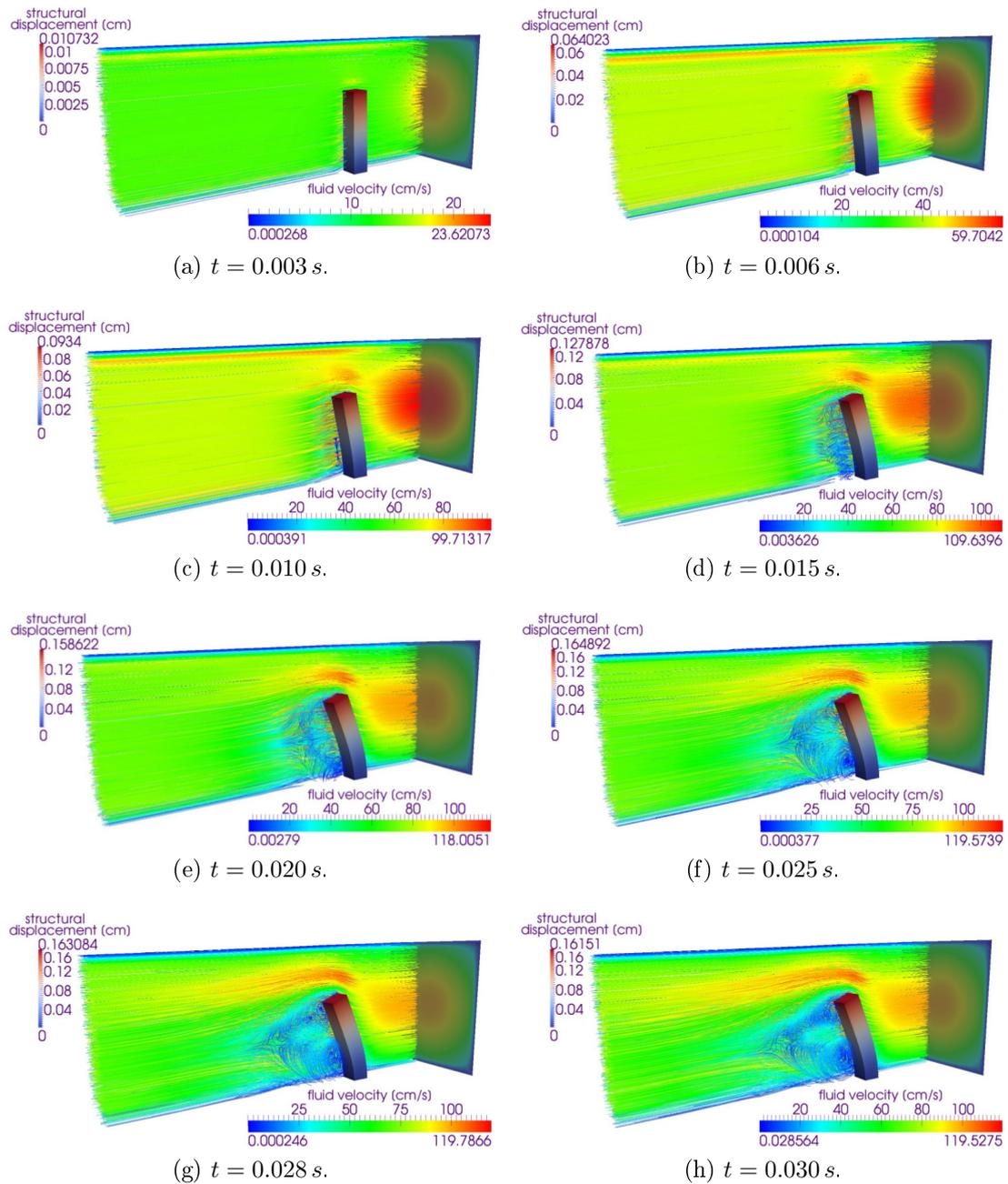
Figure 5.4: Visualization of the simulation results: we illustrate the structural displacement contour, the fluid flow velocity streamlines and, finally, the parabolic inflow velocity profile ($\mathbb{R}e = 670$).
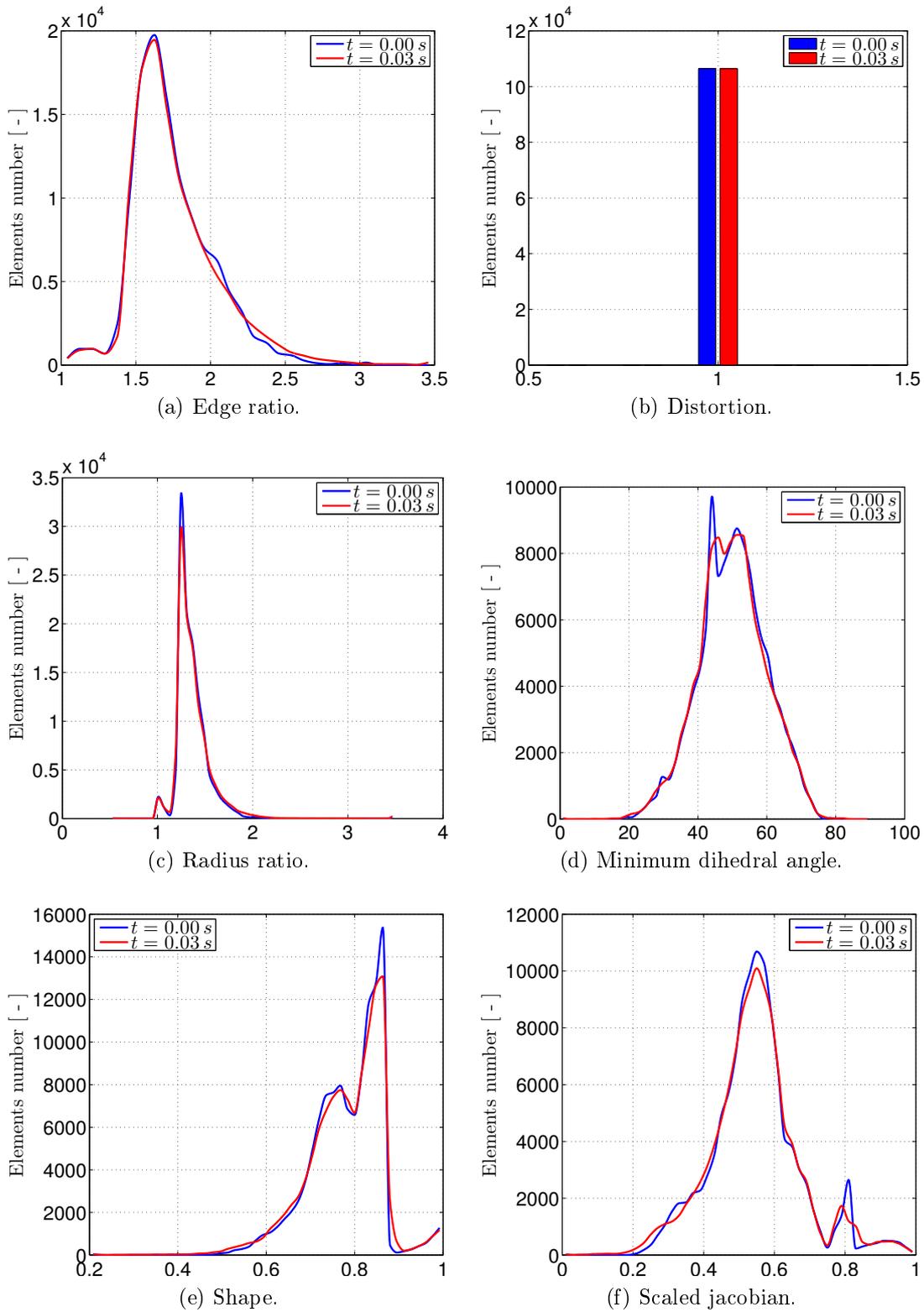
Figure 5.5: Mesh quality measures obtained using the IDW mesh motion strategy.
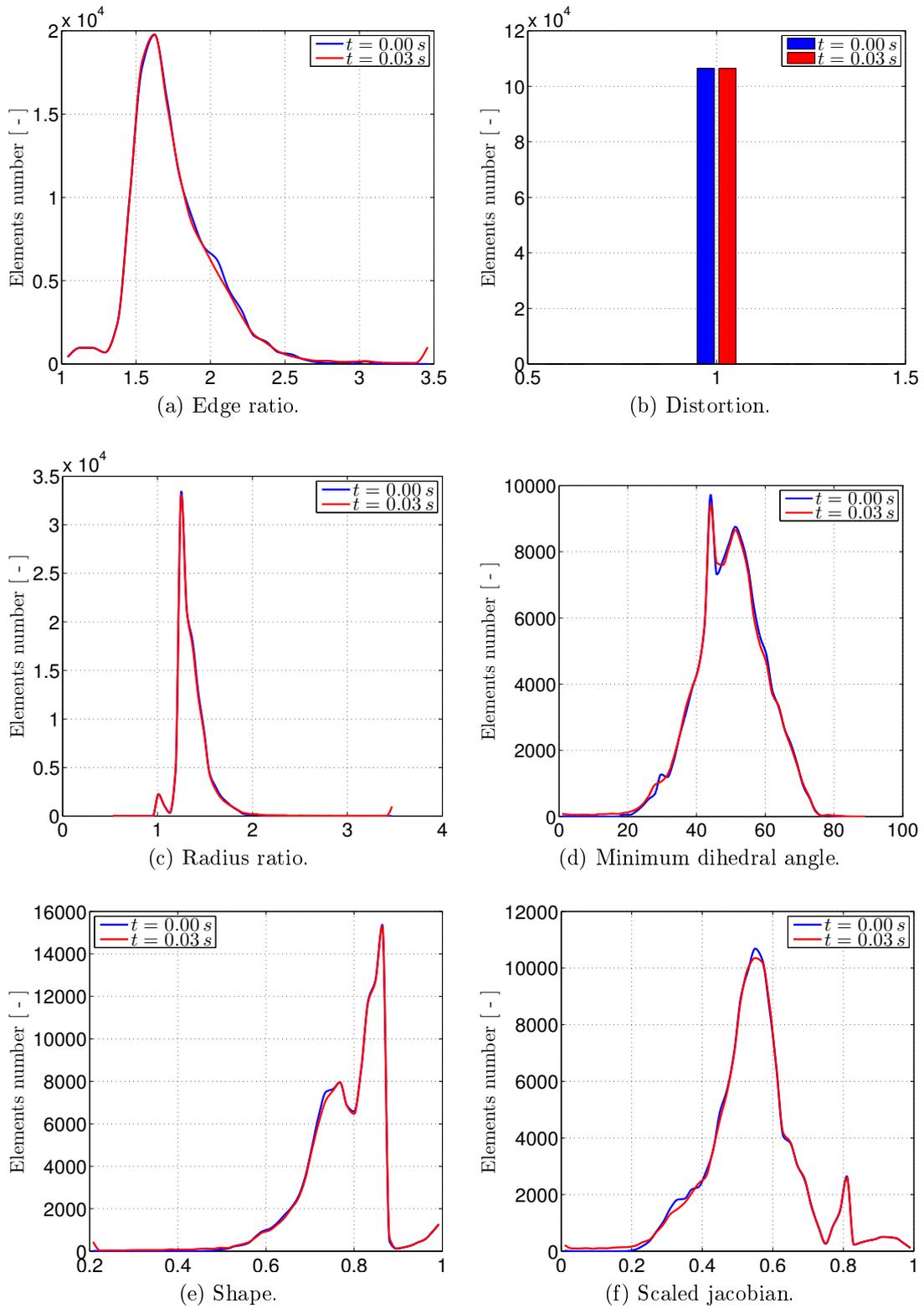
Figure 5.6: Mesh quality measures gained using the RBF mesh motion strategy.
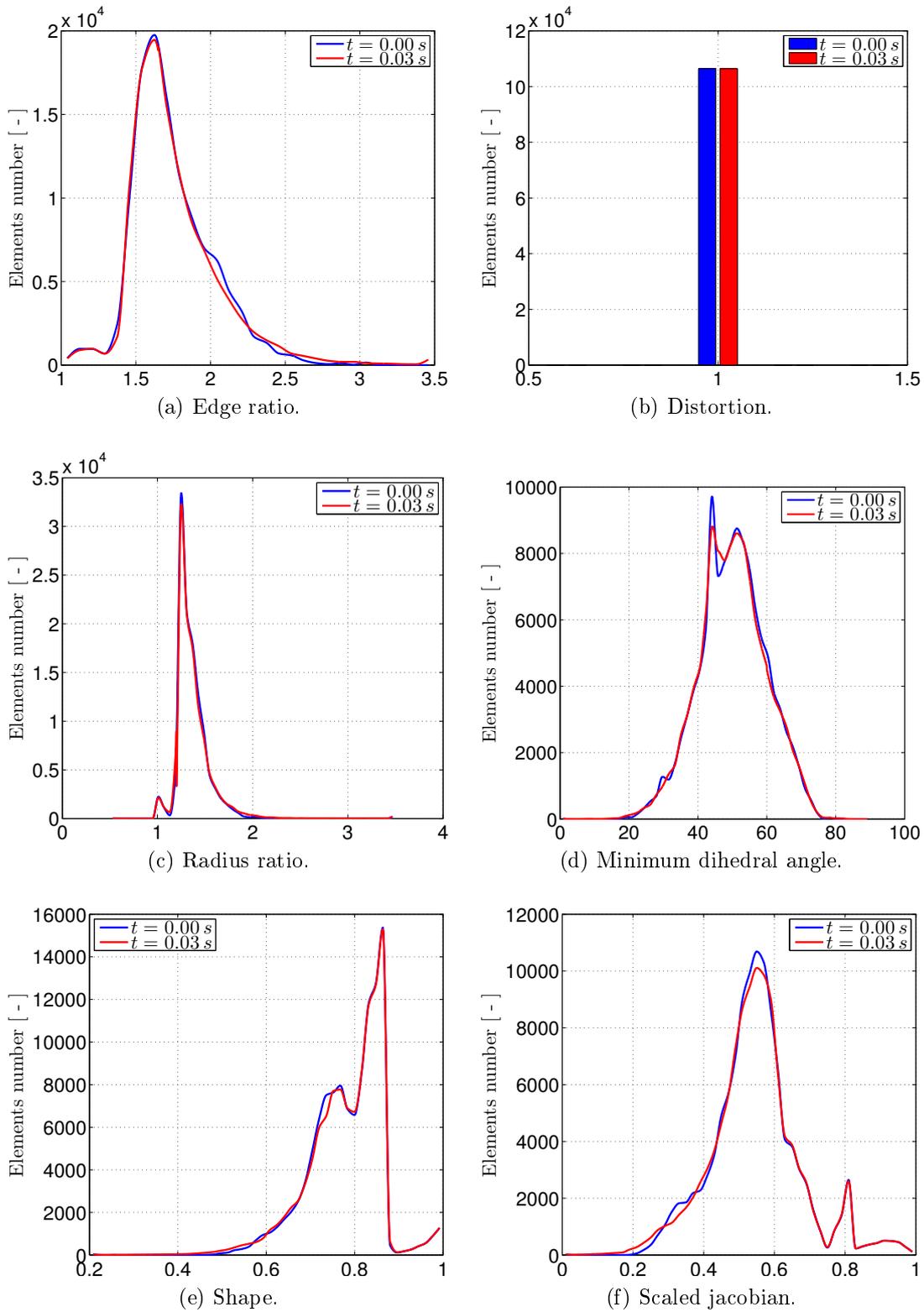
Figure 5.7: Mesh quality measures gained using the Laplacian mesh motion strategy.

Among the mesh motion strategies considered to solve such an example, we dealt with the RBF method based on the reduction of the interpolation sites: to this end we illustrate in Figure 5.8 the first three structural eigenmodes computed for the elastic beam, and, moreover, we show the main convergence results obtained in order to locate the control points over the structure.

(a) First eigenmode.   (b) Second eigenmode.   (c) Third eigenmode.
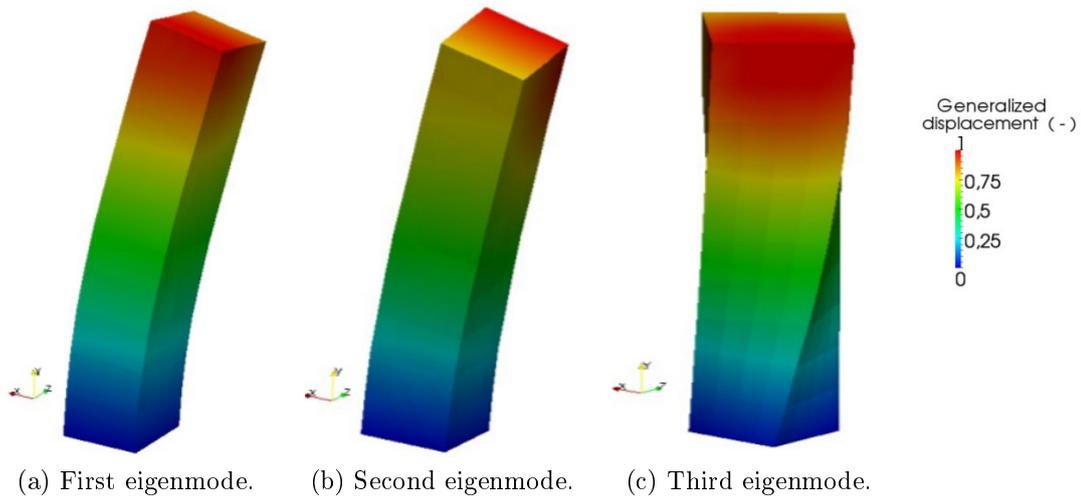
Figure 5.8: Visualization of the first three structural eigenmodes computed for the identification of the RBF interpolation sites.

(a) Cost functional behavior vs iterations number.

(b)   Interpolation sites.
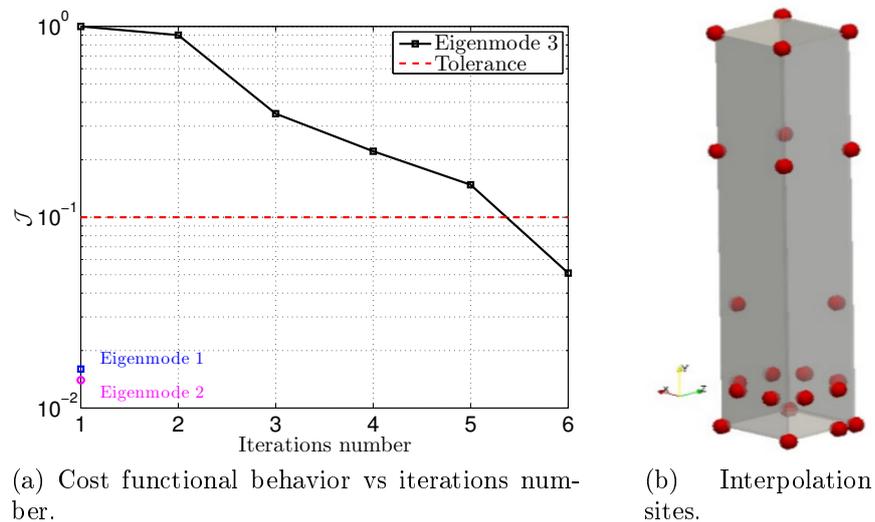
Figure 5.9: Visualization of both the control points sites over the initial structural shape, and the greedy algorithm convergence.

With reference to Figure 5.9, once we fixed for the greedy algorithm a tolerance

value of $TOL = 10E - 1$, we obtained as final number of control points $N_C = 19$. In detail, since we have considered in our bandwidth of interest the first three structural eigenmodes, as schematically illustrated in Figure 3.8 we started by recovering the third one: to this end we have found that 19 control points were enough to recover such eigenmode. Thus, with this set of parameters found we were also able to capture the second and the first eigenmodes (see Figure 5.9a), since for $i_2 = 1$ and $i_1 = 1$, respectively, the initial value of the cost functional $\mathcal{J}$ (defined in (3.17)) was already lower than the fixed tolerance. In that way it will be possible to reduce, for the RBF mesh motion strategy involved in the FSI simulation, the number of control points since we can consider, over the the fluid-structure interface, not the total structural nodes number, $N_\Gamma^S = 188$, but only some of them. Regarding the FFD technique we considered only its related domain-decomposition approach: this choice is due to the fact that in the FFD original framework, the constraint held by placing the control points into a regular lattice within the whole fluid domain would drive on one hand to unacceptable computational costs, on the other to a lack in accuracy. In Figure 5.10 we illustrate in green color the part of the domain involved in the FFD mapping and mesh motion procedure.
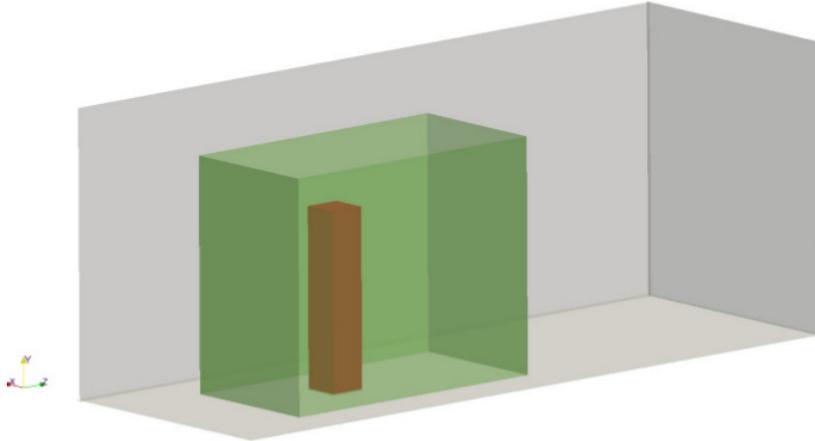
Figure 5.10: Representation, with green surfaces, of the local 3D box embedding the structure domain (red), to be mapped through the FFD technique.

In detail, as dimensions for the green patch we considered a width of $4\,mm$, a height of $6.5\,mm$ and, as length, $8\,mm$. Concerning the number of control points, we employed within the patch 18 points in the z direction, 4 in x and 10 in the y ones, respectively. At the basis of the analysis pattern introduced for both the FFD domain-decomposition approach and the RBF strategy involving a reduced set of the control points, with respect to Figure 5.11 and 5.12 we illustrate the main results concerning the comparison of the mesh quality measures (indicated in Table 5.1) between the FSI simulation time of $t = 0.03\,s$, related to the maximum structural displacement, and of $t = 0\,s$.

(a) Edge ratio.

(b) Distortion.

(c) Radius ratio.

(d) Minimum dihedral angle.

(e) Shape.

(f) Scaled jacobian.
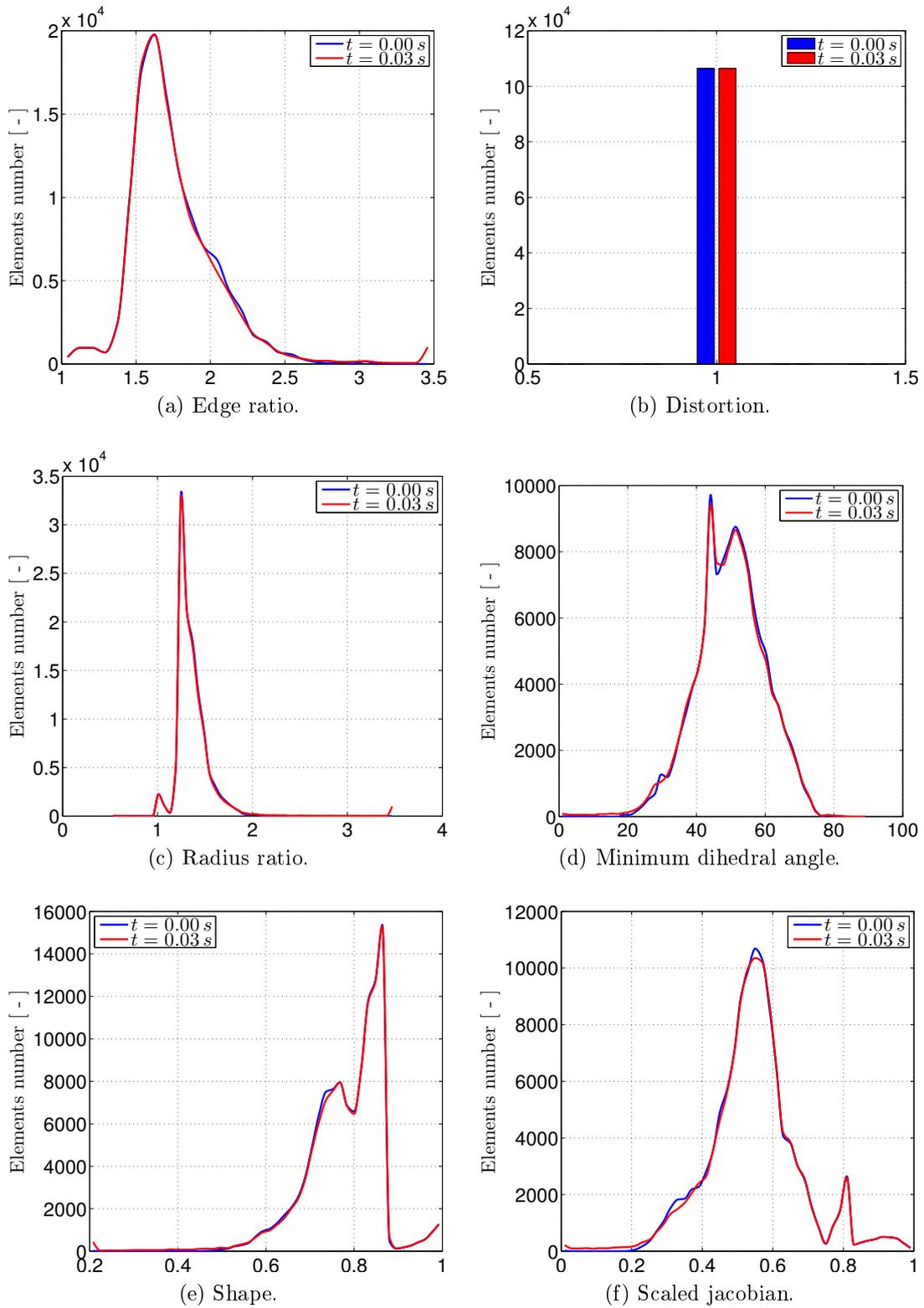
Figure 5.11: Mesh quality measures gained using the RBF mesh motion strategy with a reduced set of control points.
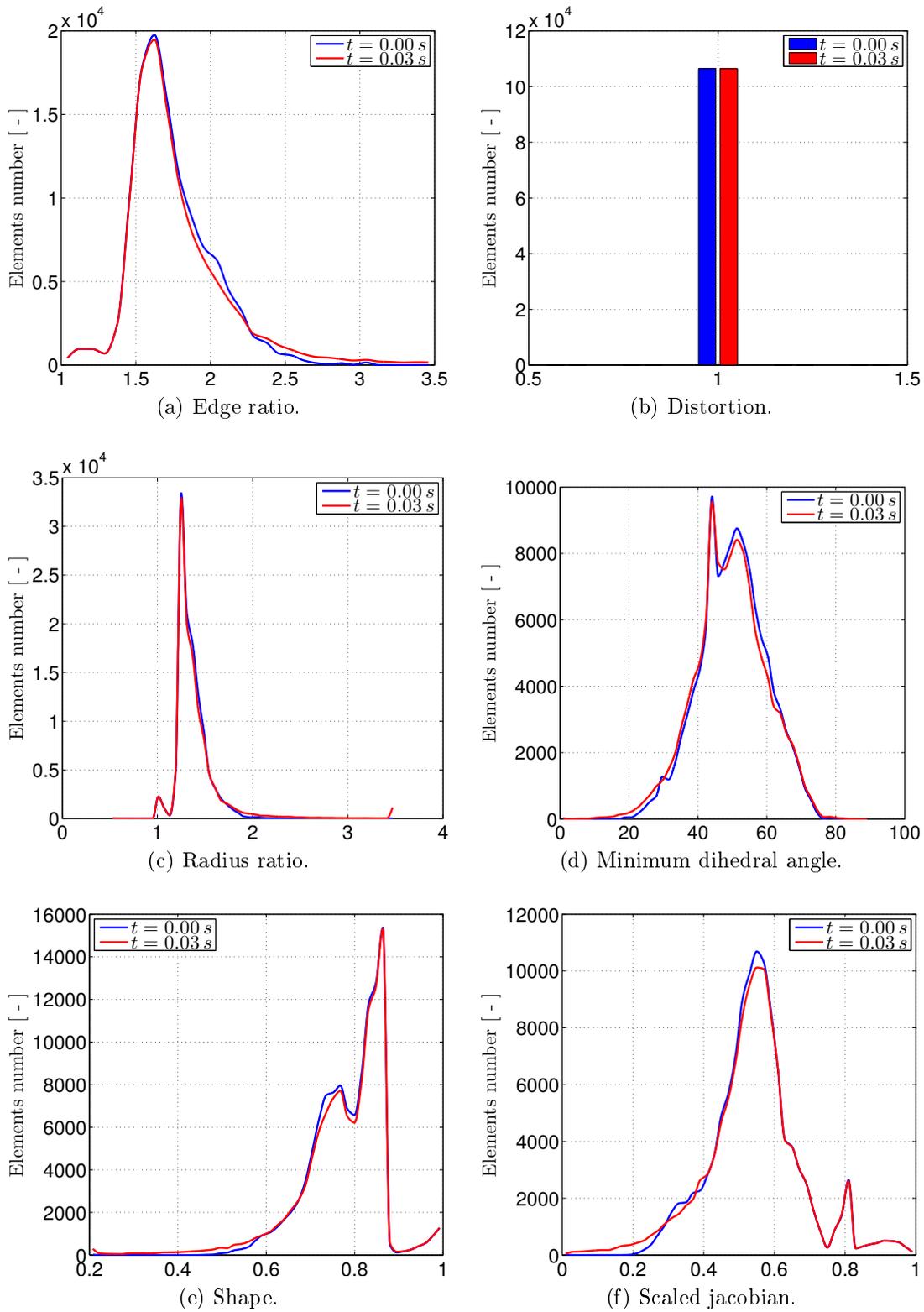
Figure 5.12: Mesh quality measures gained using the FFD domain-decomposition approach mesh motion strategy.

Regarding the mesh quality measures, reported in Figures 5.5, 5.6, 5.7, 5.11 and 5.12, we did not show the curves related to the Volume measure: during the simulations performed, involving the different mesh motion strategies, it has never occurred to find negative volume values; for that reason we decided to report all the others mesh quality measures. At this point, to better asses the accuracy of the results achieved thanks to the adoption of such mesh motion strategies proposed, in Figure 5.13, we report a comparison between the structural tip displacements gained thanks to each shape parametrization technique:
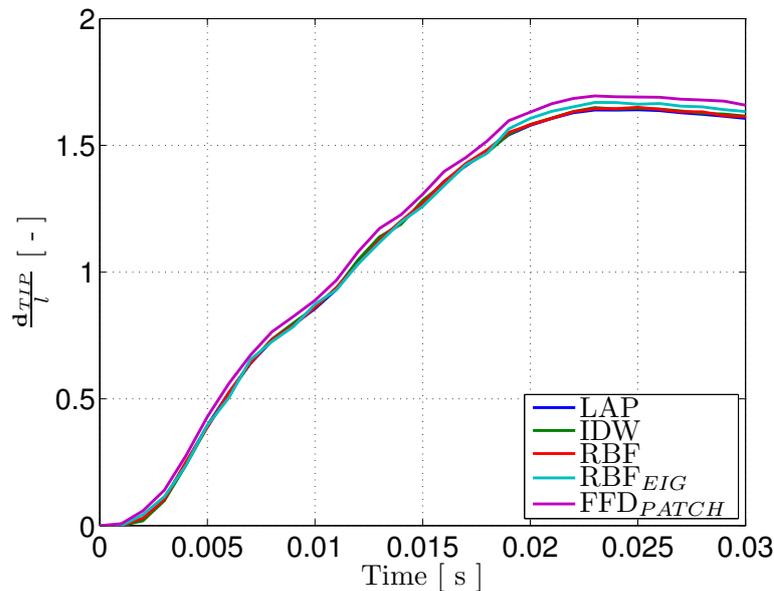
Figure 5.13: Comparison of the structural tip displacements obtained within different mesh motion strategies.

Figure 5.13 illustrates the dimensionless structural tip displacements, obtained as ratio between the physical tip displacements and the structural cross section ($l = 1\,mm$), versus the simulation time. Moreover, within the above Figure, we indicated by RBF_{EIG} and FFD_{PATCH}, the RBF strategy involving the reduced set of control points and the FFD domain-decomposition approach, respectively. At the basis of the results achieved, we can assess the accuracy of the strategies proposed: in fact, all the techniques employed show the same behavior, a part from the FFD domain-decomposition approach; nevertheless, although it is not in perfect accord with respect to the other strategies, we can state that, if on one hand it differs from the others for very small values, on the other hand it is not interpolatory, leading to a lack in accuracy. Concerning the costs, in Table 5.3 we report the computational efforts related to each mesh motion strategy employed: in detail, we show the mean time needed, for each FSI simulation time-step, to update the fluid mesh configuration on the basis of the structural

displacement computed at each FSI iteration. From a mesh quality point of view, all the strategies tested do not show significant drops in the quality measures, since they are kept, above all, within their acceptable range (reported in Table 5.1). As conclusion of this first example, we can state that the IDW technique represents a good compromise between the accuracy of the achieved results, in terms of mesh quality, and the computational effort, such that it uphold the results gained with respect of the several test performed in Chapter 2.

Method	Computational time [s]
Laplacian	18.11
IDW	11.92
RBF	32.01
RBF$_{EIG}$	24.19
FFD$_{PATCH}$	92.18

Table 5.3: Computational costs involved in the mesh motion procedure within the FSI external fluid flow example.

5.3 Internal fluid flow in a cylindrical straight vessel

In this Section we will show the main results obtained in the study of a laminar fluid flow inside a cylindrical straight vessel. The aim of this second example is to study the behavior of the mesh motion techniques proposed, in detail IDW, FFD and RBF strategies, and compare them with a Laplacian based method. In this way we will firstly show the results obtained by adopting as fluid and structural meshes the discretizations whose characteristics are reported in Table 5.4, then, by considering other coarser and finer discretizations, (see Table 5.5) we will evaluate the computational costs. Concerning the geometry of the fluid domain, we considered for the undeformed straight cylinder a radius $R = 0.5\,cm$ and a length $L = 5\,cm$; for the structure domain, embedding the fluid one, we chose an outer radius $R_{out} = 0.6\,cm$, such that its thickness is $t = 0.1\,cm$. Moreover, for the fluid domain, we considered a density $\rho_F = 1.0\,g\,cm^{-3}$ and a dynamic viscosity $\mu_F = 0.03\,poise$. The structure is characterized by a Young modulus $E = 6\,E + 6\,dyne\,cm^{-2}$, by a density $\rho_S = 1.2\,g\,cm^{-3}$ and by a Poisson ratio of $\nu = 0.45$. Concerning the time discretization of the problem we have chosen as time-step $\Delta t = 0.001\,s$, while the simulation time is $t_N = 0.01\,s$. For the Finite Element discretization we considered $\mathbb{P}_1$ elements for both the structural displacement, and the velocity and pressure fluid quantities, stabilized by Interior

Penalty (IP) [21, 22]. The boundary conditions applied to the problem are defined as it follows:

- fluid domain, inlet: flux imposed;

- fluid domain, outlet: stress free;

- fluid-structure domains, interfaces: coupling conditions of velocities (2.21) and stresses (2.22);

- structure domain, basis: embedded;

- structure domain, outer surface: stress free;
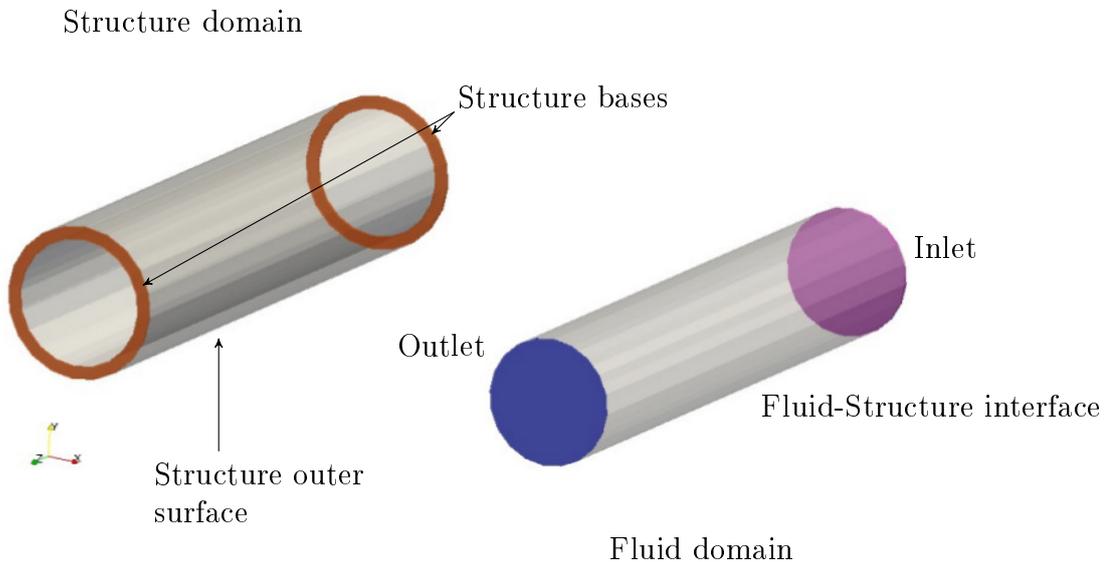
Figure 5.14: Visualization of all the different surfaces related the assignment of the boundary conditions of the problem.

Concerning the inlet boundary condition implemented for the fluid domain, we have imposed a variable flux value characterized by a behavior, that, with respect to the simulation time, is represented in Figure 5.15.
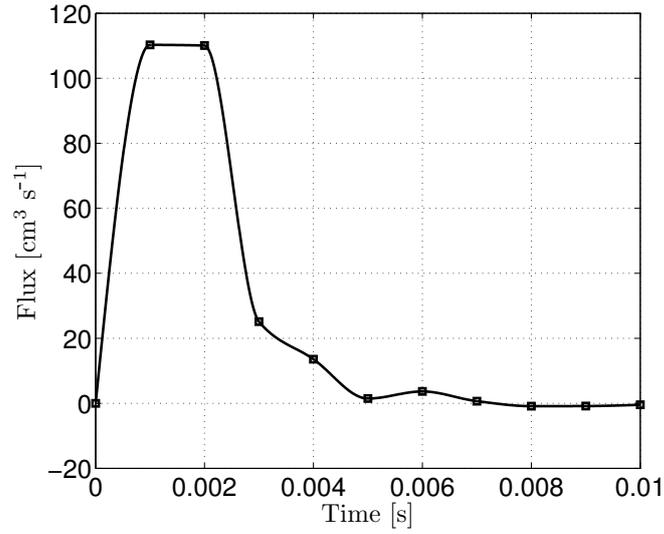
Figure 5.15: Visualization of the inlet flux profile, and its samples, considered for the FSI simulation.

For this example we considered a structured mesh for the fluid and the structure domains, whose properties are reported in Table 5.4:

Domain	Vertices	Triangles	Thetrahedra	Total elements
Fluid	1029	952	4560	5512
Structure	840	1680	2400	4080

Table 5.4: Details of the mesh properties for the fluid and the structure domains.

In Figures 5.16, 5.17, 5.18 and 5.19 we illustrate some snapshots of the structural deformed configuration at different simulation time-steps, gained by using as mesh motion strategies IDW, RBF, FFD and Laplacian techniques, respectively; through Figures 5.20 and 5.21, the fluid pressure and velocity contour plots are visualized within the application of IDW strategy. Finally, with respect to Figures 5.22, 5.23, 5.24 and 5.25 we evaluate the mesh quality measures obtained thanks to the different mesh motion strategies considered.
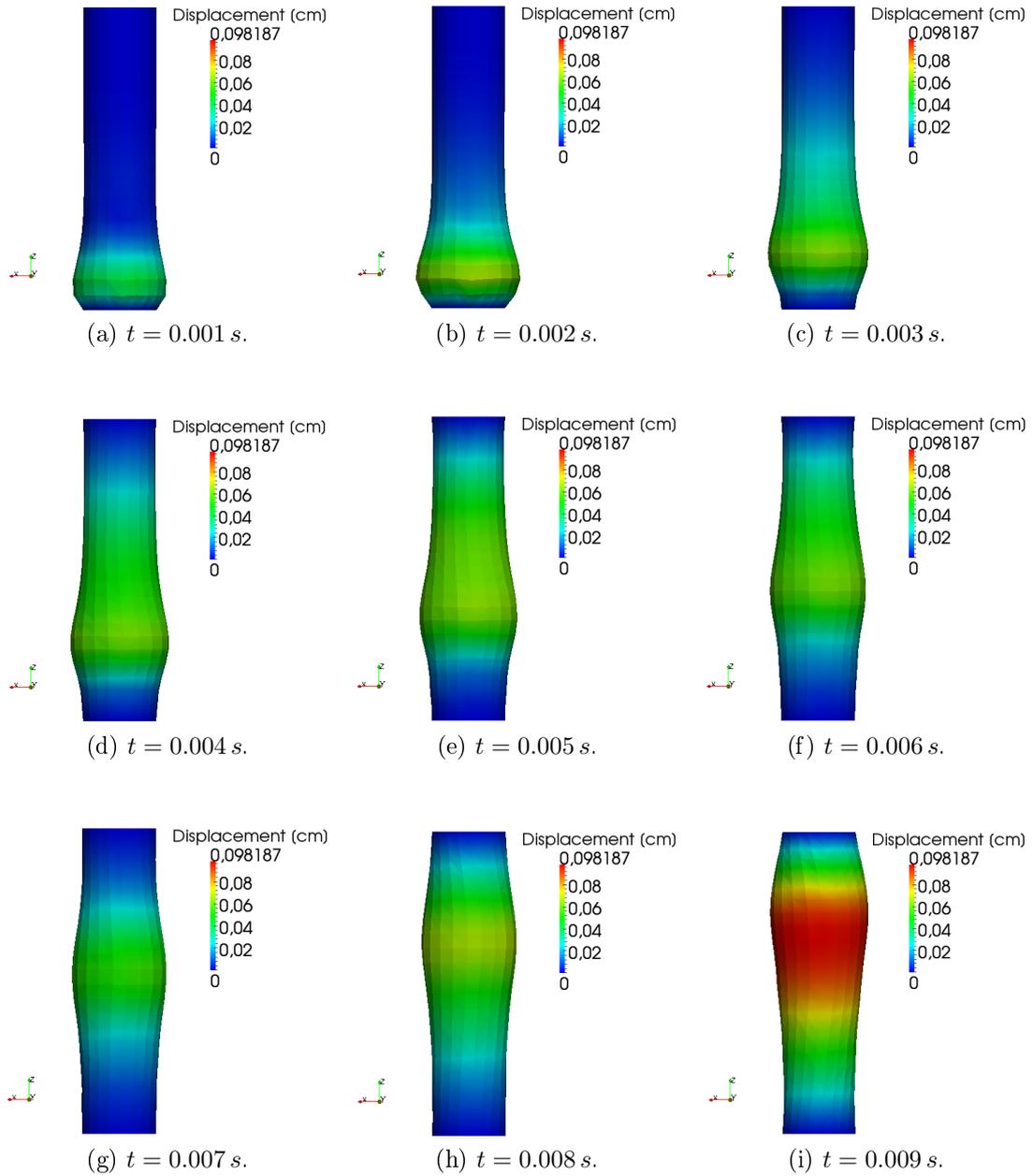
Figure 5.16: Visualization of the structural deformed configurations at different simulation time-steps obtained using, as mesh motion technique, the IDW strategy.

(a) $t = 0.001\,s$.

(b) $t = 0.002\,s$.

(c) $t = 0.003\,s$.

(d) $t = 0.004\,s$.

(e) $t = 0.005\,s$.

(f) $t = 0.006\,s$.

(g) $t = 0.007\,s$.

(h) $t = 0.008\,s$.

(i) $t = 0.009\,s$.
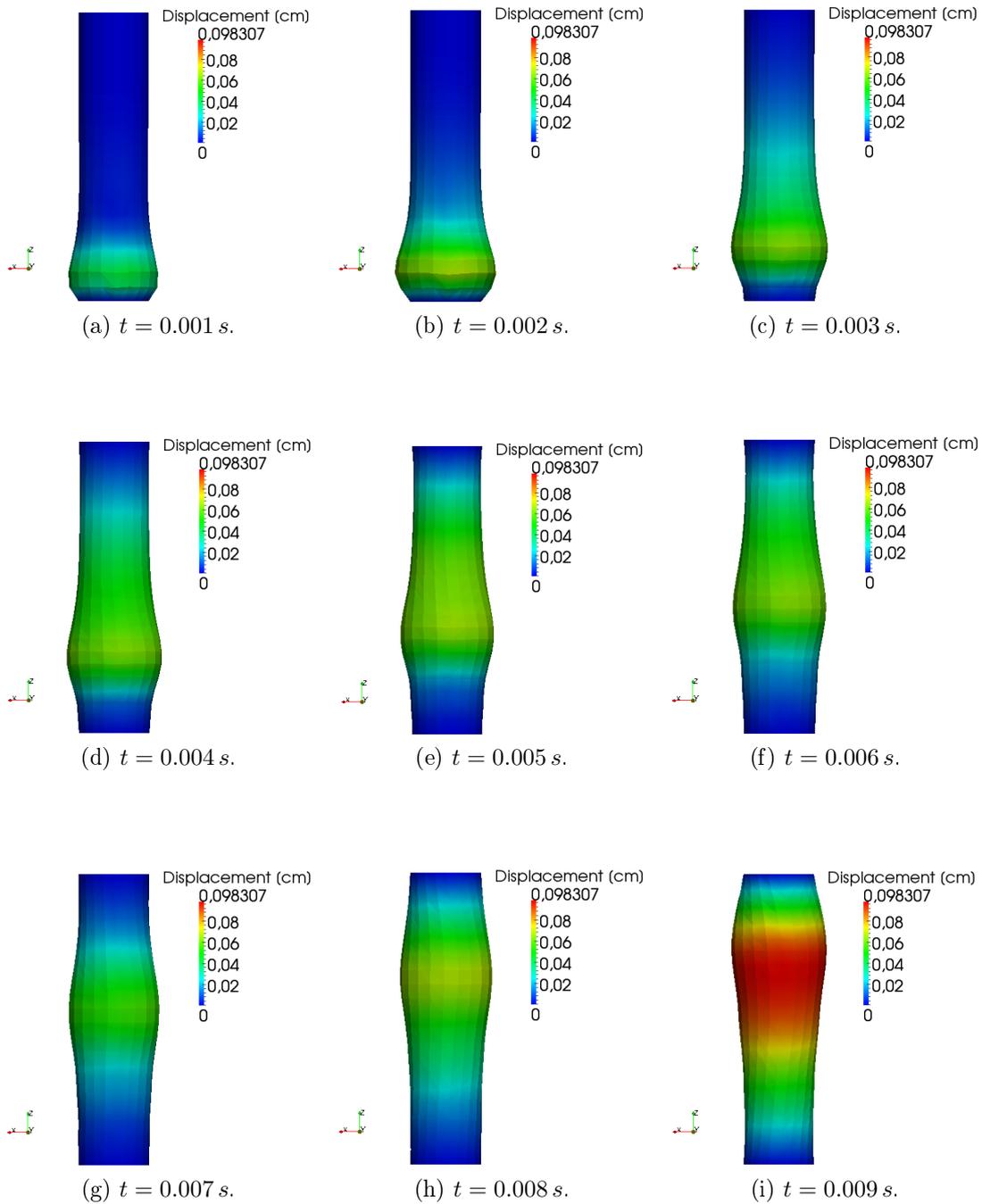
Figure 5.17: Visualization of the structural deformed configurations at different simulation time-steps obtained using, as mesh motion technique, the RBF strategy.

(a) $t = 0.001\,s$.

(b) $t = 0.002\,s$.

(c) $t = 0.003\,s$.

(d) $t = 0.004\,s$.

(e) $t = 0.005\,s$.

(f) $t = 0.006\,s$.

(g) $t = 0.007\,s$.

(h) $t = 0.008\,s$.

(i) $t = 0.009\,s$.
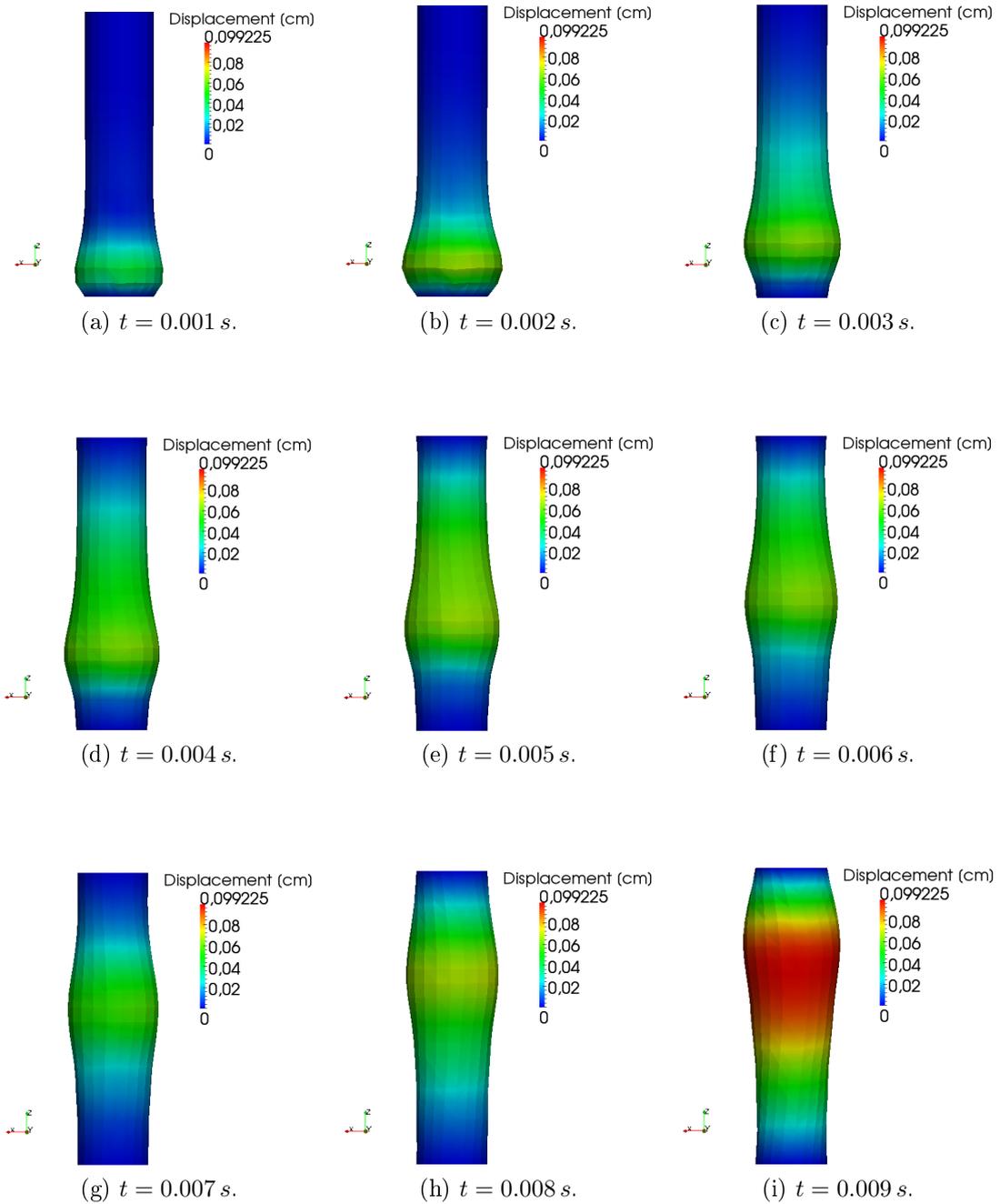
Figure 5.18: Visualization of the structural deformed configurations at different simulation time-steps obtained using, as mesh motion technique, the FFD strategy.

(a) $t = 0.001\,s$.

(b) $t = 0.002\,s$.

(c) $t = 0.003\,s$.

(d) $t = 0.004\,s$.

(e) $t = 0.005\,s$.

(f) $t = 0.006\,s$.

(g) $t = 0.007\,s$.

(h) $t = 0.008\,s$.

(i) $t = 0.009\,s$.
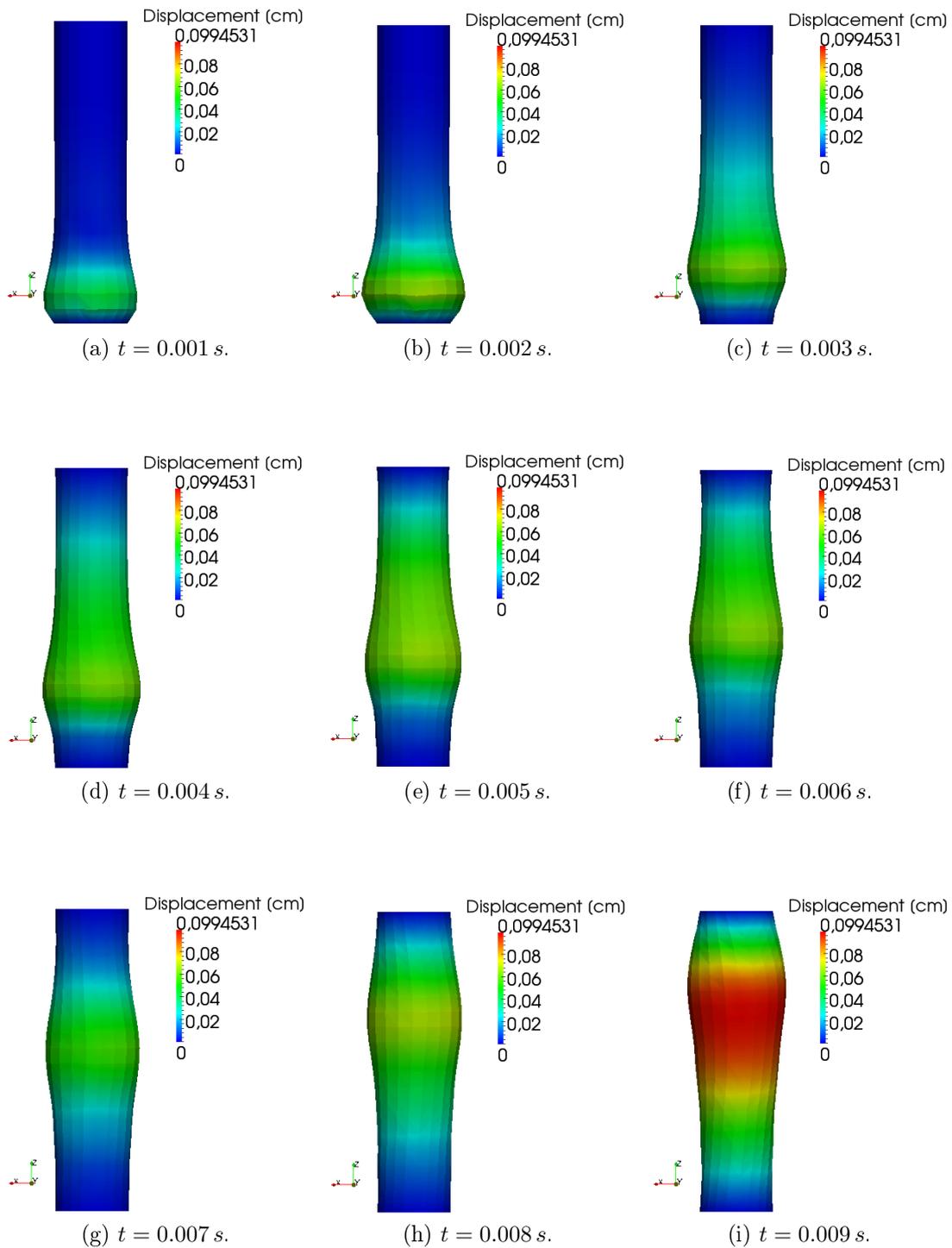
Figure 5.19: Visualization of the structural deformed configurations at different simulation time-steps obtained using, as mesh motion technique, the Laplacian strategy.

(a) $t = 0.001\,s$.

(b) $t = 0.002\,s$.

(c) $t = 0.003\,s$.

(d) $t = 0.004\,s$.

(e) $t = 0.005\,s$.

(f) $t = 0.006\,s$.

(g) $t = 0.007\,s$.

(h) $t = 0.008\,s$.
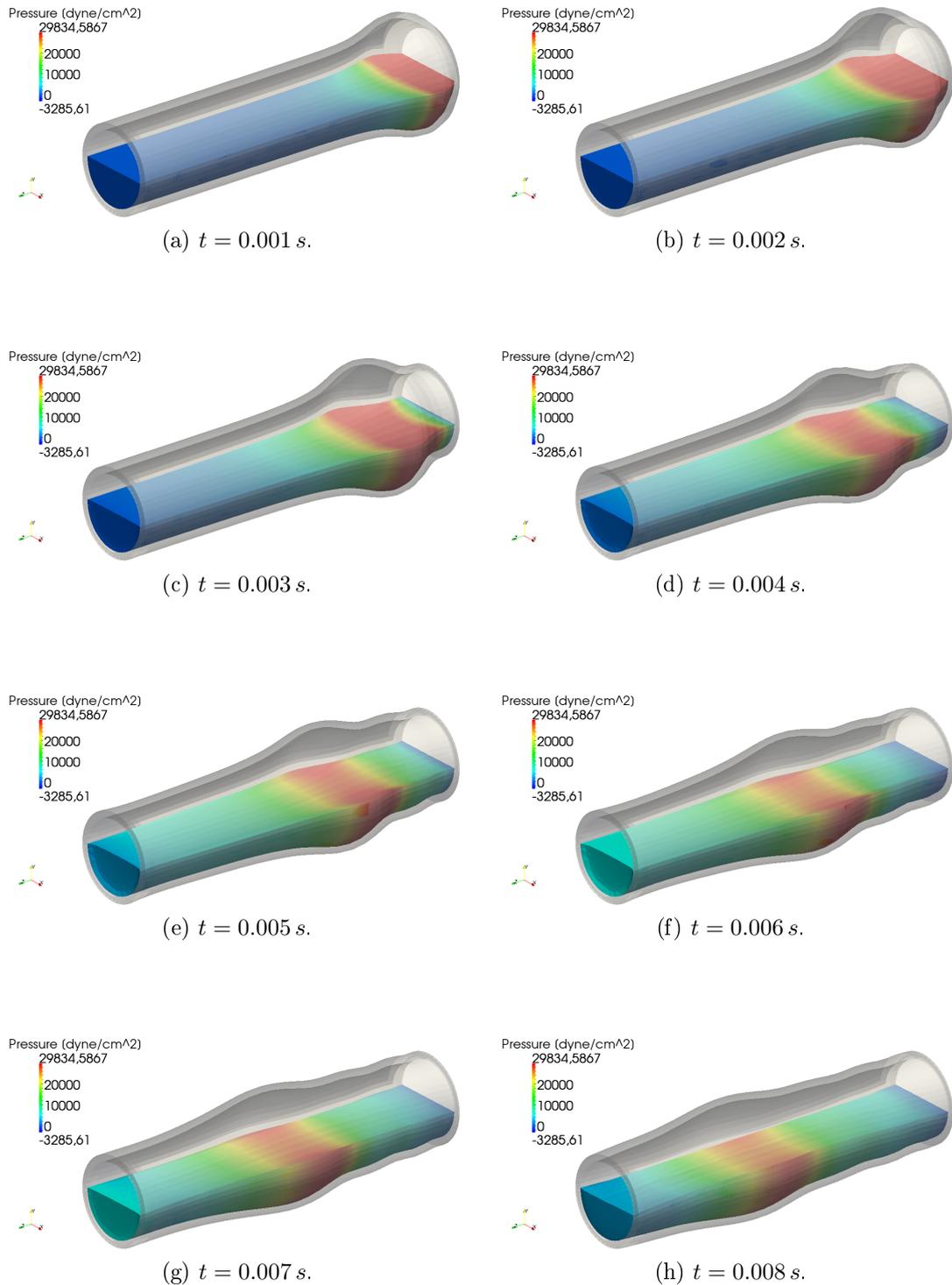
Figure 5.20: Representation of the fluid-pressure contour plot, inside the whole deformed structure, at different simulation time-steps, using as mesh motion strategy the IDW method.

(a) $t = 0.001\,s$.

(b) $t = 0.002\,s$.

(c) $t = 0.003\,s$.

(d) $t = 0.004\,s$.

(e) $t = 0.005\,s$.

(f) $t = 0.006\,s$.

(g) $t = 0.007\,s$.

(h) $t = 0.008\,s$.
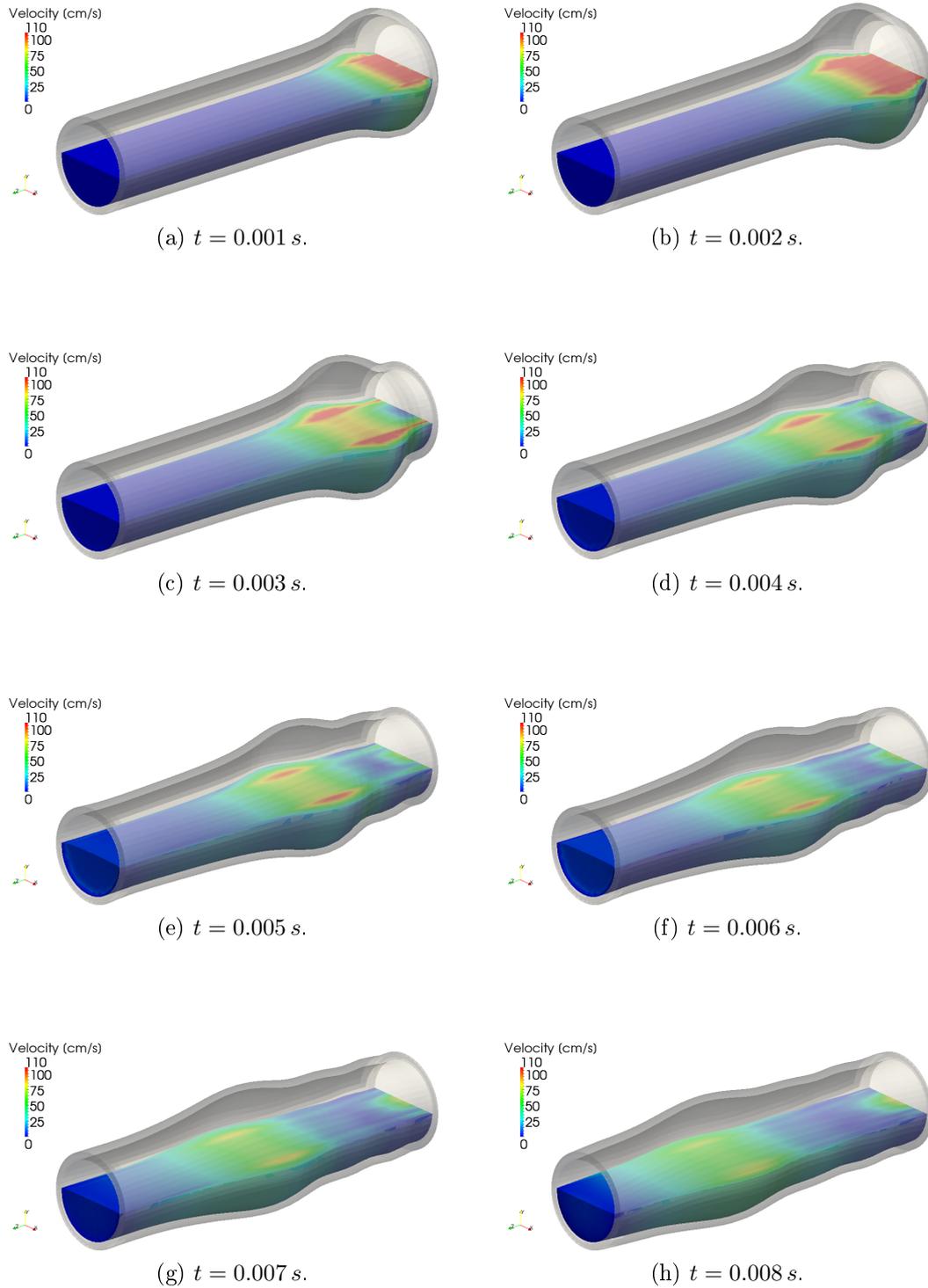
Figure 5.21: Representation of the fluid-velocity contour plot, inside the whole deformed structure, at different simulation time-steps, using as mesh motion strategy the IDW method.

(a) Edge ratio.

(b) Volume.

(c) Radius ratio.

(d) Minimum dihedral angle.

(e) Shape.

(f) Scaled jacobian.
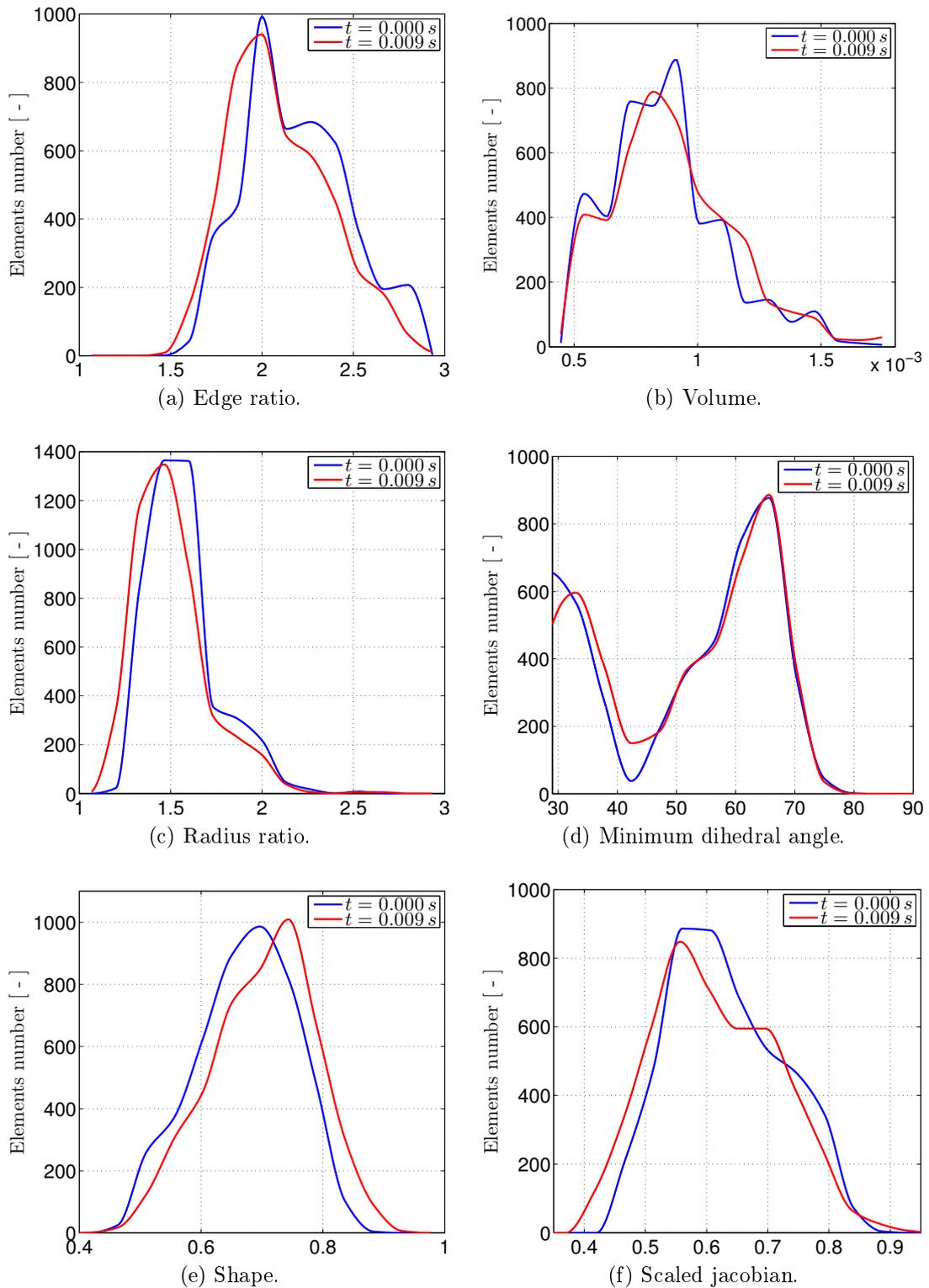
Figure 5.22: Measures of the fluid mesh quality obtained by the IDW mesh motion strategy.

(a) Edge ratio.

(b) Volume.

(c) Radius ratio.

(d) Minimum dihedral angle.

(e) Shape.

(f) Scaled jacobian.
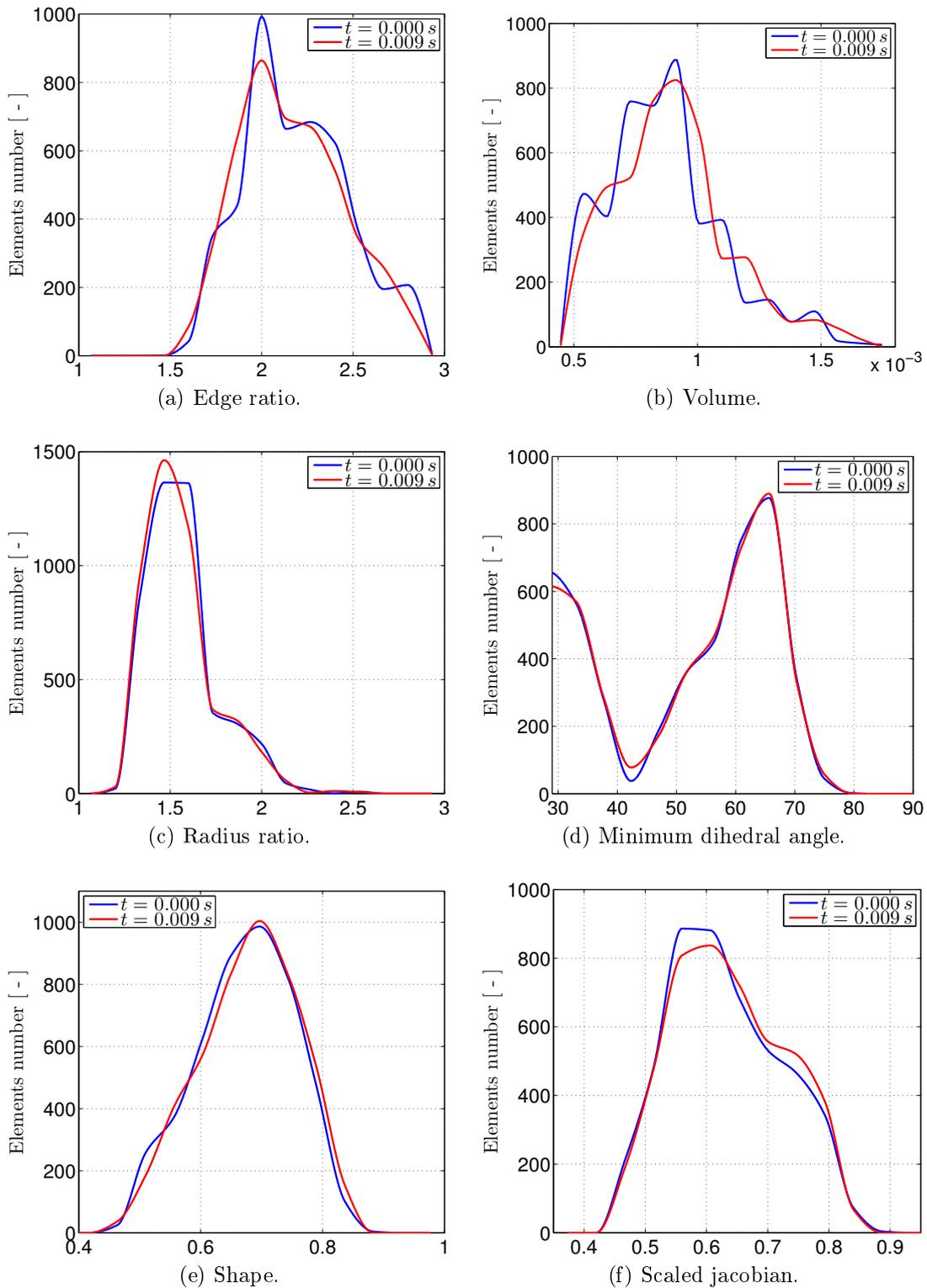
Figure 5.23: Measures of the fluid mesh quality obtained by the RBF mesh motion strategy.
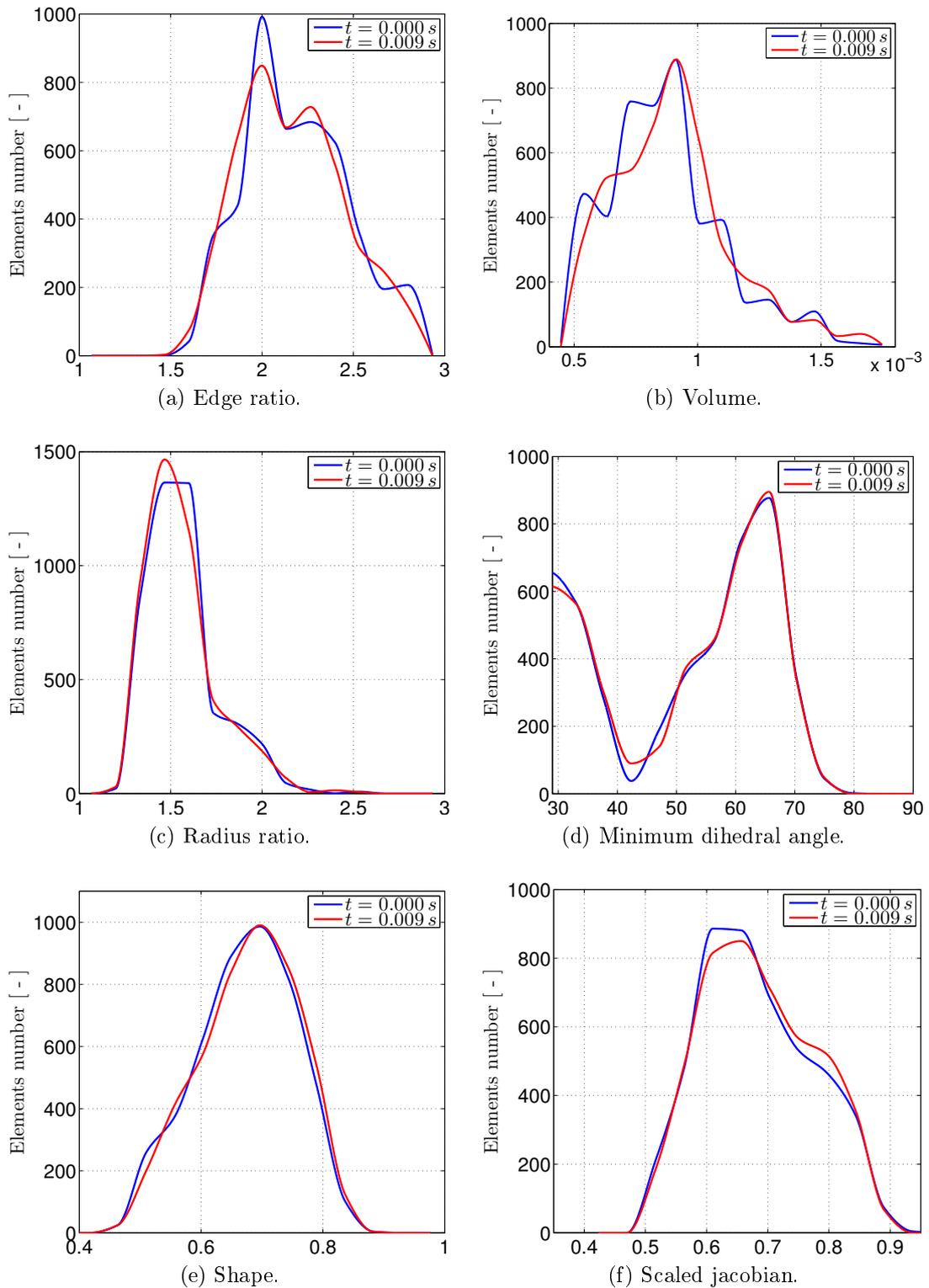
Figure 5.24: Measures of the fluid mesh quality obtained by the FFD mesh motion strategy.
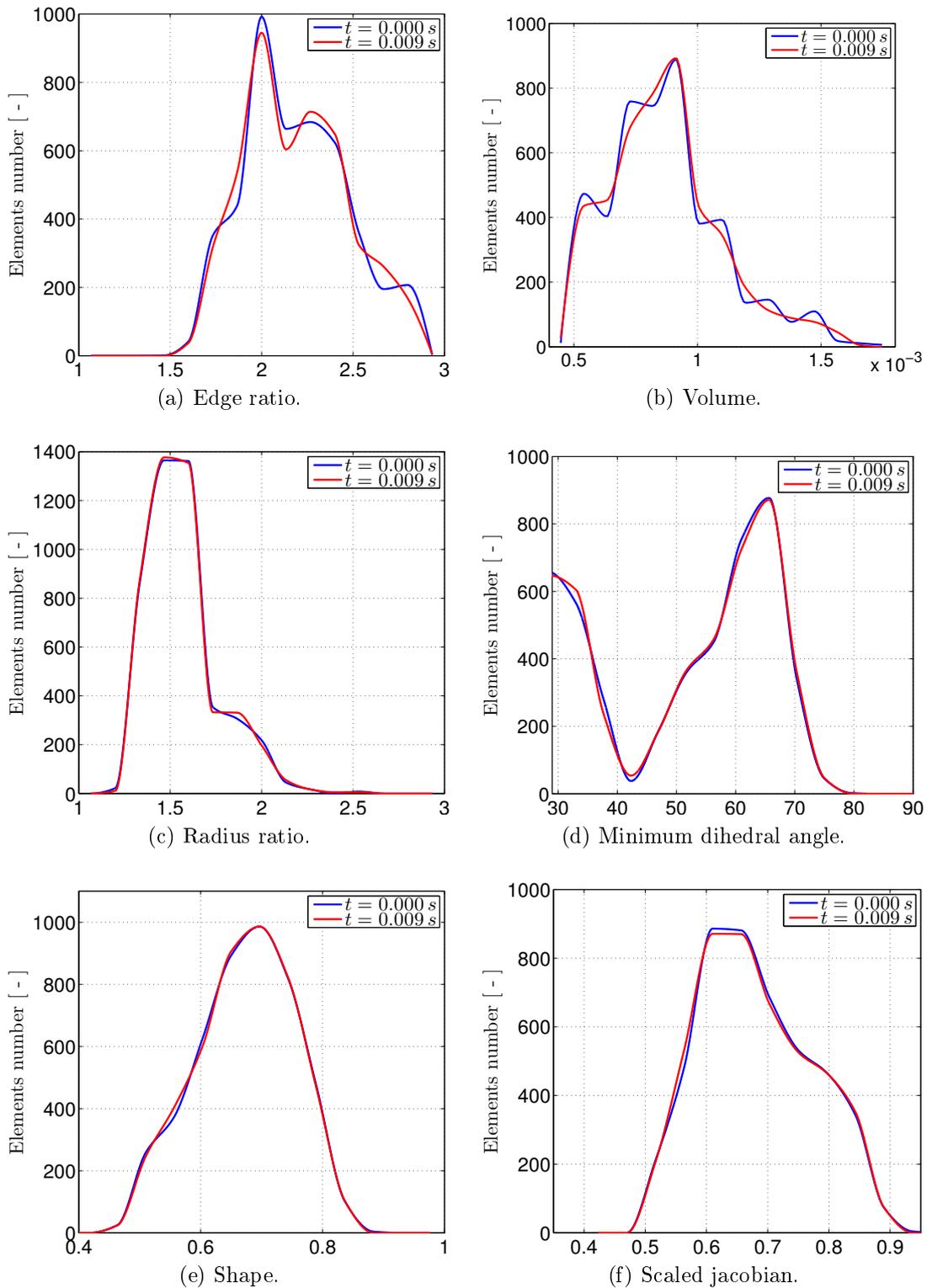
Figure 5.25: Measures of the fluid mesh quality obtained using the Laplacian mesh motion strategy.

By the analysis of the previous results, in terms of the deformed structural configuration, we can appreciate the high accuracy of the mesh motion strategies considered, since, among them, the global behavior is similar, and, moreover, the peak values of the structural displacement are quite the same. Concerning the mesh quality measures we can state that, during the FSI simulation, each quality indicator keeps values within its relative acceptable range indicated in Table 5.1. At this point, we decided to evaluate the computational costs related to the dynamic mesh handling by considering different discretizations for both the fluid and the structural domains. In detail we will evaluate, on the basis of the different meshes whose properties are indicated in Table 5.5, the mean time (across the different time iterations) needed to update the fluid mesh configuration due to the structural displacement.

	Domain	Vertices	Triangles	Thetrahedra	Total elements
Test #1	Fluid	437	496	1728	2224
	Structure	456	912	1296	2208
Test #2	Fluid	1029	952	4560	5512
	Structure	1840	1680	2400	4080
Test #3	Fluid	1869	1416	8880	10296
	Structure	1176	2352	3360	5712
Test #4	Fluid	5053	2968	25560	28528
	Structure	4960	5280	21600	26880
Test #5	Fluid	9452	4436	49698	54134
	Structure	7072	7488	30888	38376
Test #6	Fluid	13300	5476	71196	76672
	Structure	10500	9120	48960	58080

Table 5.5: Mesh properties of the many different discretizations adopted for both the fluid and the structural domains to evaluate the computational costs.

At the basis of these discretizations adopted, in Figure 5.26 we illustrate the curves of the computational costs versus the growing number of fluid elements. In particular, we report a comparison between IDW, RBF and Laplacian strategies on the left, while, on the right-hand-side we show the behavior of the FFD technique: concerning the latter, although it drives to a high accuracy of the deformed fluid mesh, from the costs point of view it is not competitive if compared to the others. As a matter of fact if one one hand we can compare on the same magnitude scale the times related to the IDW, RBF and Laplacian methods, the FFD one represents a too expensive strategy. Concerning the first three techniques (RBF, IDW and Laplacian), which are compared on the left-hand-side of Figure 5.26, we can state that when dealing with coarse grids, the costs are equivalent, but, as the fluid elements number increases, the IDW strategy represents the cheapest (better) solution.
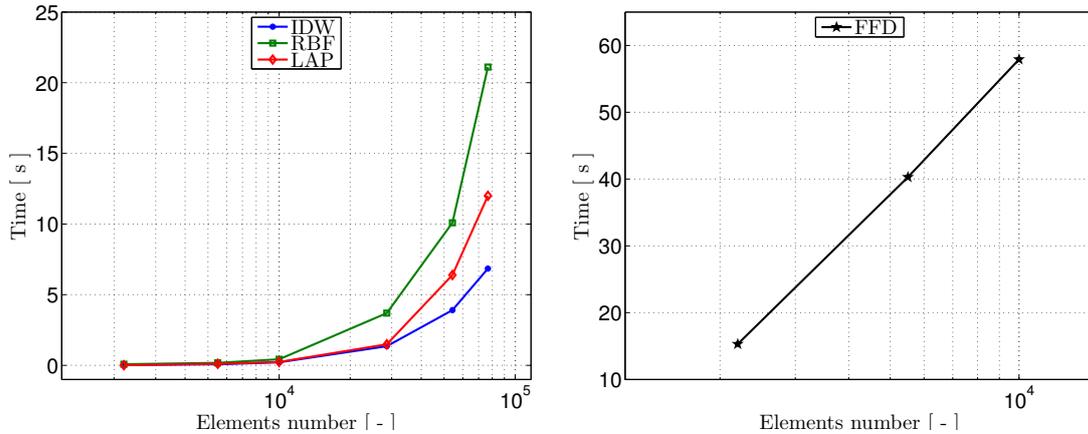
Figure 5.26: Analysis of the computational costs vs. the fluid elements number: on the left-hand-side we compare the mean time needed by the IDW, RBF and Laplacian strategies, while, on the right-hand-side, we report only for the first three meshes considered, the ones related to the FFD method.

We can conclude that the IDW strategy represents, within the dynamic mesh handling, a good compromise between the accuracy of the achieved results and the computational costs, driving also to a high quality of the deformed fluid grid as well. Concerning the other shape parametrization techniques, if on one hand they could represent a valid alternative to IDW method when dealing with coarse meshes, as the elements number increases, they would be less competitive. Nevertheless, in the framework of Reduced Order Modeling (ROM) and Reduced Basis (RB) methods [69, 80, 95, 101], shape parametrization strategies would play an important role, especially for those applications requiring real-time evaluations and/or repeated output evaluations for different values of some inputs of interest. In such situations [70, 71, 81, 82], the possibility given by FFD and RBF strategies to handle both the mesh motion issue in FSI problem, as well as the shape parametrization to control and modify the shape configuration, would represent a great advance in that scenario, that both the IDW and the Laplacian based methods do not provide at the moment.

Conclusions and future perspectives

In this work we have compared shape parametrization strategies, like Free-Form Deformations and Radial Basis Functions, as well as Inverse Distance Weighting techniques for the dynamic mesh handling for three-dimensional unsteady Fluid-Structure Interaction problems. Since shape parametrization strategies allow to represent shape deformations by means of low dimensional spaces (control points), we have tested their abilities within mesh motion problems for FSI in order to update the fluid mesh configuration due to structural displacements. Moreover, we have tested these techniques for the transfer of data across non-matching fluid-structure grids. The result of this work is the development of a mesh handling toolbox, able to deal with both mesh motion and fluid-structure interface problems for FSI; moreover such a toolbox can be easily linked with different state of the art FSI solvers (like LifeV, OpenFoam, etc.) thanks to the high flexibility of the code realized. In Chapter 1 we have detailed the motivations, as well as the objectives, that justify such a work. In Chapter 2 all the shape parametrization strategies considered, as well as the IDW techniques, are introduced. Furthermore, we have compared them on some one-,two- and three-dimensional examples. In detail it was possible to highlight the great potential of RBF technique in order to tackle both the fluid-structure interface and mesh motion problems, even if, concerning the latter, IDW performances (in terms of computational costs) were the best. For mesh motion problems, FFD strategies appeared too much expensive in terms of costs. As a result of these considerations, we proposed, in Chapter 3, some numerical procedures that lead to a quite significant reduction of costs with respect to both RBF and FFD techniques. In detail, concerning RBF, computational costs were reduced thanks to the adoption of a reduced set of parameters used to describe the shape deformation, while, with respect to FFD, such a goal was achieved by means of a so called "domain-decomposition approach". Through Chapter 4 we have introduced the mathematical model adopted for the FSI coupled problem: firstly we have recalled the equations describing both the fluid and structure physics, then we have introduced the space and time discretizations. Moreover, an overview of the Finite Element library (LifeV) adopted to numerically solve the FSI problem is proposed. Finally, in the last chapter of this work we tested the tool library developed by the numerical solution of many FSI test-cases (of both internal and

external fluid flows).

Concerning the dynamic mesh handling, among the mesh motion strategies considered, the Inverse Distance Weighting behaved as the best one, with respect to the achievable mesh qualities and to the computational costs. On the other hand, in order to tackle the fluid structure interface problem, Radial Basis Functions led to a great accuracy of the achieved results, even in those cases dealing with the transfer of patchy data across non-matching grids. At the basis of these considerations our hierarchical approach relies, gained as result of the many test proposed within this master thesis.

Concerning the future developments of this works, the testing of the mesh handling toolbox with respect to fluid-structure interface problems within FSI, could be seen as a last step in order to fully validate the work developed, since in this work we were mainly interested to face the dynamic mesh handling. Moreover, from an implementative point of view, although the code has been optimized to work in a serial way, we are supposed to re-organize it in a parallel framework, since we expect to improve significantly the performances, in particular with respect to IDW strategy. Finally, one of the greatest development of such a work relies in the application of shape parametrization strategies within Reduced Order Modeling: in detail by using FFD and RBF parametrizations not only to tackle the mesh motion and interface problems for FSI, but also to extent their application fields to such problems requiring significant reductions of both the *geometrical* and *computational* complexities.

Appendices

Appendix A: List of symbols

Appendix B: List of acronyms

Appendix A

List of symbols

Introduction:

Symbol	Description	Unit of measure
$\hat{\Omega}^F$	Reference fluid domain configuration	-
$\hat{\Omega}^S$	Reference structural domain configuration	-
Ω^F	Deformed fluid domain configuration	-
Ω^S	Deformed structural domain configuration	-
ζ	Fluid displacement field	$[cm]$
ς	Structural displacement field	$[cm]$
Γ^F	Deformed fluid interface configuration	-
Γ^S	Deformed structural interface configuration	-
ζ_L	local (deformative) component of ζ	$[cm]$
ζ_G	global (due to rigid motions) component of ζ	$[cm]$

Chapter 1:

Symbol	Description	Unit of measure
E	Young's modulus	$[dyne\,cm^{-1}]$
$N(\mathbf{x})$	Interpolation basis function	-
$w(\mathbf{x})$	Weighting function	-
$\delta(\cdot)$	Prefix used to indicate a virtual entity	-
ϕ	Radial Basis Function	-
γ	Diffusivity coefficient	-
Γ	Fluid-structure interface	-
μ, λ	Lamé coefficients	-

Symbol	Description	Unit of measure
ν	Poisson's ratio	-
$\mathbf{c}$	Center of the radial basis functions	$[cm]$
$\mathbf{f}$	Body forces	-
$\mathbf{f}_F$, $\mathbf{f}_S$	External forces acting on fluid and structural nodes	$[\,dyne\,]$
$\mathbf{H}$	Transfer matrix	-
$\mathbf{u}$	Displacement of grid nodes	$[cm]$
$\boldsymbol{\epsilon}$	Strain tensor	-
$\boldsymbol{\sigma}$	Stress tensor	-
$\boldsymbol{\varsigma}$, $\boldsymbol{\zeta}$	Structural and Fluid nodal displacements	$[\,cm\,]$
$\mathbb{R}e$	Reynolds number	-

Chapter 2:

Symbol	Description	Unit of measure
$\mathcal{N}$	Set of scattered data	-
$\mathcal{N}_c$	Number of RBF control points	-
$\mathcal{M}(\cdot, \boldsymbol{\mu})$	RBF interpolation map	-
$\mathcal{J}$	Cost functional	-
n_f	Number of fluid grid nodes	-
n_s	Number of structural grid nodes	-
n_Γ	Number of nodes lying over the fluid-structure interface	-
r	Scaling factor for RBF	$[\,cm\,]$
t^n	n-th time interation	$[\,s\,]$
N_C	Number of control points for FFD	-
R	Radius of influence for IDW	$[\,cm\,]$
$T(\cdot, \boldsymbol{\mu})$	Parametric map	-
$T_{FFD}(\cdot, \boldsymbol{\mu})$	FFD map	-
TOL	Tolerance	-
$\mathcal{D}$	Space of admissible parameters values for FFD	-
Γ^F, Γ^S	Linear selection operators	-
$\hat{\Omega}$	Reference domain	-
Ω	Current domain configuration	-
$\hat{\Omega}^F$	Reference fluid domain configuration	-
Ω^F	Current fluid domain configuration	-
$\hat{\Omega}^S$	Reference structure domain configuration	-

Symbol	Description	Unit of measure
Ω^S	Current structure domain configuration	-
$\hat{\Omega}_h$	Discretized domain in the reference configuration	-
Ω_h	Discretized domain in the current configuration	-
$\hat{\mathbf{d}}_S$	Structural displacement	[cm]
$\mathbf{P}^0_{l,m,n}$	Control points initial positions	[cm]
$\mathbf{X}_C$	RBF control points position	[cm]
$\boldsymbol{\mu}$	Parameters vector	-
$\boldsymbol{\Psi}$	Affine, differentiable and invertible map	-

Chapter 3:

Symbol	Description	Unit of measure
N_C	Number of control points	-
N_m	Number of eigenmodes considered	-
$\mathcal{J}^{(i_k)}$	Cost functional at the i_k-th iteration	-
$\mathcal{T}(\cdot)$	Frame of reference with respect to $(\cdot)$ coordinates	-
ς_R	Rigid component of the structural displacement	[cm]
ς_D	Deformative component of the structural displacement	[cm]
ω	Eigenvalue	$[Hz]$
$\hat{\Omega}^{(i)}$	i-th reference domain patch	-
$\mathbf{s}$	Rigid translations	$[cm]$
$\boldsymbol{\theta}$	Rigid rotations	$[deg]$
$\mathbf{I}$	Inertia matrix	-
$\mathbf{M}$	Mass matrix	-
$\mathbf{K}$	Stiffness matrix	-
$\mathbf{R}$	Rotation matrix	-
$\tilde{\mathbf{R}}$	Approximated rotation matrix	-
$\mathbf{X}_B$	Body coordinates	[cm]
$\boldsymbol{\theta}$	Rigid rotations	[rad]
$\boldsymbol{\Pi}$	Set of parameters	-

Chapter 4:

Symbol	Description	Unit of measure
p	Fluid pressure	$[\, dyne\, cm^{-2}\,]$
$\in$	Green-Lagrange strain tensor	-
Γ	Fluid-Structure interface	-
Ω_t	Current (Eulerian) domain configuration	-
ρ_F	Fluid density	$[\, g\, cm^{-3}\,]$
ρ_S	Structure density	$[\, g\, cm^{-3}\,]$
μ_F	Fluid dynamic viscosity	$[\, poise\,]$
$\mathcal{A}$	ALE map	-
$\hat{\mathbf{d}}_F$	Fluid domain displacement	$[\, cm\,]$
$\hat{\mathbf{d}}_S$	Fluid domain displacement	$[\, cm\,]$
$\dot{\hat{\mathbf{d}}}_S$	Structure velocity	$[\, cm\, s^{-1}\,]$
$\mathbf{F}$	Deformation gradient	-
$\mathbf{f}_F$	Fluid volume forces	$[\, dyne\, cm^{-3}\,]$
$\mathbf{f}_S$	Structure volume forces	$[\, dyne\, cm^{-3}\,]$
$\mathbf{n}$	Normal unit vector	-
$\mathbf{u}$	Fluid velocity	$[\, cm\, s^{-1}\,]$
$\mathbf{x}$	Point belonging to the current domain configuration	$[\, cm\,]$
$\hat{\mathbf{x}}$	Point belonging to the reference domain configuration	$[\, cm\,]$
$\mathbf{w}$	Mesh velocity	$[\, cm\, s^{-1}\,]$
$\boldsymbol{\sigma}_F$	Cauchy stress tensor for fluid	-
$\boldsymbol{\Sigma}_I$	First Piola stress tensor	-
$\boldsymbol{\Sigma}_{II}$	Second Piola stress tensor	-

Chapter 5:

Symbol	Description	Unit of measure
$\mathbf{P}_i$	Vertices of the tetrahedral element	-
$\mathbf{L}_i$	Edges of the tetrahedral element	-
V	Volume of the tetrahedral element	$[\, cm^3\,]$
A	Surface area of the tetrahedral element	$[\, cm^2\,]$
ρ	Edge ratio, mesh quality measure	-
δ	Distortion, mesh quality measure	-
r	Radius ratio, mesh quality measure	-
β	Minimum dihedral angle, mesh quality measure	$[\, deg\,]$

Symbol	Description	Unit of measure
σ	Shape, mesh quality measure	-
Δt	Simulation time-step	$[\ s\ ]$
$\mathbf{v_{F_{i}nflow}}$	Inflow velocity value	$[\ cm\ s^{-1}\ ]$

Appendix B

List of acronyms

Acronym	Meaning
ALE	Arbitrary Lagrangian Eulerian
B&W	Beckert and Wendland
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CMCS	Chair of Modelling and Scientific Computing
CPU	Central Processing Unit
FE	Finite Element
FEM	Finite Element Method
FFD	Free Form Deformation
FSI	Fluid Structure Interaction
GCE	Geometry Convective Explicit
GS	Gaussian Splines
IDW	Inverse Distance Weigthing
IMQ	Inverted Multi-Quadratic biharmonic splines
IP	Interface Problem
MMP	Moving Mesh Problem
MQ	Multi-Quadratic biharmonic splines
NNI	Nearest Neighbor Interpolation
NS	Navier Stokes
PCG	Preconditioned Conjugate Gradient
RB	Reduced Basis
RBF	Radial Basis Function
ROM	Reduced Order Modeling
SBRS	Solid Body Rotation Stress
TPS	Thin Plate Splines
WRM	Weighted Residual Methods

Bibliography

[1] Getpot input file and command line parsing. `http://getpot.sourceforge.net/`.

[2] Gmsh documentation. `http://geuz.org/gmsh/`.

[3] LifeV documentation. `http://www.lifev.org`.

[4] Octave documentation. `http://www.octave.org/`.

[5] OpenFOAM documentation. `http://www.openfoam.org/archive/2.0.0`.

[6] Paraview documentation. `http://www.paraview.org/`.

[7] E. Amoiralis and I. Nikolos. Freeform deformation versus B-spline representation in inverse airfoil design. *Journal of Computing and Information Science in Engineering*, 8:1–13, 2008.

[8] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999.

[9] M. Andreoli, A. Janka, and J. A. Desideri. Free-form-deformation parameterization for multilevel 3D shape optimization in aerodynamics. *INRIA Research Report no 5019*, 2003.

[10] S. Arya, D. M. Mount, N. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. In *Proceedings of the 5th ACM-SIAM Symposium on Discrete Algorithm*, pages 573–582, Arlington, VA, January 1994.

[11] S. Badia, F. Nobile, and C. Vergara. Fluid-structure partitioned procedures based on robin transmission conditions. *Journal of computational Physics*, 227:7027–7051, 2008.

[12] S. Badia, A. Quaini, and A. Quarteroni. Modular vs. non-modular precon-
ditioners for fluid-structure systems with large added-mass effect. *Com-
puter Methods in Applied Mechanics and Engineering*, 197(49-50):4216–
4232, 2008.

[13] F. Ballarin. *Ottimizzazione di forma per flussi viscosi tridimensionali in
geometrie cardiovascolari*. Master Thesis in Mathematical Engineering, Po-
litecnico di Milano, System Engineering School, December 2011.

[14] G. K. Batchelor. *An Introduction to Fluid Dynamics*. Cambridge Mathe-
matical Library series, Cambridge University Press, 1967.

[15] K. J. Bathe and G. Ledezma. Benchmark problems for incompressible fluid
flows with strucural interaction. *Computers and Structures*, 85:628–644,
2007.

[16] K. J. Bathe, H. Zhang, and M. Wang. Finite element analysis of incom-
pressible and compressible fluid flow with free surfaces and structural in-
teractions. *Computers and Structures*, 56:193–213, 1995.

[17] A. Beckert. Coupling fluid (CFD) and structural (FE) models using finite
element interpolation elements. *Aerospace Science Technology*, 4:13–22,
2000.

[18] A. Beckert and H. Wendland. Multivariate interpolation for fluid-structure-
interaction problems using radial basis functions. *Aerospace Science and
Technology*, 5:125–134, 2001.

[19] R. Boman and J. P. Ponthot. Finite element simulation of lubricated con-
tact in rolling using the arbitrary lagrangian-eulerian formulation. *Com-
putational Methods in Applied Sciences and Engineering*, 193:4323–4353,
2004.

[20] M. D. Buhmann. *Radial Basis Functions*, volume 12 of *Cambridge mono-
graphs on applied and computational mathematics*. Cambridge University
Press, UK, 2003.

[21] E. Burman and M. A. Fernández. Continuous interior penalty finite element
method for the time-dependent navier-stokes equations: space discretiza-
tion and convergence. *Numerische Mathematik*, 107(1):39–77, 2007.

[22] E. Burman and P. Hansbo. Edge stabilization for the generalized stokes
problem: A continuous interior penalty method. *Computer Methods in
Applied Mechanics and Engineering*, 195(19–22):2393–2410, 2006.

[23] G. T. Camacho and M. Ortiz. Adaptive lagrangian modelling of ballistic penetration of metallic targets. *Computer Methods in Applied Mechanics and Engineering*, 142(3-4):269–301, 1997.

[24] P. Causin, J. F. Gerbeau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Computer Methods in Applied Mechanics and Engineering*, 192(42–44):4506–4527, 2005.

[25] J. R. Cebral and R. Lohner. Conservative load projection and tracking for fluid-structure problems. *AIAA Journal*, 35:687–692, 1997.

[26] C. K. Chui and L. L. Schumaker. *Numerical solution of variational problems by radial basis functions*, volume 2 of *Approximation Theory IX*, pages 361–368. Vanderbilt University Press, Nashville, TN, 1998.

[27] S. Coquillart. Extended free-form deformation: a sculpturing tool for 3D geometric modeling. In *International Conference on Computer Graphics and Interactive Techniques: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, 1990.

[28] P. Crosetto. *Fluid-Structure Interaction Problems in Hemodynamics: Formulation, Solver, Preconditioners and Applications*. Phd thesis no 5109, École Polytechnique Fédérale de Lausanne, Lausanne, July 2011.

[29] P. Crosetto, S. Deparis, G. Fourestey, and A. Quarteroni. Parallel algorithms for fluid-structure interaction problems in haemodynamics. *SIAM Journal on Scientific Computing*, 33(4):1598–1622, 2011.

[30] A. de Boer. *Computational fluid-structure interaction: spatial coupling, coupling shell and mesh deformation*. Phd thesis, Technische Universiteit, Delft, December 2008.

[31] A. de Boer, M. S. van der Schoot, and H. Bijl. Mesh deformation based on radial basis function interpolation. *Computers and Structures*, 85:784 – 795, 2007.

[32] A. de Boer, A. H. van Zuijlen, and H. Bijl. Comparison of the conservative and consistent approaches for the coupling of non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 197:4284–4297, 2006.

[33] A. de Boer, A. H. van Zuijlen, and H. Bijl. Review of coupling methods for non-matching meshes. *Computer Methods in Applied Mechanics and Engineering*, 196:1515–1525, 2007.

[34] C. Degand and C. Farhat. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers and Structures*, 80:305–316, 2002.

[35] J. Degroote, K. J. Bathe, and J. Vierendeels. Performance of a new partitioned procedure versus a monolithic procedure in fluid-structure interaction. *Computers and Structures*, 89:793–801, 2009.

[36] S. Deparis. *Numerical analysis of axisymmetric flows and methods for fluid-structure interaction arising in blood flow simulation*. Phd thesis no 2965, École Polytechnique Fédérale de Lausanne, Lausanne, 2004.

[37] J. Duchon. Splines minimizing rotation invariant semi-norms in sobolev spaces. In *Constructive Theory of Functions of Several Variables*, pages 85–100. Springer, Berlin, 1977.

[38] T. Dunne. An eulerian approach to fluid-structure interaction and goal-oriented mesh adaptation. *International Journal for Numerical Methods in Fluids*, 51(9-10:1017–1039, 2006.

[39] R. P. Dwight. Robust mesh deformation using the linear elasticity equations. In H. Deconinck and E. Dick, editors, *Computational Fluid Dynamics 2006: Proceedings of the Fourth International Conference on Computational Fluid Dynamics*, pages 401–406, Gent, Belgium, July 2006. Springer.

[40] C. Farhat. CFD on moving grids: from theory to realistic flutter, maneuvering, and multidisciplinary optimization. *International Journal of Computational Fluid Dynamics*, 19(8):595–603, 2005.

[41] C. Farhat, C. Degand, B. Koobus, and M. Lesoinne. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163:231–245, 1998.

[42] C. Farhat and M. Lesoinne. Higher-order staggered and subiteration free algorithms for coupled dynamic aeroelasticity problems. In *Proceedings of the 36th AIAA Aerospace Sciences Meeting and Exibit*, Reno, NV, 1998.

[43] C. Farhat, M. Lesoinne, P. Stern, and S. Lantéri. High performance solution of three-dimensional nonlinear aeroelastic problems via parallel partitioned algorithms: Methodology and preliminary results. *Advances in Engineering Software*, 28:43–61, 1997.

[44] C. Farhat, M. Lesoinne, and P. Le Tallec. Load and motion transfer algorithms for fluid/structure interaction problems with non-matching discrete interfaces: Momentum and energy conservation, optimal discretization and application to aeroelasticity. *Computer Methods in Applied Mechanics and Engineering*, 157:95–114, 1998.

[45] C. Farhat, K. Pierson, and C. Degand. Multidisciplinary simulation of the maneuvering of an aircraft. *Engineering with Computers*, 17:16–27, 2001.

[46] C. Farhat, G. van der Zee, and P. Geuzaine. Second-order time-accurate loosely-coupled solution algorithms for nonlinear FSI problems. In *Proceedings of the 2004 European Congress on Computational Methods in Applied Sciences and Engineering*, Jyvaskyla, Finland, July 2004.

[47] M. Fernández and M. Moubachir. A newton method using exact jacobians for solving fluid-structure coupling. *Computers and Structures*, 83(2-3):127–142, 2005.

[48] B. Fonberg, T. A. Drinscoll, G. Wright, and R. Charles. Observations on the behavior of radial basis function approximations near boundaries. *Computers and Mathematics with Applications*, 43:473–490, 2002.

[49] L. Formaggia, J. F. Gerbeau, and C. Prud'homme. *LifeV Developer Manual*. The LifeV Project.

[50] Paul Jackson FRAeS, editor. *Jane's All the World's Aircraft*. Jane's Information Group, United Kingdom, 1996.

[51] C. Geuzaine and J. F. Remacle. Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79:1309–1331, 2009.

[52] G. P. Guruswamy. A review of numerical fluid/structure interface methods for computations using high-fidelity equations. *Computer and Structures*, 80:31–41, 2002.

[53] R. L. Harder and R. N. Desmarais. Interpolation using surface splines. *Journal of Aircraft*, 9:189–197, 1972.

[54] M. Heil. An efficient solver for the fully coupled solution of large displacement fluid structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 193(1):1–23, 2004.

[55] R. Heinrich, N. Kroll, J. Neumann, and B. Nagel. Fluid-structure coupling for aerodynamic analyses and design. In *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exibit*, Reno, NV, 2007.

[56] B. T. Helenbrook. Mesh deformation using the biharmonic operator. *International Journal for Numerical Methods in Engineering*, 56(7):1007–1021, 2003.

[57] A. Henderson. *The ParaView Guide: A Parallel Visualization Application*. Kitware Inc., Voorheesville, 2004.

[58] M. A. Heroux and J. M. Willenbring. Trilinos users guide. Technical Report SAND2003-2952, Sandia National Laboratories, 2003.

[59] M. A. Heroux, J. M. Willenbring, and R. Heaphy. Trilinos developers guide. Technical Report SAND2003-1898, Sandia National Laboratories, 2003.

[60] J. Hron and S. Turek. A monolithic FEM/multigrid solver for ALE formulation of fluid structure interaction with application in biomechanics. In *Fluid-Structure Interaction - Modelling, Simulation, Optimisation*, pages 146–170. Springer-Verlag, 2006.

[61] S. R. Idelsohn, F. Del Pin, R. Rossi, and E. Onate. Fluid-structure interaction problems with strong added-mass effect. *International Journal for Numerical Methods in Engineering*, 80:1261–1294, 2009.

[62] S. Jakobsson and O. Amoignon. Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computers and Fluids*, 36:1119–1136, 2007.

[63] S. Jakobsson, M. Patriksson, J. Rudholm, and A. Wojciechowski. A method for simulation based optimization using radial basis functions. *Optimization and Engineering*, 11:501–532, 2010.

[64] H. Jasak. Dynamic mesh handling in OpenFOAM. In *Proceedings of the 47th AIAA Aerospace Sciences Meeting*, Montreal, Canada, June 2009.

[65] X. Jiao and M. T. Heath. Common-refinement-based data transfer between non-matching meshes in multiphysics simulations. *International Journal for Numerical Methods in Engineering*, 61:2402–2427, 2004.

[66] H. Kanchi and A. Masud. A 3D adaptive mesh moving scheme. *International Journal for Numerical Methods in Fluids*, 54:923–944, 2007.

[67] A. Koshakji. *Free Form Deformation Techniques for 3D Shape Optimization Problems*. Master Thesis in Aeronautical Engineering, Politecnico di Milano, Industrial Engineering School, April 2011.

[68] H. L. Lamousin and W. N. Waggenspack. NURBS-based free-form deformations. *IEEE Computer Graphics and Applications*, 14(6):59–65, 1994.

[69] T. Lassila, A. Quarteroni, and G. Rozza. A reduced basis model with parametric coupling for fluid-structure interaction problems. *SIAM Journal on Scientific Computing*, 34:1187–1213, 2012.

[70] T. Lassila and G. Rozza. Model reduction of steady fluid-structure interaction problems with free-form deformations and reduced basis methods. In *Proceedings of the 10th Finnish Mechanics Days*, pages 454–465, Jyvaskyla, Finland, December 2009.

[71] T. Lassila and G. Rozza. Parametric free-form shape design with PDE models and reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24):1583–1592, 2010.

[72] R. Lohner, C. Yang, J. R. Cebral, J. D. Baum, H. Luo, D. Pelessone, and C. Charman. Fluid-structure interaction using a loose coupling algorithm and adaptive unstructured grids. In M. Hafez and K. Oshima, editors, *Computational Fluid Dynamics Review*, pages 755–776. Jhon Wiley and Sons, 1995.

[73] M. Lombardi, N. Parolini, and A. Quarteroni. Radial basis functions in FSI problems: interpolation and mesh motion. *International Journal for Numerical Methods in Fluids*, 2012. Submitted.

[74] M. Lombardi, N. Parolini, A. Quarteroni, and G. Rozza. Numerical simulation of sailing boats: dynamics, FSI, and shape optimization. In G. Buttazzo and A. Frediani, editors, *Variational Analysis and Aerospace Engineering: Mathematical Challenges for Aerospace Design*, Springer Optimization and Its Applications, chapter 15, pages 339–378. Springer, 2012.

[75] E. J. López, N. M. Nigro, M. A. Storti, and J. A. Toth. A minimal element distortion strategy for computational mesh dynamics. *International Journal for Numerical Methods in Engineering*, 69(9):1898–1929, 2007.

[76] S. Lukaszyk. A new concept of probability metric and its applications in approximation of scattered data sets. *Computational Mechanics*, 33:299–304, 2004.

[77] D. R. Lynch and K. O'Neill. *Elastic grid deformation for moving boundary problems in two space dimensions*, volume 3 of *Finite Elements in Water Resources*, chapter 7, pages 111–120. University of Mississippi, Oxford, 2003.

[78] A. Madrane, A. Raichle, and A. Sturmer. Parallel implementation of a dynamic overset unstructured grid approach. In *Proceedings of the ECCOMAS 2004 Conference*, Jyvaskyla, Finland, July 2004.

[79] N. Maman and C. Farhat. Matching fluid and structure meshes for aeroelastic computations: a parallel approach. *Computers and Structures*, 54:779–785, 1995.

[80] A. Manzoni, A. Quarteroni, and G. Rozza. Computational Reduction for Parametrized PDEs: Strategies and Applications. *Milan Journal of Mathematics*, 1:1–27, 2012.

[81] A. Manzoni, A. Quarteroni, and G. Rozza. Model reduction techniques for fast blood flow simulation in parametrized geometries. *International Journal for Numerical Methods in Biomedical Engineering*, 28:604–625, 2012.

[82] A. Manzoni, A. Quarteroni, and G. Rozza. Shape optimization for viscous flows by reduced basis methods and free-form deformation. *International Journal for Numerical Methods in Fluids*, 2012. Available online, in press.

[83] A. Masud. Effects of mesh motion on the stability and convergence of ALE based formulations for moving boundary flows. *Computational Mechanics*, 38(4):430–439, 2006.

[84] A. K. Michler. Aircraft control surface deflection using RBF-based mesh deformation. *International Journal for Numerical Methods in Engineering*, 88:986–1007, 2011.

[85] A. M. Morris, C. B. Allen, and T. C. S. Rendall. CFD-based optimization of aerfoils using radial basis functions for domain element parametrization and mesh deformation. *International Journal for Numerical Methods in Fluids*, 58:827–860, 2008.

[86] F. Nobile. *Numerical approximation of fluid-structure interaction problems with application to haemodynamics.* Phd thesis no 2458, École Polytechnique Fédérale de Lausanne, Lausanne, 2001.

[87] W. F. Noh. *A time-dependent two-space-dimensional coupled Eulerian-Lagrangian code*, volume 3 of *Methods in computational physics*. Academic Press, New York, 1964.

[88] K. C. Park, C. A. Filippa, and G. Rebel. A simple algorithm for localized construction of non-matching structural interfaces. *International Journal for Numerical Methods in Engineering*, 53:2117–2142, 2002.

[89] V. N. Parthasarathy, C. M. Graichen, and A. F. Hathaway. A comparison of tetrahedron quality measures. *Finite Elements in Analysis and Design*, 15(3):255–261, 1994. technical note.

[90] S. Pierret and R. A. van den Braembussche. Turbomachinery blade design using a navier-stokes solver and artificial neural network. *Journal of Turbomachinery*, 121(2):326–333, 1999.

[91] S. Piperno and C. Farhat. Design and analysis of staggered fluid-structure time integrators for interface momentum and energy conservation. In *Proceedings of the 4th U.S. National Congress on Computational Mechanics*, San Francisco, CA, August 1997.

[92] M. A. Potsdam and G. P. Guruswamy. A parallel multiblock mesh movement scheme for complex aeroelastic applications. In *Proceedings of the 39th AIAA Aerospace Sciences Meeting and Exhibit*, volume 716, pages 1–14, Reno, NV, January 2001.

[93] G. Quaranta, P. Masarati, and P. Mantegazza. A conservative mesh-free approach for fluid-structure interface problems. In *International Conference on Computational Methods for Coupled Problems in Science and Engineering*, Santorini, Greece, May 2005.

[94] A. Quarteroni. *Numerical Models for Differential Problems*. Springer-Verlag Italia, Milan, 2009.

[95] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximation for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1:3:1–44, 2011.

[96] A. Quarteroni, G. Rozza, and A. Manzoni. Certified reduced basis approximations for parametrized partial differential equations and applications. *Journal of Mathematics in Industry*, 1(3):1–44, 2011.

[97] T. C. S. Rendall and C. B. Allen. Efficient mesh motion using radial basis functions with data reduction algorithms. *Journal of Computational Physics*, 54:79–94, 2009.

[98] O. Reynolds. *Papers on mechanical and physical subjects-the sub-mechanics of the Universe*, volume 3 of *Collected Work*. Cambridge University Press, UK, 1903.

[99] G. Romanelli. *Computational Aeroservoelasticity of Free-Flying Deformable Aircraft*. Phd thesis, Politecnico di Milano, Milano, March 2012.

[100] G. Romanelli and E. Serioli. *Un approccio libero alla moderna Aeroelasticitá Computazionale*. Master Thesis in Aeronautical Engineering, Politecnico di Milano, Industrial Engineering School, July 2008.

[101] G. Rozza, D. B. P. Huynh, and A. T. Patera. Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. *Archives of Computational Methods in Engineering*, 15(3):1–47, 2007.

[102] M. Sala, M. A. Heroux, and D. M. Day. Trilinos tutorial. Technical Report SAND2004-2189, Sandia National Laboratories, 2004.

[103] J. A. Samareh. A survey of shape parametrization techniques. Technical Report NASA/CP-1999-209136, NASA, 1999.

[104] J. A. Samareh. Aerodynamic shape optimization based on free-form deformation. In *Proceedings of the 10th AIAA/ISSMO Multidiscipinary Analysis and Optimization Conference*, volume 6, pages 3672–3683, Albany, NY, 2004.

[105] D. T. Sandwell. Biharmonic spline interpolation of geos-3 and seasat altimeter data. *Geophysical research letters*, 14(2):139–142, 1987.

[106] S. S. Sarakinos, E. Amoiralis, and I. Nikolos. Exploring freeform deformation capabilities in aerodynamic shape parametrization. In *Proceedings of the International Conference on Computer as a tool*, volume 1, pages 535–538, Belgrade, Serbia and Montenegro, November 2005.

[107] R. Schaback and H. Wendland. Numerical techniques based on radial basis functions. In A. Cohen, C. Rabut, and L. L. Shumaker, editors, *Curve and Surface Fitting*, pages 359–374. Vanderbilt University Press, Saint Malo, 1999.

[108] A. Schutte, G. O. Einarsson, A. Raichle, B. Schoning, M. Orlt, J. Neumann, J. Arnold, W. Monnich, and T. Forkert. Numerical simulation of maneuvering aircraft by aerodynamic, flight mechanics and structural mechanics coupling. In *Proceedings of the 45th AIAA Aerospace Sciences Meeting and Exibit*, Reno, NV, January 2007.

[109] T. W. Sederberg and S. R. Parry. Free-form deformation of solid geometric models. In *Proceedings of SIGGRAPH - Special Interest Group on GRAPHics and Interactive Techniques*, pages 151–159, Dallas, TX, Agust 1986.

[110] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proceedings of the 23rd ACM National Conference*, pages 517–524, New York, NY, 1968.

[111] M. J. Smith, C. E. S. Cesnik, and D. H. Hodges. Evaluation of some data transfer algorithms for noncontiguous meshes. *Journal of Aerospace Engineering*, 13(2):52–58, 2000.

[112] M. J. Smith, D. H. Hodges, and C. E. S. Cesnik. Evaluation of computational algorithms suitable for fluid structure interaction. *Journal of Aerospace Engineering*, 37(2):282–294, 2000.

[113] E. L. Sutanto, J. D. Mason, and K. Warwick. Mean-tracking clustering algorithm for radial basis function centre selection. *International Journal of Control*, 67(6):961–977, 1997.

[114] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Computer Methods in Applied Mechanics and Engineering*, 190:3039–3067, 2001.

[115] P. Thévenaz, T. Blu, and M. Unser. Interpolation revisited. *IEEE Transactions on Medical Imaging*, 19(7):739–758, 2000.

[116] S. Turek and J. Hron. Proposal for numerical benchmarking of fluid-structure interaction between an elastic object and laminar incompressible flow. In *Fluid-Structure Interaction*, volume 53, pages 371–385. Springer, 2006.

[117] A. H. van Zuijlen, A. de Boer, and H. Bijl. Higher-order time integration through smooth mesh deformation for 3D fluid-structure interaction simulations. *Journal of Computational Physics*, 224:414 – 430, 2007.

[118] J. Vierendeels, K. Dumont, E. Dick, and P. Verdonck. Analysis and stabilization of fluid-structure interaction algorithm for rigid body motion. *AIAA Journal*, 43:2549–2557, 2005.

[119] K. Wang and C. Farhat. Algorithms for interface treatment and load computation in embedded boundary methods for fluid-structure interaction problems. In *Proceedings of the 2011 SIAM Conference on Computational Science and Engineering*, Nevada, NV, February 28-March 4 2011.

[120] K. Wang, A. Rallu, J. F. Gerbeau, and C. Farhat. Algorithms for interface treatment and load computation in embedded boundary methods for fluid and fluid-structure interaction problems. *International Journal for Numerical Methods in Fluids*, 67:1175–1206, 2011.

[121] H. Wendland. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. *Advances in Computational Mathematics*, 4:573–582, 1994.

[122] H. Wendland. Local polynomial reproduction and moving least square approximation. *Journal of Numerical analysis*, 21(1):285–300, 2001.

[123] J. A. S. Witteveen and H. Bijl. Explicit mesh deformation using inverse distance weighting interpolation. In *Proceedings of the 19th AIAA Computational Fluid Dynamics Conference*, San Antonio, TX, June 2009.

[124] Z. Wu. Multivariate compactly supported positive definite radial functions. *Advances in Computational Mathematics*, 4:283–292, 1995.

[125] D. Yang. *C++ and Object Oriented Numeric Computing for Scientists and Engineers*. Springer-Verlag, New York, NY, 2001.

[126] Q. Zhang and T. Hisada. Studies of the strong coupling and weak coupling methods in fsi analysis. *International Journal for Numerical Methods in Engineering*, 60:2013–2029, 2004.

[127] Y. Zhang and C. Bajaj. Adaptive and quality quadrilateral/hexahedralmeshing from volumetric data. *Computer Methods in Applied Mechanics and Engineering*, 195(9-12):942–960, 2006.

Acknowledgements

A conclusione di questo lavoro di tesi desidero ringraziare tutti coloro che mi sono stati vicino e mi hanno aiutato nel corso di questi anni di studio. Desidero ringraziare il Prof. Alfio Quarteroni, per la possibilità datami di sviluppare il presente lavoro di tesi presso l'EPFL, in particolar modo per la sua accoglienza e grande attenzione mostrata nei miei confronti, nonchè per aver pensato al mio futuro. Un grande ringraziamento inoltre al Dr. Ing. Gianluigi Rozza, al quale devo molto soprattutto per la fiducia mostrata nei miei confronti nell'affrontare, per questo lavoro, tematiche così interessanti ed innovative. Lo ringrazio inoltre per la sua grande gentilezza, per gli incoraggiamenti e l'attenzione avuta miei confronti. Rivolgo inoltre sentiti ringraziamenti al Prof. Luca Formaggia, sia per la grande passione trasmessami nei confronti del calcolo numerico e dell'implementazione in ambito scientifico, sia per l'avermi per primo proposto di svolgere la tesi presso l'EPFL. Particolari ringraziamenti, inoltre, li rivolgo al Dr. Simone Deparis per i preziosi suggerimenti riguardanti gli aspetti algoritmici per la soluzione del problema FSI.

Ringrazio per la loro grande disponibilità tutti i dottorandi e gli assistenti della cattedra del gruppo CMCS dell'EPFL, ed in particolare: Paolo e Claudia per i numerosi aiuti riguardanti l'aspetto implementativo con LifeV, Toni ed Andrea per l'aver condiviso ed approfondito le tematiche introdotte nel Capitolo 3 di questo lavoro, Aymen per le lunghissime chiacchierate e risate fatte in ufficio, Matteo e Laura per i continui incoraggiamenti. Un sentitissimo ringraziamento a Matteo per la sincera disponibilità avuta nei miei confronti nell'ospitarmi più volte a Losanna.

Il più grande ringraziamento va alla mia famiglia, in particolar modo ai miei genitori: per il loro sostegno ed il loro amore soprattutto nei momenti di difficoltà. I loro grandi sacrifici, la loro benevolenza e dedizione verso il lavoro hanno da sempre rappresentato per me un grande esempio da seguire. Ringrazio inoltre la mia carissima nonna Graziosa, che a suo modo mi è stata e sempre mi sarà vicino, per l'avermi sempre accolto con uno Splendido sorriso e per le sue infinite preghiere. Ringrazio la mia dolce e speciale Aurora: in questi anni mi sei sempre stata vicina, mi hai sempre sostenuto ed incoraggiato aiutandomi sia a superare

i momenti difficili sia nel prendere decisioni importanti. Ringrazio inoltre la sua famiglia, per la loro sempre calda accoglienza, disponibilità ed interesse nei miei confronti. Infine ringrazio i miei speciali "Colleghi" per l'aver condiviso tutto questo percorso di studi in un ambiente sempre sereno e caratterizzato da un sincero spirito di collaborazione.