Fluid-Structure Interaction Problems in Hemodynamics: Parallel Solvers, Preconditioners, and Applications

THÈSE N^{O} 5109 (2011)

PRÉSENTÉE LE 25 JUILLET 2011

À LA FACULTÉ SCIENCES DE BASE
CHAIRE DE MODÉLISATION ET CALCUL SCIENTIFIQUE
PROGRAMME DOCTORAL EN MATHÉMATIQUES

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Paolo CROSETTO

acceptée sur proposition du jury:

Prof. K. Hess Bellwald, présidente du jury
Prof. A. Quarteroni, Dr S. Deparis, directeurs de thèse
Dr M. Picasso, rapporteur
Dr C. Vergara, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2011

Riassunto

I principali obiettivi di questo lavoro sono la descrizione, lo studio e la simulazione numerica del problema di interazione fluido–struttura (FSI) applicato all'emodinamica (dinamica del sangue) nelle arterie. Lo studio numerico dell'emodinamica nel sistema cardiovascolare é un soggetto di ricerca molto attivo, in quanto permetterebbe, una volta validato, di predire l'insorgere di patologie, di scegliere la terapia piu' opportuna o di comprendere meglio l'influenza di fattori (come lo sforzo di taglio a parete o wall shear stress, WSS) notoriamente legati alla nascita di disturbi (come l'ateroscerosi).

Questo lavoro é suddiviso in tre parti, formate da due capitoli ciascuna. Nella prima parte vengono ricavate le equazioni differenziali che costituiscono il problema accoppiato: le equazioni di Navier–Stokes in un dominio mobile per il fluido (sangue), viscoso e incomprimibile, l'equazione dell'elasticitá per la struttura (parete arteriosa). In particolare viene descritta in dettaglio la rappresentazione delle equazioni del fluido in un sistema di riferimento Lagrangiano–Euleriano arbitrario (ALE), una scelta frequente in ambito FSI che verrá adottata nei capitoli successivi. In seguito vengono introdotte le condizioni di accoppiamento, ovvero le continuitá della velocitá e degli sforzi all'interfaccia tra fluido e struttura (la superficie endoteliale della parete arteriosa). Nella prima parte vengono discusse anche la discretizzazione spaziale in elementi finiti (FE) e temporale del sistema FSI. Questa discretizzazione permette di rappresentare il sistema di equazioni in uno spazio di dimensione finita, dando luogo ad un sistema discreto la cui soluzione é unica. La scelta della discretizzazione temporale coinvolge due aspetti: la discretizzazione in tempo dei due problemi (fluido e struttura) e quella delle condizioni di accoppiamento (continuitá delle velocitá e degli sforzi). Per entrambi gli aspetti la scelta puó influire sulla stabilitá del sistema discreto. Alla fine della prima parte di questa tesi viene riportata una descrizione dello stato dell'arte, focalizzata suilla stabilitá del sistema.

Il sistema di equazioni discretizzato é nonlineare. In particolare la dipendenza del domimio fluido dallo spostamento della struttura e la formulazione ALE scelta per il fluido introducono una forte nonlinearitá, il cui trattamento costituisce uno dei due principali argomenti della seconda parte della tesi. In letteratura la nonlinearitá del sistema FSI viene frequentemente risolta per mezzo del metodo di punto fisso, che ha come vantaggi robustezza ed implementazione relativamente immediata, ma anche, nella sua implementazione classica, lo svantaggio di essere inefficiente in alcune circostanze (tipicamente per esempio nel caso dell'emodinamica). Un algoritmo piú efficace, che viene utilizzato in questo lavoro, é quello di Newton. La maggiore difficoltá di questo approccio consiste nel calcolo della matrice Jacobiana, che richiede la valutazione delle derivate dei termini nonlineari. In particolare la nonlinearitá dovuta alla dipendenza del dominio fluido dallo spostamento della struttura fa intervenire delle derivate di forma nello Jacobiano del sistema. Il calcolo analitico e l'assemblaggio di queste derivate nella matrice Jacobiana non sono banali (questi termini sono spesso trascurati o approssimati

numericamente in letteratura), e vengono descritti nel terzo capitolo. Considerazioni orientate all'implementazione di questa parte in un codice ad elementi finiti sono riportate alla fine del capitolo e costituiscono un contributo originale di questo lavoro.

La seconda parte prosegue, nel quarto capitolo, con lo studio dei metodi di risoluzione del sistema lineare (Jacobiano). Un metodo efficace solitamente adottato per questo tipo di sistemi (grandi, sparsi e non simmetrici) é GMRES, un metodo di Krylov *matrix–free*, che richiede soltanto delle moltiplicazioni matrice–vettore per la soluzione del sistema. Solitamente questo metodo é utilizzabile in applicazioni pratiche soltanto se precondizionato. Dopo una rassegna sui metodi comunemente usati in FSI per risolvere il sistema linearizzato viene proposto un nuovo tipo di precondizionatori, che permette di trattare separatamente i blocchi corrispondenti ai diversi problemi (fluido e struttura). Un'analisi di questo tipo di precondizionatori, proposta alla fine della seconda parte, mostra che il condizionamento del sistema globale precondizionato dipende in gran parte dal condizionamento dei singoli problemi disaccoppiati.

Nella terza parte i metodi descritti nei capitoli precedenti vengono applicati a dei casi clinici. Sono simulati diversi battiti cardiaci in un arco aortico ed in un bypass femorale. Viene calcolato il WSS, vengono messi a confronto diversi metodi (modelli 1D, con parete rigida ed FSI) e diverse discretizzazioni spaziali e temporali. Infine, nell'ultimo capitolo, sono presentati risultati di scalabilitá forte e debole su diverse griglie, utilizzando diverse discretizzazioni temporali. Le simulazioni di questo capitolo sono state lanciate su piattaforme parallele ad alta performance (Cray XT5, Cray XT6, Blue Gene/P).

Résumé

Les objectifs de ce travail sont la description, l'étude et la simulation numérique du problème d'interaction fluide–structure (FSI) appliqué à la dynamique du sang (hémodynamique) dans les artères. L'étude numérique du système cardiovasculaire d'un point de vue hémodynamique est un sujet de recherche très actif, permettant, une fois validé, de prédire le développement de pathologies (par example l'athérosclérose), de mieux comprendre l'influence de facteurs (comme le wall shear stress, WSS) qu'y sont associés et de l'appliquer à la pratique clinique.

Ce travail est divisé en trois parties, chacune formée de deux chapitres. Dans la première partie les équations différentielles qui constituent le problème couplé sont introduites : les équations de Navier–Stokes dans un domaine déformable pour le fluide (sang), visqueux et incompressible, l'équation de l'élasticité pour la structure (paroi artérielle). En particulier on décrit en détail la représentation des équations du fluide dans un repère arbitraire Lagrangien-Eulerien (ALE), un choix fréquent en FSI qui sera adopté durant cette thèse. Ensuite on décrit les conditions de couplage : continuité des vitesses et des contraintes à l'interface fluide–structure. On introduit également dans la première partie les discrétisations spatiale, en éléments finis, et temporelle du système FSI. Ces discrétisations permettent de représenter le système d'équations dans un espace de dimension finie, ce qui mène à un problème discret dont la solution est unique. Le choix de la discrétisation temporelle influence deux aspects : la discrétisation en temps des deux problèmes (fluide et structure) et celle des conditions de couplage (continuité des vitesses et des contraintes). Pour les deux aspects le choix peut affecter la stabilité du système discret. Une description concernant en particulier la stabilité du système se trouve à la fin de la première partie de ce travail.

Le système d'équations discrétisé n'est pas linéaire, le fait que le domaine du fluide dépende du déplacement de la structure, ainsi que la formulation ALE pour le fluide, introduisent une forte nonlinéarité dont le traitement est un des deux arguments principaux de la deuxième partie de cette thèse. Il est souvent proposé dans la littérature de résoudre la nonlinéarité du système FSI en appliquant la méthode du point fixe. Elle a pour avantages d'être robuste et d'avoir une implémentation assez simple. Par contre, dans sa forme classique, cette méthode présente le désavantage de ne pas être efficace dans tous les cas, notamment l'hémodynamique. Un algorithme plus performant, qui est utilisé dans ce cas, est celui de Newton. La difficulté principale de cette méthode vient du calcul de la matrice Jacobienne, qui requiert l'évaluation des dérivées des termes nonlinéaires. En particulier, la nonlinéarité due à la dépendance du domaine fluide du déplacement de la la structure fait intervenir des dérivées de forme dans le Jacobien du système. Le calcul analytique et l'assemblage de ces dérivées ne sont pas triviaux (ces termes sont souvent négligés ou approximés dans la littérature), et sont décrits dans le troisième chapitre. A la fin de ce chapitre on décrit l'implémentation de cette partie dans un code aux éléments finis, ce qui constitue une contribution originale de ce travail.

Dans le quatrième chapitre on étudie des méthodes de résolution du système linéaire (Ja-

cobien). Une méthode efficace normalement utilisée pour ce type de systèmes (grands, creux et nonlinéaires) est GMRES. Cette méthode est utilisée normalement dans les cas pratiques avec un préconditionneur. Après avoir résumé des méthodes utilisées en FSI pour résoudre le système linéairisé, un nouveau type de préconditionneurs est proposé. Ces preconditionnerus permettent de traiter séparément les blocs qui correspondent aux problèmes différents (comme fluide et structure). Une analyse proposée pour ce type de préconditionneurs montre que le conditionnement du système global préconditionné ne dépend que du conditionnement des problèmes découplés.

Dans la troisième partie, les méthodes décrites dans les chapitres précédents sont appliquées à des cas cliniques. Plusieurs battements cardiaques consécutifs sont simulé dans le cas de l'aorte thoracique d'un sujet sain, ainsi que dans le cas d'un pontage fémoro–poplité. Le WSS est calculé et différentes méthodes sont comparées (modèle 1D, paroi rigide, FSI), ainsi que différentes discrétisations spatiales et temporelles. Enfin dans le dernier chapitre, les résultats de scalabilité forte et faible sont montrés, sur des maillages différents, avec différentes méthodes. Les simulations de ce chapitre ont été réalisées sur des clusters à haute performance (Cray XT5, Cray XT6, Blue Gene/P).

Mots clé : Interaction fluide-structure, préconditionneurs parallèles, hémodynamique, éléments finis

Abstract

In this work we aim at the description, study and numerical investigation of the fluid–structure interaction (FSI) problem applied to hemodynamics. The FSI model considered consists of the Navier–Stokes equations on moving domains modeling blood as a viscous incompressible fluid and the elasticity equation modeling the arterial wall. The fluid equations are derived in an arbitrary Lagrangian–Eulerian (ALE) frame of reference. Several existing formulations and discretizations are discussed, providing a state of the art on the subject. The main new contributions and advancements consist of:

- A description of the Newton method for FSI–ALE, with details on the implementation of the shape derivatives block assembling, considerations about parallel performance, the analytic derivation of the derivative terms for different formulations (conservative or not) and for different types of boundary conditions.

- The implementation and analysis of a new category of preconditioners for FSI (applicable also to more general coupled problems). The framework set up is general and extensible. The proposed preconditioners allow, in particular, a separate treatment of each field, using a different preconditioning strategy in each case. An estimate for the condition number of the preconditioned system is proposed, showing how preconditioners of this type depend on the coupling, and explaining the good performance they exhibit when increasing the number of processors.

- The improvement of the free (distributed under LGPL licence) parallel finite elements library LifeV. Most of the methods described have been implemented within this library during the period of this PhD and all the numerical tests reported were run using this framework.

- The simulation of clinical cases with patient–specific data and geometry, the comparison on simulations of physiological interest between different models (rigid, FSI, 1D), discretizations and methods to solve the nonlinear system.

A methodology to obtain patient–specific FSI simulations starting from the raw medical data and using a set of free software tools is described. This pipeline from imaging to simulation can help medical doctors in diagnosis and decision making, and in understanding the implication of indicators such as the wall shear stress in the pathogenesis.

Keywords: Fluid-structure interaction, parallel preconditioners, hemodynamic, finite elements

A nonna Rita

Acknowledgments

During my PhD I was fascinated by the new parallel architectures which keep improving in terms of speed, size and energy consumption. My conclusion, at the end of three years ($\approx$) is that the most performant parallel machine is here at the EPFL. It's my supervisor Prof. Alfio Quarteroni: no matter how much the workload increases, he manages to be up to date with everybody's work, to send suggestions, corrections, so I guess he meets the definition of weak scalability. I would like to thank him also for his capability of understanding how to deal with each one of his students, in order to get the best possible outcome.

Secondly, I want to thank my other supervisor, Dr. Simone Deparis, who looked after me for all this period, forgiving me all my forgetfulness. Besides the outstanding professional skills, he is a person with a deep humanity and a magic capability of "breaking the ice" with a laugh and taking it easy.

A huge thanks goes to my family: my mother Irene, father Alberto, sister Elena and my grandma Margherita for the provisions of good food, wine, clothes, and for being available for any problem and supportive in any circumstances.

I want to thank the CMCS crew, I am sure that I could find plenty of qualities for all of them, from both the professional and human points of view. I just acknowledge explicitly Jean for correcting all the accents in the rêsümè, Claudia, who had the "luck" of reading the first part of this thesis at an early stage (and gave me several corrections), and Radu, who is already a celebrity in Moretta, for a list of moments ranging from Sat's Karmeliet to the historical B-fest. I did not forget the people who left the crew, in particular Annalisa, with her smiles and complaints, Lausanne is less colorful ever since she went to Texas.

Among the the "swiss" people I want to thank Elena, Ale and Ojas, historical friends, it seems to me that I known them since forever. Among the "italians" I'd like to thank Davide and Alexandra for having left for me a spot on their couch in Torino, every Friday when I was back from Lausanne, Andrea (prugna), probably the most honest person on the face of earth, and the B-team. Thanks also to Chiara and Cristina, that I just crossed during this period, but I have wonderful souvenirs of them.

Thanks to the owners of the arteries simulated during this thesis: Dr. Philippe Reymond, whose passion for research always motivates me, and who contributed to many results contained in the simulations section, and Monsieur Martin. Thanks also to Gilles, who was generous of precious contributions and suggestions. I acknowledge the Swiss National Science Foundation grant 200020-117587 for the support, CSCS, HECToR, CADMOS and DIT for the availability of the supercomputers, the European grant Mathcard, the Swiss Platform for High-Performance and High-Productivity Computing (HP2C).

Eventually I would like to thank my coffee machine Alicia, and the "head scratcher" (those who spent some time in my office know how much support they gave me).

Contents

I Physics of the Problem **7**

Derivation of the Equations for Fluid and Structure **9**
1.1 The Kinematics of Continuous Media 9
1.2 Lagrangian, Eulerian and ALE Formulations 13
1.3 The Equations of Continuum Mechanics 16
1.4 The Equations for a Fluid 19
1.5 The Equations for a Solid 21

Modeling Fluid–Structure Interaction Problems **25**
2.1 Coupling Conditions 25
2.2 Three Fields Formulation 26
2.3 Time Discretization for the Structure Problem 28
2.4 Time Discretization for the Fluid–Geometry Problem 29
2.5 Fully Implicit and Convective Explicit Schemes 32
2.6 Geometry–Convective Explicit Scheme 32
2.7 Space Discretization 33
2.7.1 Interior Penalty Stabilization 38
2.8 Analysis of the Coupled Problem 38
2.8.1 Analysis of Simplified Coupled Models 38
2.8.2 Stability and Geometric Conservation Law 40
2.8.3 Strong versus Weak Coupling 42
Added Mass Effect 42
Fractional–Step and Explicit Couplings 45
2.9 Geometrical Multiscale 46

II Solution Strategies for the FSI problem **49**

Nonlinearities and Newton Method **51**
3.1 Newton Method 51
3.2 Fixed Point Formulations 52
3.2.1 Dirichlet–Neumann Formulation 54
3.2.2 Steklov–Poincaré Formulation 56
3.3 Newton Method for Geometrical Multiscale 60
3.4 Computing the Exact Jacobian Matrix 61
3.4.1 Shape Derivatives of Domain Functionals 62

3.4.2 Shape derivatives in FSI-ALE . . . 65
3.4.3 Implementation . . . 69

Solution of the Algebraic Linear System **71**
4.1 Block Triangular Preconditioners . . . 71
4.1.1 Schur Complement Preconditioners . . . 72
4.1.2 Block Gauss–Seidel Preconditioners . . . 74
4.2 Applications to the FSI system . . . 74
4.2.1 Robin–Robin Preconditioners . . . 77
4.2.2 Dirichlet–Dirichlet Preconditioners . . . 79
4.2.3 Other Additive Preconditioners . . . 81
4.2.4 Extension to Other Time Discretizations . . . 82
4.3 Parallel Preconditioners for FSI . . . 84
4.3.1 State of the Art . . . 85
4.3.2 Composed Preconditioners for Geometry–Convective Explicit FSI . . . 87
4.3.3 Composed Preconditioners for Geometry Implicit FSI . . . 88
4.3.4 Spectral Analysis . . . 89

III Applications and Simulations **95**

Applications to Hemodynamics **97**
5.1 Applications and Motivation . . . 97
5.1.1 Circulation . . . 97
5.1.2 Cardiovascular Diseases . . . 99
5.2 Problem Description . . . 101
5.2.1 From DICOM Images to Numerical Simulations . . . 102
5.2.2 Unsteady Blood Flow in a Compliant Iliac Artery . . . 104
5.3 Quality Mesh Generation for Cardiovascular Flow Simulations . . . 105
5.3.1 Surface Remeshing Techniques . . . 106
Harmonic Mapping . . . 107
Conformal Mapping . . . 107
5.3.2 High Quality Meshes . . . 107
5.3.3 Unsteady Blood Flow in a Compliant Femoropopliteal Bypass . . . 110
5.4 Blood Flow in the Aorta . . . 112
5.4.1 Boundary Conditions . . . 114
5.4.2 Geometrical Model . . . 117
5.4.3 Timings and Validation for FSI (GCE) . . . 118
5.4.4 Numerical Assessment . . . 125
5.4.5 Wall Shear Stress . . . 125
5.5 Comparisons . . . 129
5.5.1 GCE versus FI . . . 129
5.5.2 Exact versus Inexact Newton Method . . . 147
5.5.3 FSI (GCE) versus Rigid Walls and 1D . . . 149

Scalability and Parallel Performances **151**
6.1 Introduction . . . 151
6.2 Scalability and Results . . . 151
6.2.1 Strong Scalability Tests . . . 152
Strong Scalability for GCE . . . 153
Strong Scalability for CE . . . 154
Physiological geometries . . . 157
6.2.2 Weak Scalability Test . . . 158
Weak Scalability for GCE . . . 159
Weak scalability for CE . . . 162
Conclusion . . . 162

Introduction

The modeling of the cardiovascular system is receiving increasing attention from both the medical and mathematical environments because of, from the one hand, the great influence of hemodynamics on cardiovascular diseases and, from the other hand, its challenging complexity that keeps open the debate about the setting up of appropriate models and algorithms. A wide variety of approaches can be found in literature, dealing with different formulations of the problem and solution strategies.

Below we give an overview of some of the most popular methodologies to solve numerically the system of equations arising from the hemodynamic model. We briefly introduce the coupled Fluid–Structure Interaction (FSI) problem, listing many of the different approaches commonly adopted to tackle it. We also report an outline of the thesis, which is concerned with the development of algorithms for the solution of the FSI problem and their application to blood flow in situations of clinical relevance.

The equations considered in the present work consist of those describing the flow field variables (blood velocity and pressure) and those that govern the mechanical deformation of the "structure" (the vessel walls). The first distinction between the different methodologies comes from the formulation of the problem.

A common choice in the FSI context is to describe the fluid equations using an Arbitrary Lagrangian-Eulerian (ALE) frame of reference (see e.g. [Nob01, SH07], cf. Chapter 1). The advantage with respect to an Eulerian description is that the coupling can be satisfied exactly on the fluid-structure interface. However the introduction of a new equation for the fluid domain motion is required, and its dependence on the solution of the FSI problem introduces a further nonlinearity.

A different approach consists of using a space-time formulation within an Eulerian framework. Usually, the latter involves a discretization of the computational domain in *time slabs*, and each solution in a time slab is computed sequentially (see [TSS06, HWD04], or [BCHZ08] for a description of this formulation).

Other approaches are based on a standard Eulerian formulation [CMM08, WCLB08, MPGW10]. With the latter approach the computation of the fluid domain displacement is avoided, however a method to keep track of the fluid-structure interface must be employed.

Also the Lagrangian meshless finite elements methods [IOP03, OnO11] have been coupled with structure equations in order to model FSI problems. The dynamic of the fluid is modeled in a Lagrangian frame of reference, which has the advantage that the convective term disappears and the disadvantage that at each time step the domain discretization needs to be recomputed.

The lattice Boltzmann method, quite popular in computational fluid dynamics, has recently been used also for FSI. The coupling with a finite elements method for the struc-

ture mechanics is investigated e.g. in [Kwo08], while the immersed boundary method is used in [CZ10] to identify the fluid-structure interface.

Once the system of equations describing the physical problem is set up, a further optional step consists of splitting the global system into *subdomain problems*, one domain being that of the fluid, the other that of the solid, within standard domain decomposition (DD) approaches. In this context, Dirichlet–Neumann schemes are the most popular ones adopted in FSI (see e.g. [BQQ08a, KW08b, DDFQ07, MNS06, FM05]). Robin-Neumann and Robin-Robin schemes are applied in [BNV08, GGNV10] to the FSI context, while other standard domain decomposition strategies (e.g. Neumann-Neumann, FETI) are described for a general problem e.g. in [TW05]. Another similar option consists of reformulating the problem on the fluid-structure interface through the Steklov–Poincaré operators, see e.g. [DDQ06, DDFQ06].

All these strategies correspond to a particular choice of the subdomains and of the interface conditions assigned in the course of the subdomain iterations. Following the definitions given in [CK02] all these formulations of the problem can be qualified as *nonlinear preconditioners* (see Section §3.2). These domain decomposition schemes are particularly suited to the case when separate (and independent) solvers for the subdomain problems are available, because the solution of the global system can be obtained through repeated solutions of the subdomain problems (this property is often referred to as *modularity*).

The choice of the time discretization introduces further distinctions among the methods. The fully coupled nonlinear problem can be discretized in time by considering all the terms in the equations implicitly, which leads to a *Fully Implicit* (FI) method [BCHZ08, TSS06, HHB08, KGF$^+$09, BC10a, DP07]. This is the most stable but also most expensive choice. A large variety of alternative time discretizations can be devised. For instance a *Geometry-Convective Explicit* (GCE) discretization is proposed in [BQQ08a], where the moving geometry is taken at the previous time step and the convective term is treated partly explicitly (see Section §2.7 for details). Even in the space-time framework the fluid domain in a time slab can be extrapolated using the informations relative to previous time slabs, e.g. [TSS06, HWD04]. In this thesis (cf. Chapter 5) we compare the FI and GCE methods, together with two other intermediate options, obtained by varying the time discretization of the convective term.

A natural way to handle the nonlinearity is based on the use of the Aitken accelerated fixed point algorithm in all its variants, see e.g. [KW08b, BQQ08a, DDFQ06]. In this way each fixed point iteration requires one residual evaluation.

Otherwise the time discretized problem can be linearized via the Newton method, either considering the full Jacobian matrix, as in [BCHZ08, HHB08, FM05, TSS06, GKW10], or neglecting some of its contributions, as in [BC10a, GV03, DBV09, Hei04, Dep04]. In the Newton method the full Jacobian matrix is often available only as matrix-vector multiplication (this is the case in [FM05] and in [GKW10]). In these cases a matrix–free method must be employed to solve exactly the Jacobian system. Each iteration of this method requires a solve of the linearized subproblems. Thus the cost of each nonlinear iteration corresponds to the cost of one residual evaluation plus a variable number of solutions of the linearized subproblems. A detailed explanation of this kind of algorithms is provided in Chapter 3.

A further distinction between the different methods comes from the way the coupling conditions are advanced in time. We can devise three main different coupling strategies: strong coupling, weak coupling, fractional step schemes, see Section §2.8.3. This choice has an impact on both the stability and the order of accuracy of the overall scheme.

For what concerns the fully coupled discretized equations where no domain decomposi-

Time Discr.	System Formulation		Solution Algorithm		Preconditioner
FI/CE	Newton		GMRES/direct		P_{AS}, P_{GS}, $P_{AS}(P_{GS})$ ...
	inexact Newton		GMRES/direct		P_{AS}, P_{GS}, $P_{AS}(P_{GS})$ ...
	DD	Newton	out. GMRES/Rich.	inn. GMRES/dir.	P_{sub}
	DD	inexact N.	out. GMRES/Rich.	inn. GMRES/dir.	P_{sub}
	DD	Fixed Point		inn. GMRES/dir.	P_{sub}
GCE	Linear System		GMRES/direct		P_{AS}, P_{GS}, $P_{AS}(P_{GS})$ ...
			out. GMRES/Rich.	inn. GMRES/dir.	P_{sub}

Table 1: Methodologies for the solution of fluid-structure interaction problems ("Rich" stands for Richardson, "dir" for direct, "out" and "inn" for outer, respectively inner, iterations).

tion is employed, the key aspect that characterizes the different methodologies is the choice of the preconditioner. In fact by choosing block preconditioners such as block Jacobi or block Gauss–Seidel, the preconditioned system can be solved in a modular fashion. These strategies represent the algebraic version of the domain decomposition algorithms cited above. Approximating the Schur complements in a block LU factorization is a strategy proposed in [BQQ08b] for a fractional step scheme; this method corresponds to an algebraic splitting of the FSI linear system.

A similar strategy adopted in [PS09] in a different context uses the approximation of the block LU factorization as a preconditioner for GMRES within a strongly coupled scheme, a choice supported by the analysis of the condition number of the preconditioned system carried out in [Axe94].

We remark that in literature for the sake of classification the terms *monolithic* and *partitioned* are used to split the different approaches in two categories. However, as we pointed out in [CDFQ11], since we did not find an unambiguous definition we prefer to refrain from using these notations.

A picture representing some of the methodologies listed above is given in Table 1. Here the symbol P_{AS} is used to denote the algebraic additive Schwarz preconditioner, while P_{GS} indicates a generic block Gauss–Seidel preconditioner. The preconditioner P_{sub} refers to the linear system on the subdomains (or sub-blocks in the algebraic case).

All the methodologies represented in Table 1 are implemented in the C++ parallel finite elements library LifeV and many of them were implemented as part of my PhD project.

This thesis is divided in three parts. In the first part the equations at the basis of the FSI problem are derived, an overview of some of the possible models for fluid and structure are discussed, the FSI problem is introduced with particular attention to some of the different coupling strategies proposed in literature.

The fluid problem in a moving domain is described using an ALE frame. This introduces further nonlinearities and stability conditions (the so called *discrete geometric conservation law*). On the other hand the coupling of the fluid with an elastic structure can also generate instabilities in some cases (mainly due to the importance of the added mass effect, when the mass density of the fluid is close to the one of the structure).

The second part contains the main original contributions of the present work in terms of solvers and preconditioners for the FSI problem. The Newton algorithm for FSI is described, a unified notation is introduced to present the various algorithms studied in literature and in particular those implemented in the finite element library LifeV. The same notation is then used also to represent the Newton algorithm applied to the geometrical multiscale framework

(i.e., the coupling of 3D FSI models with 1D models for arteries). Special attention is devoted to the derivation of the full Jacobian matrix of the whole FSI problem. This Jacobian includes a block containing cross derivatives of the fluid equations with respect to the domain motion. These (shape) derivatives have a non-trivial expression whose derivation is usually omitted for the sake of simplicity. We report here in Section §3.4 all the calculations leading to the expression of these derivatives together with implementation-oriented observations, which enhance the scalability and efficiency of the current approach.

The preconditioning techniques for FSI are discussed and an overview of the classical preconditioning strategies used in this field is provided. A preconditioning strategy is introduced which consists in a combination of block Gauss–Seidel and domain decomposition preconditioners. This technique allows to split the preconditioner in as many factors as the number of sub problems considered and for this reason it is well suited for multiphysics problems. The preconditioner is obtained as the product of the preconditioners built for each factor. Each one of these factors may then, in their turn, exploit a preconditioning strategy tuned for the specific problem, and if a factor does not change during the whole simulation the corresponding preconditioner can be reused, saving computational time. An estimate for the condition number of the preconditioned system is derived, showing that it depends on the quality of the preconditioners for each single factor in the aforementioned factorization and on the maximum singular value of a specific matrix, whose form depends on one of the coupling blocks (cf. Section §4.3.4). We remark that this maximum singular value plays a role similar to the CBS constant (see [KM09, AK10, Axe94], cf. Chapter 4), but it is not tied to symmetric positive definite matrices.

The third part of this thesis contains the numerical simulations as well as a comparison in terms of computational efficiency and accuracy of the results obtained using different solvers and preconditioners here proposed. The application domain is the hemodynamic in large arteries. Our simulations concern the blood dynamics in a compliant aortic arch under physiological conditions and in a femoropopliteal bypass, equipped with boundary conditions taken from clinical measurements. The results show that the values of relevant hemodynamic factors such as the wall shear stress (WSS), an important indicator for several diseases, depend quite substantially from the model and the discretization used. In particular in the case of the aorta a comparison between FSI, rigid walls and 1D model is carried out, showing significant differences in the WSS magnitude. A considerable difference, especially with a "large" timestep, is observed also between different time discretizations of the FSI problem (in particular between the GCE and FI time discretizations), this because of the large displacements in the aortic arch (more than 20% of the diameter) which induce a large nonlinearity. We also report WSS comparisons on different space discretizations (using meshes with different characteristic sizes) for the simulation of blood flow in a femoropopliteal bypass. The results obtained allow us to conclude that, if the mesh is not fine enough and no boundary-layer meshes are used, the WSS is severely underestimated.

A comparison on the aorta simulation shows that in some cases taking into account the shape derivatives block in the Jacobian greatly improves the efficiency of the algorithm during systole with respect to an inexact Newton method where shape derivatives are neglected. Furthermore we show that, using a special preconditioner, P_{AS-DN}, obtained by means of a triple factorization and by computing an additive Schwarz preconditioner for each factor, and an efficient implementation of the shape derivatives assembly, the nonlinear iterations performed using exact or inexact Newton methods have almost the same computational cost

(neither the Jacobian assembling nor the GMRES iterations are influenced by the shape derivatives computation).

The weak and strong scalability of the algorithms are tested using different preconditioners and different time discretizations on benchmark and physiological geometries. The preconditioning techniques that are used for comparison are a classical algebraic additive Schwarz preconditioner, computed on the matrix of the whole coupled system, and the composition of block Gauss Seidel preconditioning strategies, which lead to a representation of the preconditioners as a product of several factors, with the same algebraic additive Schwarz preconditioner. The preconditioners obtained by composition are more efficient than the classical algebraic additive Schwarz both in terms of number of iterations and computational time. The computation of the preconditioners introduced shows to be scalable, while there is room for improvement concerning the solution of the linear system. However, thanks to the condition number estimate derived in Section §4.3.4, the improvements can be achieved by choosing more suitable parallel preconditioning strategies for the different factors in the preconditioners (e.g. multilevel preconditioners like algebraic multigrid and multilevel domain decomposition preconditioners [KM09, TW05], or specific to each single field, like the pressure correction preconditioners for the fluid field [ESW05]).

We end this introduction by listing the parallel supercomputers used to run the algorithms proposed in this thesis.

- (**Callisto**) The Callisto cluster at EPFL, composed of blades with two 4-cores processors Intel Harptown (3.0 Ghz) each. The blades are interconnected through InfiniBand.
- (**Cray XT4**) The Cray XT4 supercomputer in the UK National Supercomputing Service HECToR[1], composed by blades containing 4 quad-core (AMD 2.3 GHz) nodes each. The nodes are connected in a 3D torus topology with Seastar communication chips on each node running Portals communication protocol.
- (**Cray XT5**) The Cray XT5 supercomputer Rosa of the Swiss National Supercomputing Center[2]. A Cray XT5 node is composed by two AMD 2.4GHz "Istambul" Opteron processors with six cores each. The nodes are connected in a 3D torus topology with Seastar+ communication chips on each node.
- (**Cray XT6**) The Cray XT6 supercomputer in the UK National Supercomputing Service HECToR. Each node on the Cray XT6 supercomputer is composed by two 12-cores 64-bit AMD Opteron "Magny–Corus" (2.5 GHz) processors. The nodes are connected in a 3D torus topology with Seastar2+ communication chips on each node.
- (**Blue Gene/P**) The IBM Blue Gene/P supercomputer of the center for advanced modeling science (CADMOS). Each node in a Blue Gene/P is composed of compute chips which integrate four IBM PowerPC 450 32-bit processor cores. A dual-pipeline floating-point unit is attached to each core. The nodes are connected with a 3D torus topology with Serdes communication chips, see [IBM08].

[1] http://www.hector.ac.uk
[2] http://www.cscs.ch

Part I

Physics of the Problem

Derivation of the Equations for Fluid and Structure

In this chapter we derive the equations describing the fluid and solid problems, we introduce the formalism which is adopted throughout this work, and we give a brief overview of some of the possible approaches to model fluid and structure in the context of hemodynamics.

The structure of this chapter is divided into two main parts. The first (represented by Sections §1.1 §1.2 and §1.3) recalls some basis of continuous mechanics, while in the second (Sections §1.4 and §1.5) the models describing the fluid and the solid dynamics are respectively derived.

1.1 The Kinematics of Continuous Media

We recall in this part some standard mathematical concepts which are used to describe the continuous media. We refer mainly to Fernández, Formaggia, Gerbeau, Quarteroni [FQV09, Ch.3] and Scovazzi, Hughes [SH07] for what concerns the modeling part and the derivation of the basic equations.

Let us define a bounded open reference domain $\widehat{\Omega} \subset \mathbb{R}^3$, which represents the body in its original undeformed configuration, with boundary $\widehat{\gamma}$. Most of the quantities defined in this chapter in the reference configuration can be ported to a deformed one (which will be introduced later) by means of a change of variables. Some quantities are represented only on either one or the other configuration. When there is ambiguity we distinguish between the two different representations by adding a *hat* ˆ if the quantity is represented in the reference domain. In order to derive some important relations which are at the basis of continuum mechanics we introduce some concepts from differential geometry, which will be useful also in Section §3.4. We refrain though from reporting all the definitions necessary for a rigorous derivation, since the latter would be beyond the scope of the current discussion. We refer to e.g. [Fla89] for an introduction to differential forms and exterior algebra. This detour may be skipped for the moment for those readers who are not interested in the derivations of all the formulas.

We introduce a set of local coordinates of the domain following [SZ92] (see Figure 1.1). We suppose that there are m overlapping sets $O_i \subset \mathbb{R}^3$ that cover $\widehat{\Omega}$. Given

$$B = \{\xi = (\xi_1, \xi_2, \xi_3) \in \mathbb{R}^3 : \|\xi\| \leq 1\},$$

and

$$B_o = \{\xi = (\xi_1, \xi_2, \xi_3) \in B : \xi_3 = 0\},$$

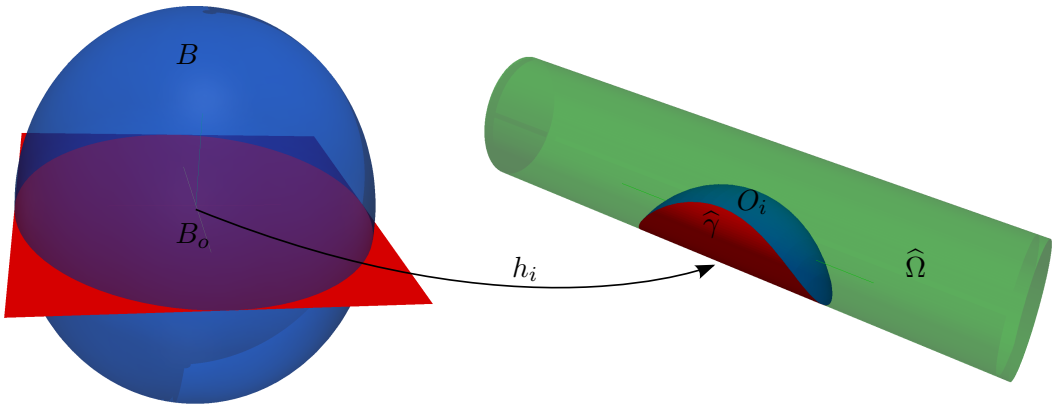


Figure 1.1: Sketch of a mapping defining local coordinates on $\widehat{\Omega}$.

we define m one-to-one functions $c_i : O_i \to B \subset \mathbb{R}^3$ of class C^k and invertible, with inverse of class C^k, such that

$$\begin{aligned} c_i(O_i \cap \widehat{\Omega}) &= \{\xi \in B : \xi_3 \geq 0\} \\ c_i(O_i \cap \widehat{\gamma}) &= \{\xi \in B_o\}. \end{aligned} \tag{1.1.1}$$

Then we define the *coordinate functions* $h_i : B \to O_i$ such that $h_i(c_i(x)) = x$ for $x \in O_i \cap \widehat{\gamma}$. A point x in the overlap $O_i \cap O_j \cap \widehat{\gamma}$ can be represented using both coordinate functions h_i and h_j.

Suppose to fix $x \in O_i \cap \widehat{\gamma}$. The Jacobian of the coordinate functions is a 3×3 matrix which will be noted (omitting the partition index i) $\nabla_\xi h$. We can define the *metric tensor* $\mathbf{G} = (\nabla_\xi h)^T \nabla_\xi h$, which is a 3×3 matrix.

$\{\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3\}$ is the canonical orthonormal basis in $\mathbb{R}^3$. Considering the usual scalar product in $\mathbb{R}^3$ the tangent space on the point $x = h(\xi)$ on $\widehat{\gamma}$ is spanned by the vectors $\mathbf{t}_i = (\nabla_\xi h)\mathbf{e}_i$ for $i = 1, 2$. The vector $\mathbf{t}_3 = (\nabla_\xi h)\mathbf{e}_3$ is not necessarily orthogonal to $\mathbf{t}_1$ and $\mathbf{t}_2$. The normal vector can be defined as $\mathbf{t}_n = (\nabla_\xi h^T)^{-1}\mathbf{e}_3$. If we call h_j, for $j = 1, 2, 3$, the new set of local coordinates, we can define the vector $dx = \mathbf{t}_1 dh_1 + \mathbf{t}_2 dh_2 + \mathbf{t}_3 dh_3$. Then the metric tensor satisfies $dx^T \mathbf{G} dx = d\xi^T d\xi$.

The relation between the differentials in the two different coordinate systems (according to the definition of *contravariant vector* [Fla89, Ch.5.4]) reads

$$\sum_i (\nabla_\xi h)^i_j d\xi_i = dh_j. \tag{1.1.2}$$

Lemma 1.1.1. *The volume measure in $\widehat{\Omega}$ is given by the determinant of $\nabla_\xi h$,*

$$\int_{\widehat{\Omega}} dh = \int_B \det(\nabla_\xi h) d\xi. \tag{1.1.3}$$

The proof follows from the relation between the differentials (1.1.2) and the definition of determinant ([Fla89, Ch.2.2]).

The norm of the vector $\mathbf{t}_n$ in general is different from one. Let us call

$$\mathbf{n} = \frac{\mathbf{t}_n}{\|\mathbf{t}_n\|}, \tag{1.1.4}$$

the normalized vector orthogonal to the plane spanned by $\mathbf{t}_1$ and $\mathbf{t}_2$. Then a surface element can be represented by $S_h = \|\mathbf{t}_1 \times \mathbf{t}_2\| = \mathbf{n} \cdot (\mathbf{t}_1 \times \mathbf{t}_2)$. The change of measure on the boundary $\widehat{\gamma}$ is retrieved in the following proposition.

Proposition 1.1.2 (Nanson's formula)**.** *With the previous notations, the measure on the manifold $\widehat{\gamma}$ is represented by*

$$\|\mathrm{cof}\,(\nabla_\xi h)\mathbf{e}_3\| = \|\det(\nabla_\xi h)(\nabla_\xi h^T)^{-1}\mathbf{e}_3\|, \tag{1.1.5}$$

where $\mathrm{cof}\,(\nabla_\xi h) = |\nabla_\xi h|(\nabla_\xi h^T)^{-1}$ *is the* cofactor matrix *of* $\nabla_\xi h$*. This implies that*

$$\int_{\widehat{\gamma}} dh = \int_{B_o} \|\mathrm{cof}\,(\nabla_\xi h)\mathbf{e}_3\| d\xi.$$

Proof. We derive this formula for the unit surface element $S_\xi = \|(\mathbf{e}_1 \times \mathbf{e}_2)\| = (\mathbf{e}_1 \times \mathbf{e}_2) \cdot \mathbf{e}_3$. We want to find how S_ξ transforms when changing the frame of coordinates. We can represent with the help of (1.1.3) the unit volume element with the triple product $V_\xi = (\mathbf{e}_1 \times \mathbf{e}_2) \cdot \mathbf{e}_3$, which coincides with $V_\xi = \|(\mathbf{e}_1 \times \mathbf{e}_2)\|\|\mathbf{e}_3\|$ because of the orthogonality of the canonical basis. Then using the parametrization formerly introduced, due to Lemma 1.1.1, we have that the volume V_h can be expressed as

$$\begin{aligned}
\mathbf{t}_3 \cdot (\mathbf{t}_1 \times \mathbf{t}_2) &= \det(\nabla_\xi h)\mathbf{e}_3 \cdot (\mathbf{e}_1 \times \mathbf{e}_2) \\
(\nabla_\xi h)\mathbf{e}_3 \cdot (\mathbf{t}_1 \times \mathbf{t}_2) &= \det(\nabla_\xi h)\|\mathbf{e}_1 \times \mathbf{e}_2\| \\
\mathbf{e}_3(\nabla_\xi h)^T \cdot (\mathbf{t}_1 \times \mathbf{t}_2) &= \det(\nabla_\xi h) S_\xi.
\end{aligned} \tag{1.1.6}$$

On the other hand we have that, if θ is the angle between $\mathbf{t}_3$ and $\mathbf{n}$,

$$S_h \|\mathbf{t}_3\| \cos\theta = \mathbf{t}_3 \cdot (\mathbf{t}_1 \times \mathbf{t}_2) = V_h.$$

Since $\cos\theta = (\frac{\mathbf{t}_3}{\|\mathbf{t}_3\|} \cdot \mathbf{n})$ and $S_h = (\mathbf{t}_1 \times \mathbf{t}_2) \cdot \mathbf{n}$, the previous equation can be equivalently rewritten as follows

$$\begin{aligned}
(\frac{\mathbf{t}_3}{\|\mathbf{t}_3\|} \cdot \mathbf{n})(\mathbf{t}_1 \times \mathbf{t}_2) \cdot \mathbf{n} &= (\mathbf{t}_1 \times \mathbf{t}_2) \cdot \frac{\mathbf{t}_3}{\|\mathbf{t}_3\|} \\
\frac{\nabla_\xi h \mathbf{e}_3 \cdot (\nabla_\xi h)^{-T}\mathbf{e}_3}{\|\nabla_\xi h \mathbf{e}_3\|\|(\nabla_\xi h)^{-T}\mathbf{e}_3\|} S_h &= V_h \frac{1}{\|\nabla_\xi h \mathbf{e}_3\|}.
\end{aligned}$$

Being $\nabla_\xi h \mathbf{e}_3 \cdot (\nabla_\xi h)^{-T}\mathbf{e}_3 = 1$, reordering the previous expression we obtain

$$S_h = (\mathbf{t}_1 \times \mathbf{t}_2) \cdot \mathbf{n} = \|(\nabla_\xi h)^{-T}\mathbf{e}_3\| V_h.$$

Thus, substituting (1.1.6) we obtain

$$S_h = \|(\nabla_\xi h)^{-T}\mathbf{e}_3\| \det(\nabla_\xi h) S_\xi,$$

which yields the expected result. □

For the standard derivation of the continuum mechanics equations there is no need to introduce a local coordinate system. We therefore use for the rest of this chapter the Cartesian coordinates of $\mathbb{R}^3$. The differential form $d\widehat{\Omega} = d\widehat{x}_1 \wedge d\widehat{x}_2 \wedge d\widehat{x}_3$, represents the element of volume in the reference configuration, and we call $d\widehat{x}$ the vector $d\widehat{x} = d\widehat{x}_1\mathbf{e}_1 + d\widehat{x}_2\mathbf{e}_2 + d\widehat{x}_3\mathbf{e}_3$.

We summarize in this paragraph the main basic concepts needed for the description of the equations governing the continuum mechanics. The interested reader can find a more extensive description in [FQV09, Ch.3].

We define a bounded open deformed domain $\Omega_t \subset \mathbb{R}^3$ which represents the body in the deformed configuration at fixed time $t \in T \subset \mathbb{R}$. A *deformation* is a one-to-one regular map $\phi_t : \widehat{\Omega} \to \Omega_t$. To each deformation it is possible to associate the *displacement* vector field $\widehat{\mathbf{d}} : \widehat{\Omega} \to \mathbb{R}^3$, such that $\widehat{\mathbf{d}}(\widehat{x}) = \phi_t(\widehat{x}) - \widehat{x}$. The deformation gradient $\mathbf{F}(\widehat{x}) = \nabla_{\widehat{x}}\phi_t(\widehat{x})$ is one of the fundamental bricks used to describe the mechanics of continuous media. We can now introduce the right Cauchy–Green strain tensor, $\mathbf{C} = \mathbf{F}^T\mathbf{F}$. Its definition is analogous to that of a *metric tensor* $\mathbf{G}$. Indeed the Cauchy–Green strain tensor represents the change of metric due to the deformation: the distance $\widehat{\boldsymbol{\delta}}$ between two points P and Q in the reference configuration becomes in the deformed configuration

$$\|\phi_t(\widehat{\boldsymbol{\delta}})\| = \|\nabla_{\widehat{x}}\phi_t|_P\widehat{\boldsymbol{\delta}}\| + o(\|\widehat{\boldsymbol{\delta}}\|) = \sqrt{(\widehat{\boldsymbol{\delta}})^T\mathbf{F}^T\mathbf{F}\widehat{\boldsymbol{\delta}}} + o(\|\widehat{\boldsymbol{\delta}}\|).$$

If $x = \phi_t(\widehat{x}) \in \Omega_t$, this expression in differential form reads

$$\|dx\| = \sqrt{d\widehat{x}^T\mathbf{F}^T\mathbf{F}d\widehat{x}}.$$

Exploiting the relation between the differentials one can obtain

$$\int_{\Omega_t} d\Omega_t = \int_{\widehat{\Omega}} \det\mathbf{F}\, d\widehat{\Omega}, \tag{1.1.7}$$

which represents the volume of the domain in the current configuration. Thus the determinant of the deformation gradient $J = \det\mathbf{F}$ has the same role played in Lemma 1.1.1 by $\det(\nabla_\xi h)$, i.e., it measures the change of volume in the transformation.

A key element in the context of continuum mechanics is the following definition.

Definition 1.1.1 (Piola transform)**.** *Given a sufficiently regular second order tensor field $\boldsymbol{\sigma}$ defined in $\widehat{\Omega}$, we define its* Piola transform *as $\boldsymbol{\Pi} : \widehat{\Omega} \to \mathbb{R}^{3\times3}$ such that*

$$\boldsymbol{\Pi}(\widehat{x}) = \mathcal{P}(\boldsymbol{\sigma}(\widehat{x})) = J(\widehat{x})\mathbf{F}^{-1}(\widehat{x})\boldsymbol{\sigma}(\widehat{x}). \tag{1.1.8}$$

We can notice the analogy of the Piola transform with the surface measure previously introduced by means of the cofactor matrix $\mathrm{cof}\,(\mathbf{F})$. The following result comes from a direct calculation

Proposition 1.1.3 (Piola identity)**.**

$$\nabla_{\widehat{x}}\cdot(J\mathbf{F}^{-T}) = 0. \tag{1.1.9}$$

Proof. A possible proof makes use of Nanson's formula applied to the transformation ϕ_t and Gauss's divergence theorem: given a constant nonzero field $\mathbf{f}$ and an arbitrary subset $\widehat{\omega}$ of $\widehat{\Omega}$, we have

$$\begin{aligned} 0 = \int_\omega \nabla_x\cdot\mathbf{f}\; d\Omega_t = \int_{\partial\omega} \mathbf{f}\cdot\mathbf{n}\; d\gamma = \int_{\partial\widehat{\omega}} J\mathbf{f}\,\mathbf{F}^{-T}\cdot\widehat{\mathbf{n}}\; d\widehat{\gamma} = \\ = \int_{\widehat{\omega}} \nabla_{\widehat{x}}\cdot(J\mathbf{f}\,\mathbf{F}^{-T})\; d\widehat{\Omega} = \mathbf{f}\int_{\widehat{\omega}} \nabla_{\widehat{x}}\cdot(J\mathbf{F}^{-T})\; d\widehat{\Omega}. \end{aligned}$$

The thesis follows from the fact that $\widehat{\omega}$ is arbitrary and $\mathbf{f}$ is different from zero. □

Proposition 1.1.4. *The following property of the Piola transform holds*

$$\nabla_{\widehat{x}} \cdot \mathbf{\Pi} = J \nabla_x \cdot \boldsymbol{\sigma}. \tag{1.1.10}$$

Proof. We recall that the operator $\nabla_{\widehat{x}} = \mathbf{F} \nabla_x$ transforms like a covariant vector. Then we can write, using the Piola identity $\nabla_{\widehat{x}} \cdot (J\mathbf{F}^{-T}) = 0$,

$$\nabla_{\widehat{x}} \cdot \mathbf{\Pi} = \nabla_{\widehat{x}} \cdot (\boldsymbol{\sigma} J \mathbf{F}^{-T}) = \nabla_{\widehat{x}} \cdot \boldsymbol{\sigma} J \mathbf{F}^{-T} = \nabla_x \cdot \boldsymbol{\sigma} J.$$

□

This proposition sheds light on the meaning of the Piola transform in relation with the definition of measure given in Proposition 1.1.2. In fact using the divergence theorem

$$\int_{\partial \widehat{\omega}} \mathbf{\Pi} \widehat{\mathbf{n}} \, d\widehat{\gamma} = \int_{\widehat{\omega}} \nabla_{\widehat{x}} \cdot \mathbf{\Pi} \, d\widehat{\Omega} = \int_{\omega} J \nabla_x \cdot \boldsymbol{\sigma} \, d\Omega_t = \int_{\partial \omega} \boldsymbol{\sigma} \mathbf{n} \, d\gamma.$$

Thus the quantity $J \| \mathbf{F}^{-T} \cdot \widehat{\mathbf{n}} \|$ is a measure of the change of surface induced by the deformation ϕ_t.

1.2 Lagrangian, Eulerian and ALE Formulations

The *motion* is a smooth function $\phi : \widehat{\Omega} \times \mathbb{R}^+ \to \Omega_t \subset \mathbb{R}^3$ such that $\phi(\widehat{x}, t) = \phi_t(\widehat{x})$ represents a deformation evolving in time.

The invertibility of the deformation ϕ_t allows us to write the equations either in the reference domain or in the deformed one. The image of a point $\widehat{x} \in \widehat{\Omega}$ in the reference configuration through the function ϕ_t, $\phi_t(\widehat{x}) \in \Omega_t$, is the representation of the *material point* $\widehat{x}$ in its deformed configuration. The description of the mechanics of continuous media with respect to the material points $\widehat{x}$ is called *Lagrangian description*, or *material description*. A scalar or vector field $\widehat{V}$ is called *Lagrangian* if it is defined in $\widehat{\Omega}$.

Another possible description of the dynamics, usually adopted in fluid mechanics, is the *Eulerian*, or *spatial*, description. A scalar or vector field is called *Eulerian* if it is defined in Ω_t. An Eulerian vector field $V(x,t)$ is written in Lagrangian form when it depends on the coordinates $\widehat{x}$ (which are constant in time) of the reference domain $\widehat{\Omega}$ and on time: $V(\phi_t(\widehat{x}), t) = \widehat{V}(\widehat{x}, t)$. The Eulerian description involves the definition of a fixed control volume $\mathbf{V}_C$ in the deformed configuration, such that it remains contained in the deformed configuration for all the time interval considered, $\mathbf{V}_C \subset \Omega_t \ \forall t \in T \subset \mathbb{R}$. The Eulerian counterpart of the vector field V is $V(x,t)$ for $x \in \mathbf{V}_C \subset \Omega_t$. Notice that $\mathbf{V}_C \subset Im_{\phi_t}(\widehat{\Omega}) \ \forall t \in T$ and we can define the counter image of the control volume, $\widehat{\mathbf{V}}_C = \{\widehat{x} \in \widehat{\Omega} : \widehat{x} = {\phi_t}^{-1}(x), x \in \mathbf{V}_C\}$. In the Eulerian formulation the vector fields are written with respect to the variable x which is in the current configuration, and thus it depends on time.

The *Eulerian or spatial time derivative* of an Eulerian vector field is the partial derivative with respect to time, which reads $\partial_t V(x,t) = \frac{\partial V}{\partial t}(x,t)$. Since the point $x = \phi(\widehat{x}, t)$ depends on time, however, to express the total derivative we need to use the chain rule

$$D_t V(x,t) = \frac{dV}{dt}(x,t) = \frac{\partial V}{\partial x} \frac{\partial \phi}{\partial t}({\phi_t}^{-1}(x), t) + \frac{\partial V(x,t)}{\partial t}. \tag{1.2.1}$$

The total derivative is also called *material* or *Lagrangian* derivative. The total time derivative in the Lagrangian formulation coincides with the partial derivative, since the material points in the reference domain $\widehat{x}$ are fixed

$$\frac{d\widehat{V}(\widehat{x},t)}{dt} = \frac{\partial \widehat{V}(\widehat{x},t)}{\partial t}.$$

A particularly important vector field is the *velocity of the material points* $\mathbf{u}$, which is defined as the partial time derivative of the displacement:

- Lagrangian velocity $\widehat{\mathbf{u}} = D_t\phi(\widehat{x},t)$
- Eulerian velocity $\mathbf{u} = \partial_t\phi(\phi^{-1}(x,t),t)$.

In general the Lagrangian frame of reference is used in solid mechanics, while the Eulerian one is preferred in fluid mechanics. This is due to the following main reasons:

1. The constitutive relations of the solids in general involve the displacement, thus the deformation function is actually used to compute the solid stresses and the computation of the deformation gradient cannot be avoided. On the other hand in fluid mechanics the stresses depend in general on the gradient of the velocity vector. So they do not depend on the history of the material displacement and the solution can be found using only quantities on the current domain.

2. In solid mechanics it is usually necessary to impose boundary conditions on the material boundary, which is moving with the particles. In fluid mechanics it is more common to impose boundary conditions on fixed boundaries, which are crossed by the material fluid particles.

In some applications (e.g. fluid–structure interaction, free surface problems) the goal is to solve the equations for the fluid dynamics in a moving domain and the physical boundary conditions should be imposed on the moving (material) boundary. In these cases we cannot define a fixed control volume, because this prevents the imposition of the boundary conditions on the true boundary. However the point one above is still valid. Thus in order to account for the displacement of the fluid domain it is possible to modify the Eulerian frame of reference so that the control volume is no longer constant, but it follows the material particles on the moving boundary. Note that this constraint on the displacement of the fluid domain involves only the boundary, thus the domain displacement is arbitrary in the domain interior. This idea is at the basis of the Arbitrary Lagrangian Eulerian (ALE) description.

To implement it we need to define a reference control volume $\widehat{\Omega}_{\mathcal{A}} \subset \mathbb{R}^3$ and an arbitrary map $\mathcal{A} : \widehat{\Omega}_{\mathcal{A}} \times \mathbb{R}^+ \to \Omega_{\mathcal{A}} \subset \Omega_t$ that for any time t maps the reference control volume to the arbitrary domain $\Omega_{\mathcal{A}}$ in the deformed configuration. Let us write the equality of the partial derivatives of the Eulerian and ALE representations of a vector field V. If $\widetilde{x} \in \widehat{\Omega}_{\mathcal{A}}$, $x \in \Omega_{\mathcal{A}}$ and $\widehat{x} \in \widehat{\Omega}$, having that $\phi(\widehat{x},t) = x = \mathcal{A}(\widetilde{x},t)$,

$$V(\phi(\widehat{x},t),t) = V(\mathcal{A}(\widetilde{x},t),t). \tag{1.2.2}$$

The corresponding partial derivatives read

$$\frac{\partial V}{\partial t} \circ \phi(\widehat{x},t) = \frac{\partial V}{\partial t} \circ \mathcal{A}(\widetilde{x},t). \tag{1.2.3}$$

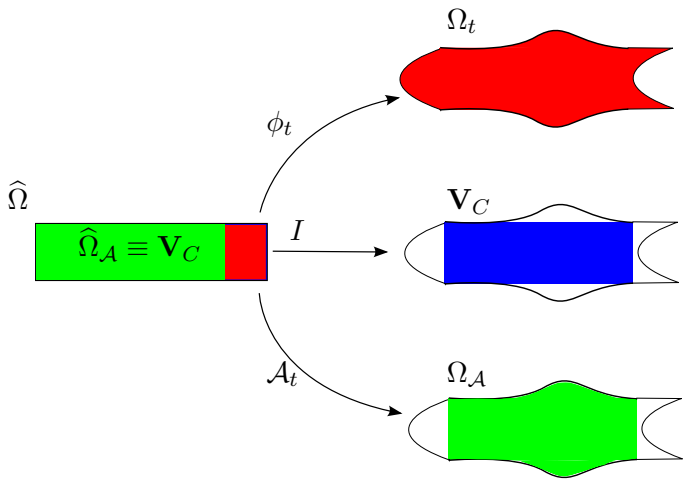


Figure 1.2: Sketch representing a possible choice for the different descriptions. From top to bottom we have the Lagrangian description, the Eulerian, and an ALE one.

This equality allows us in the next section to derive the conservation equations in ALE form.

We can define the *ALE derivative* as the total derivative of the ALE field, i.e.

$$\left.\frac{\partial V}{\partial t}\right|_{\widetilde{x}} (\mathcal{A}(\widetilde{x},t),t) = \frac{\partial V}{\partial t}(x,t) + \frac{\partial \mathcal{A}}{\partial t}(\widetilde{x},t)\frac{\partial V}{\partial x}(x,t). \tag{1.2.4}$$

In the following we denote with $\mathbf{w}$ the *domain velocity* in the deformed configuration $\Omega_{\mathcal{A}}$, $\mathbf{w} = \partial_t \mathcal{A} \circ \mathcal{A}^{-1}$, and with $\boldsymbol{\beta}$ the relative velocity $\boldsymbol{\beta} = \mathbf{u} - \mathbf{w}$ of the particles.

Notice that by substituting in (1.2.1) and using (1.2.3) we can obtain an expression for the material derivative of an ALE field:

$$\frac{dV}{dt} = \left.\frac{\partial V}{\partial t}\right|_{\widetilde{x}} + \boldsymbol{\beta}\nabla_x V.$$

In the practical applications the most suitable reference volume $\widehat{\Omega}_{\mathcal{A}}$ often consists of a part of the undeformed reference domain $\widehat{\Omega}$. A sketch of possible domains used in the different formulations is provided in Figure 1.2.

We remark that the Eulerian and Lagrangian descriptions can be found by choosing as ALE map the identity map $\mathcal{A} = I$ (which gives $\mathbf{w} = 0$) or $\mathcal{A} = \phi$ (which leads to $\mathbf{w} = \mathbf{u}$), respectively.

The following property holds for the total time derivative of the Jacobian determinant J

$$D_t J = J\nabla_x \cdot \mathbf{u}. \tag{1.2.5}$$

The proof of this formula is postponed to Section §3.4 (equation (3.4.7)) where the derivatives of domain functionals are discussed.

Let us define the arbitrary domains $\omega \subset \mathbf{V}_C$ and $\omega_{\mathcal{A}} \subset \Omega_{\mathcal{A}}$. Using (1.2.5) we can derive the Reynolds transport theorem.

Theorem 1.2.1 (Reynolds transport theorem)**.** *With the previous notations, let $\alpha(x,t)$ be a scalar field in the Eulerian frame of reference. Then*

$$D_t \int_\omega \alpha(x,t)\, d\Omega_t = \int_\omega \frac{\partial \alpha(x,t)}{\partial t}\, d\Omega_t + \int_{\partial\omega} \alpha(x,t)\mathbf{u}\cdot\mathbf{n}\, d\gamma. \tag{1.2.6}$$

Proof. We recall a proof reported e.g. in [SH07]. It is sufficient to recast the integrals to the reference configuration and then to exploit the fact that $\widehat{\Omega}$ is fixed in time:

$$D_t \int_\omega \alpha(x,t)\, d\Omega_t = D_t \int_{\widehat{\omega}} J\alpha(x,t)\, d\widehat{\Omega} = \int_{\widehat{\omega}} [D_t(J)\alpha(x,t) + JD_t(\alpha(x,t))]\, d\widehat{\Omega},$$

using the definition of material derivative (1.2.1) and the formula (1.2.5) the total derivative becomes

$$\int_{\widehat{\omega}} J\left[\alpha(x,t)\nabla_x\cdot\mathbf{u} + \frac{\partial(\alpha(x,t))}{\partial t} + \nabla_x\alpha(x,t)\cdot\mathbf{u}\right]\, d\widehat{\Omega}.$$

Eventually, changing again frame of reference and using the divergence theorem,

$$D_t \int_\omega \alpha(x,t)\, d\Omega_t = \int_\omega \left[\alpha(x,t)\nabla_x\cdot\mathbf{u} + \frac{\partial(\alpha(x,t))}{\partial t} + \nabla_x\alpha(x,t)\cdot\mathbf{u}\right]\, d\Omega_t =$$

$$= \int_\omega \frac{\partial\alpha(x,t)}{\partial t}\, d\Omega_t + \int_{\partial\omega} \alpha(x,t)\mathbf{u}\cdot\mathbf{n}\, d\gamma.$$

□

The ALE counterpart of this theorem is the following

Theorem 1.2.2 (Leibnitz transport theorem)**.** *With the previous notations, let $\alpha(x,t)$ be an ALE scalar field in the Eulerian frame of reference.Then*

$$D_t \int_{\omega_{\mathcal{A}}} \alpha(x,t)\, d\Omega_t = \int_{\omega_{\mathcal{A}}} \frac{\partial\alpha(x,t)}{\partial t}\, d\Omega_t + \int_{\partial\omega_{\mathcal{A}}} \alpha(x,t)\mathbf{w}\cdot\mathbf{n}\, d\gamma. \tag{1.2.7}$$

The proof of this result is very similar to the previous one, the interested reader may refer to [SH07]. Also in Section §3.4.1 these results are derived directly from the differentiation of a shape functional. These theorems clearly hold true also for vector fields.

It is worth to point out that using the ALE formulation, as stressed in [BCHZ08], the time derivatives are taken in the reference space–time domain, while the spatial derivatives are taken in the deformed one. The difference is that in the ALE case the time derivative is made by keeping the point $\widetilde{x} \in \widehat{\Omega}_{\mathcal{A}}$ fixed, while in the Eulerian case the point $x \in \Omega_t$ is kept fixed.

1.3 The Equations of Continuum Mechanics

In this section we derive the general form of scalar or vectorial conservation laws in the Eulerian and ALE frames of reference.

Let us denote $\alpha(x,t)$ and $\boldsymbol{\alpha}(x,t)$ a scalar and a vectorial field on Ω_t, respectively. The conservation law in Eulerian form for a scalar field α reads

$$D_t \int_\omega \alpha(x,t)\, d\Omega_t = \int_{\partial\omega} \boldsymbol{\delta}(x,t)\cdot\mathbf{n}\, d\gamma + \int_\omega b(x,t)\, d\Omega_t, \tag{1.3.1}$$

where the quantity $\int_{\partial\omega} \boldsymbol{\delta}(x,t)\cdot\mathbf{n}\, d\gamma$ is the flux of α across the boundary $\partial\omega$, while $\boldsymbol{\delta}$ is an Eulerian vector field defined in Ω_t determining the flux, and the scalar function b is the source/sink term. The vectorial counterpart is

$$D_t \int_\omega \boldsymbol{\alpha}(x,t)\, d\Omega_t = \int_{\partial\omega} \Theta(x,t)\cdot\mathbf{n}\, d\gamma + \int_\omega \mathbf{b}(x,t)\, d\Omega_t, \tag{1.3.2}$$

where Θ is a second order tensor field and $\mathbf{b}$ is a vector field. We omit in the following the dependence on (x,t).

Assuming the smoothness of the displacement map $\widehat{\mathbf{d}}$, using the Reynolds transport theorem 1.2.1, we can re-write the conservation equations in another form:

$$\int_\omega \frac{\partial\alpha}{\partial t}\, d\Omega_t = \int_{\partial\omega} (\boldsymbol{\delta} - \alpha\mathbf{u})\cdot\mathbf{n}\, d\gamma + \int_\omega b\, d\Omega_t, \tag{1.3.3}$$

$$\int_\omega \frac{\partial\boldsymbol{\alpha}}{\partial t}\, d\Omega_t = \int_{\partial\omega} (\Theta - \boldsymbol{\alpha}\otimes\mathbf{u})\cdot\mathbf{n}\, d\gamma + \int_\omega \mathbf{b}\, d\Omega_t.$$

Using the divergence theorem we can write (e.g. for the scalar field)

$$\int_\omega \frac{\partial\alpha}{\partial t}\, d\Omega_t = \int_\omega \nabla_x\cdot(\boldsymbol{\delta} - \alpha\mathbf{u})\, d\Omega_t + \int_\omega b\, d\Omega_t.$$

Using the localization argument, due to the arbitrariness of the domain ω,

$$\frac{\partial\alpha}{\partial t} = \nabla_x\cdot(\boldsymbol{\delta} - \alpha\mathbf{u}) + b \text{ in } \omega, \qquad \frac{\partial\boldsymbol{\alpha}}{\partial t} = \nabla_x\cdot(\Theta - \boldsymbol{\alpha}\otimes\mathbf{u}) + \mathbf{b} \text{ in } \omega. \tag{1.3.4}$$

The same considerations are valid if we consider the scalar and vector fields in an ALE representation. To retrieve the ALE counterpart of the conservation laws we first consider the scalar field.

Reordering the Leibnitz formula (1.2.7) on the volume $\omega_{\mathcal{A}}$ we have

$$\int_{\omega_{\mathcal{A}}} \frac{\partial\alpha}{\partial t}\, d\Omega_t = D_t \int_{\omega_{\mathcal{A}}} \alpha\, d\Omega_t - \int_{\partial\omega_{\mathcal{A}}} \alpha\mathbf{w}\cdot\mathbf{n}\, d\gamma. \tag{1.3.5}$$

Proceeding like in the proof of Theorem 1.2.1, i.e., recasting to the reference configuration to pass the time derivative under the integral sign, we obtain

$$\int_{\omega_{\mathcal{A}}} \frac{\partial\alpha}{\partial t}\, d\Omega_t = \int_{\omega_{\mathcal{A}}} {J_{\mathcal{A}}}^{-1} \left.\frac{\partial J_{\mathcal{A}}\alpha}{\partial t}\right|_{\widetilde{x}} d\Omega_t - \int_{\partial\omega_{\mathcal{A}}} \alpha\mathbf{w}\cdot\mathbf{n}\, d\gamma. \tag{1.3.6}$$

Substituting in the conservation law (1.3.3) (on the domain $\omega_{\mathcal{A}}$ instead of ω and using (1.2.3)) yields

$$\int_{\omega_{\mathcal{A}}} {J_{\mathcal{A}}}^{-1} \left.\frac{\partial J_{\mathcal{A}}\alpha}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\partial\omega_{\mathcal{A}}} (\boldsymbol{\delta} - \alpha\boldsymbol{\beta})\cdot\mathbf{n}\, d\gamma + \int_{\omega_{\mathcal{A}}} b\, d\Omega_t. \tag{1.3.7}$$

This form is hybrid, since the integrals are taken in the domain $\omega_{\mathcal{A}}$ which is in the current configuration, while the integrands still depend on the quantity $J_{\mathcal{A}}$. However equation (1.3.7) can be further manipulated by computing the derivative on the left hand side and using the divergence theorem on the right hand side:

$$\int_{\omega_{\mathcal{A}}} {J_{\mathcal{A}}}^{-1} \frac{dJ_{\mathcal{A}}}{dt}\alpha + {J_{\mathcal{A}}}^{-1} J_{\mathcal{A}} \left.\frac{\partial\alpha}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\omega_{\mathcal{A}}} \nabla_x\cdot(\boldsymbol{\delta} - \alpha\boldsymbol{\beta}) + b\, d\Omega_t,$$

then, using a formula analogous to (1.2.5) with $J_{\mathcal{A}}$ and $\mathbf{w}$ instead of J and $\mathbf{u}$

$$\int_{\omega_{\mathcal{A}}} \nabla_x{\cdot}\mathbf{w}\alpha + \left.\frac{\partial \alpha}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\omega_{\mathcal{A}}} \nabla_x{\cdot}(\boldsymbol{\delta} - \alpha\boldsymbol{\beta}) + b\, d\Omega_t.$$

Eventually, recalling the definition of $\boldsymbol{\beta} = \mathbf{u} - \mathbf{w}$, we obtain

$$\int_{\omega_{\mathcal{A}}} \left.\frac{\partial \alpha}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\omega_{\mathcal{A}}} \nabla_x{\cdot}(\boldsymbol{\delta} - \alpha\mathbf{u}) + \mathbf{w}\cdot\nabla_x\alpha + b\, d\Omega_t. \tag{1.3.8}$$

In an analogous way, for a vectorial field,

$$\int_{\omega_{\mathcal{A}}} \left.\frac{\partial \boldsymbol{\alpha}}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\omega_{\mathcal{A}}} \nabla_x{\cdot}(\Theta - \boldsymbol{\alpha}\otimes\mathbf{u}) + (\mathbf{w}\cdot\nabla_x)\boldsymbol{\alpha} + \mathbf{b}\, d\Omega_t. \tag{1.3.9}$$

This equation is valid on any domain $\omega_{\mathcal{A}} \subset \Omega_{\mathcal{A}}$, thus for the localization argument it holds pointwise

$$\left.\frac{\partial \boldsymbol{\alpha}}{\partial t}\right|_{\widetilde{x}} = \nabla_x{\cdot}(\Theta - \boldsymbol{\alpha}\otimes\mathbf{u}) + (\mathbf{w}\cdot\nabla_x)\boldsymbol{\alpha} + \mathbf{b}.$$

The last equation leads to the *non-conservative* form of the fluid momentum conservation law, as discussed in Section §1.4. Notice that to write this equation we used the Reynolds transport formula (1.2.6) and the Euler formula (1.2.5).

As for the Eulerian representation, the time derivatives can be brought out of the integral sign by substituting directly (1.3.5) on the conservation law (1.3.3) (on $\omega_{\mathcal{A}}$), which gives

$$D_t \int_{\omega_{\mathcal{A}}} \alpha\, d\Omega_t = \int_{\partial\omega_{\mathcal{A}}} (\boldsymbol{\delta} - \alpha\boldsymbol{\beta})\cdot\mathbf{n}\, d\gamma + \int_{\omega_{\mathcal{A}}} b\, d\Omega_t, \tag{1.3.10}$$

and

$$D_t \int_{\omega_{\mathcal{A}}} \boldsymbol{\alpha}\, d\Omega_t = \int_{\partial\omega_{\mathcal{A}}} (\Theta - \boldsymbol{\alpha}\otimes\boldsymbol{\beta})\cdot\mathbf{n}\, d\gamma + \int_{\omega_{\mathcal{A}}} \mathbf{b}\, d\Omega_t. \tag{1.3.11}$$

These equations lead in Section §1.4 to define the *conservative* form of the fluid momentum equation.

Notations: *we denote the general ALE reference domain $\widehat{\Omega}_{\mathcal{A}}$ considered so far as the* fluid reference domain $\widehat{\Omega}^f$*, which represents the portion of space occupied by the fluid in the reference configuration. The quantities referring to the fluid domain will be marked with the f label. In the same way we introduce the* solid reference domain $\widehat{\Omega}^s$*, which represents the portion of space occupied by the solid in the reference configuration. All the solid quantities will have the label s.*

The variables considered in the FSI model will be: the fluid velocity $\mathbf{u}$, the fluid pressure p, the fluid domain displacement $\widehat{\mathbf{d}}_f$ (introduced because of the ALE representation of the fluid), the solid displacement $\widehat{\mathbf{d}}_s$. $\mathbf{u}$ and p are taken in the current configuration, while $\widehat{\mathbf{d}}_f$ and $\widehat{\mathbf{d}}_s$ are taken in the reference one.

In the following sections we write the conservation equations that are used in our FSI formulation. For the momentum conservation equations we need to introduce the *Cauchy stress tensor*. This tensor derives from the *Cauchy's theorem* stating that the *traction vector* $\mathbf{t}$ on a surface S, such that the force exerted on S reads $\int_S \mathbf{t}\, d\gamma$, is a linear function of the normal $\mathbf{n}$ to the surface S. This theorem implies that there exists a unique tensor σ called

Cauchy stress tensor, such that $\mathbf{t} = \sigma\mathbf{n}$. Its symmetry can be easily shown (see e.g. [Cur04, Ch.3.8]).

Notations: *We denote by σ_f and σ_s the Cauchy stress tensors for the fluid and the solid respectively in the deformed configuration. Their counterparts represented in the reference configuration (called* Piola-Kirchhof stress tensors*) are denoted $\widehat{\sigma}_f$ and $\mathbf{\Pi}$ respectively.*

1.4 The Equations for a Fluid

In this section we report the conservation equations for mass and momentum on moving domains written in the Eulerian and ALE frames. For an incompressible Newtonian fluid these conservation laws describe the Navier–Stokes equations.

We start by considering some conservation equations for a generic fluid in Eulerian and ALE form. The mass conservation equation is obtained by taking the *density* of the continuum medium as the scalar field α of the previous section. The flux and the source/sink terms corresponding to $\boldsymbol{\delta}$ and b are in this case zero. Thus we have, substituting in (1.3.3),

$$\int_{\omega} \frac{\partial \rho_f}{\partial t}\, d\Omega_t = \int_{\omega} -\nabla_x\cdot(\rho_f \mathbf{u})\, d\Omega_t. \tag{1.4.1}$$

If the fluid is incompressible, then the density ρ_f is constant and the left hand side vanishes. Thus (1.4.1) becomes the classical mass conservation equation appearing in the Navier–Stokes equations

$$\nabla_x\cdot\mathbf{u} = 0 \text{ in } \Omega_t, \tag{1.4.2}$$

where we used the localization argument.

Expression (1.4.1), which is written now in Eulerian representation, can be written in ALE form substituting in (1.3.8):

$$\int_{\omega_{\mathcal{A}}} \left.\frac{\partial \rho_f}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\omega_{\mathcal{A}}} -\nabla_x\cdot(\rho_f \mathbf{u}) + \mathbf{w}\cdot\nabla_x\rho_f\, d\Omega_t. \tag{1.4.3}$$

To write the momentum conservation equation we consider as vector field in (1.3.3) $\rho_f\mathbf{u}$, while the flux vector $\sigma_f\cdot\mathbf{n}^f$ is expressed as function of the fluid velocity $\mathbf{u}$ in a *constitutive relation.* The source/sink term $\mathbf{f}_f$ represents the momentum generated by the volume forces acting on the fluid. We write here the momentum equation directly in ALE form, substituting the definitions of $\boldsymbol{\alpha}$, Θ, and $\mathbf{b}$ in (1.3.9):

$$\int_{\omega_{\mathcal{A}}} \left.\frac{\partial (\rho_f\mathbf{u})}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\omega_{\mathcal{A}}} \nabla_x\cdot(\sigma_f - \rho_f\mathbf{u}\otimes\mathbf{u}) + \mathbf{w}\cdot\nabla_x(\rho_f\mathbf{u}) + \mathbf{f}_f\, d\Omega_t. \tag{1.4.4}$$

This equation represents the non-conservative form of the momentum balance. It can be simplified using standard algebra

$$\int_{\omega_{\mathcal{A}}} \left.\frac{\partial \rho_f}{\partial t}\right|_{\widetilde{x}} \mathbf{u} + \rho_f \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\omega_{\mathcal{A}}} \nabla_x\cdot\sigma_f - \nabla_x\cdot(\rho_f\mathbf{u})\mathbf{u} - \rho_f(\mathbf{u}\cdot\nabla_x)\mathbf{u} + (\mathbf{w}\cdot\nabla_x)\rho_f\mathbf{u} + \mathbf{w}\rho_f\nabla_x\mathbf{u} + \mathbf{f}_f\, d\Omega_t,$$

and by reordering,

$$\int_{\omega_{\mathcal{A}}} \rho_f \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\omega_{\mathcal{A}}} \nabla_x\cdot\sigma_f - \rho_f(\mathbf{u}-\mathbf{w})\cdot\nabla_x\mathbf{u} + \mathbf{f}_f\, d\Omega_t + \int_{\omega_{\mathcal{A}}} \mathbf{u}\left[-\left.\frac{\partial \rho_f}{\partial t}\right|_{\widetilde{x}} - \nabla_x\cdot(\rho_f\mathbf{u}) + \mathbf{w}\nabla_x\cdot\rho_f\right] d\Omega_t. \tag{1.4.5}$$

Due to (1.4.3) the term in brackets vanishes. Using the localization argument, from the arbitrariness of the domain $\omega_{\mathcal{A}}$, the momentum conservation equation can be written in the form

$$\rho_f \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\widetilde{x}} = \nabla_x \cdot \sigma_f - \rho_f(\mathbf{u}-\mathbf{w})\cdot\nabla_x \mathbf{u} + \mathbf{f}_f. \tag{1.4.6}$$

This is the non conservative form of the momenum equation. Writing the conservation of the momentum in conservative form is accomplished by simply substituting the definitions of $\boldsymbol{\alpha}$, Θ, and $\mathbf{b}$ into (1.3.11)

$$D_t \int_{\omega_{\mathcal{A}}} \rho_f \mathbf{u}\, d\Omega_t = \int_{\gamma^f} (\sigma_f - \rho_f \mathbf{u}\otimes\boldsymbol{\beta})\cdot \mathbf{n}^f \, d\gamma + \int_{\omega_{\mathcal{A}}} \mathbf{f}_f \, d\Omega_t. \tag{1.4.7}$$

Then applying the divergence theorem we obtain

$$D_t \int_{\omega_{\mathcal{A}}} \rho_f \mathbf{u}\, d\Omega_t = \int_{\omega_{\mathcal{A}}} \nabla_x \cdot \sigma_f - \nabla_x \cdot (\rho_f \mathbf{u}\otimes\boldsymbol{\beta}) + \mathbf{f}_f \, d\Omega_t. \tag{1.4.8}$$

In fluid dynamics, as previously mentioned, the *stress tensor* usually depends on the velocity $\mathbf{u}$ through a constitutive relation. The constitutive law to be such has to satisfy a number of principles, like the *principle of frame indifference* stating that the constitutive law must be independent of the observer. *Newtonian fluids* correspond to a particular choice for the constitutive equation, when the stress tensor depends linearly on the symmetric part of the velocity gradient

$$\sigma_f = \mu_f(\nabla_x \mathbf{u} + (\nabla_x \mathbf{u})^T) - pI. \tag{1.4.9}$$

Here p denotes the pressure. This constitutive law models incompressible Newtonian viscous fluids. Although these assumptions are usually accepted for a macroscopic description of blood flow in large arteries, the model becomes inappropriate for modeling the hemodynamics in other locations. One of the main limitations of Newtonian fluids in this sense is the constant viscosity. In fact when the velocity decreases the red blood cells tend to interact, increasing the viscosity of blood. This phenomenon is called *shear thinning*, and depends also on the density of red blood cells (*hematocrit*). However taking into account these kind of phenomena on one side introduces further nonlinearities to the model, on the other side it requires the knowledge of more parameters (such as the relation between viscosity and shear rate). See A. Robertson, A. Sequeira, G. Owens [FQV09, Ch.6] and references therein for a deeper discussion about non-Newtonian fluids.

In our model we consider blood as a Newtonian fluid and thus it is reliable only for large arteries. However most of the methods presented in this work do not depend on the type of constitutive equation chosen.

Substituting the constitutive law (1.4.9) in (1.4.6) we have

$$\int_{\omega_{\mathcal{A}}} \rho_f \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\omega_{\mathcal{A}}} \nabla_x \cdot (\mu_f(\nabla_x \mathbf{u} + (\nabla_x \mathbf{u})^T)) - \nabla_x p - \rho_f(\mathbf{u}-\mathbf{w})\cdot\nabla_x \mathbf{u} + \mathbf{f}_f \, d\Omega_t. \tag{1.4.10}$$

Under the hypotheses of constant viscosity and incompressibility (i.e.,$\nabla_x \cdot \mathbf{u} = 0$) this equation can be rewritten as

$$\int_{\omega_{\mathcal{A}}} \rho_f \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\widetilde{x}} d\Omega_t = \int_{\omega_{\mathcal{A}}} \mu_f \Delta_x \mathbf{u} - \nabla_x p - \rho_f(\mathbf{u}-\mathbf{w})\cdot\nabla_x \mathbf{u} + \mathbf{f}_f \, d\Omega_t. \tag{1.4.11}$$

The corresponding conservative form for the momentum conservation is trivially obtained performing the same substitutions done in (1.4.10) and (1.4.11) on equation (1.4.8). Again the momentum conservation equation can be written pointwise using the localization argument as

$$\rho_f \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\widetilde{x}} = \mu_f \Delta_x \mathbf{u} - \nabla_x p - \rho_f (\mathbf{u} - \mathbf{w}) \cdot \nabla_x \mathbf{u} + \mathbf{f}_f.$$

1.5 The Equations for a Solid

In our application we consider a Lagrangian frame of reference to describe the solid deformation. Since the coordinates $\widehat{x}$ are fixed, the conservation equations have a simpler form. The conservation of mass simply reads, $\forall \widehat{\omega} \subset \widehat{\Omega}^s$ and being $\widehat{\rho}_s = J\rho_s$ the *solid density*,

$$0 = D_t \int_{\omega} \rho_s \ d\Omega_t{}^s = \int_{\widehat{\omega}} \frac{\partial J\rho_s}{\partial t} \ d\widehat{\Omega}^s = \int_{\widehat{\omega}} \frac{\partial \widehat{\rho}_s}{\partial t} \ d\widehat{\Omega}^s,$$

which using the localization argument becomes

$$\frac{\partial \widehat{\rho}_s}{\partial t} = 0. \tag{1.5.1}$$

The momentum conservation can be obtained from (1.3.2) by recasting all the integrals back to the reference configuration. The quantity conserved is $\rho_s \frac{\partial(\widehat{\mathbf{d}}_s \circ \mathcal{A}_t{}^{-1})}{\partial t}$ while the flux is given by $\sigma_s \cdot \mathbf{n}^s$, where the *stress tensor* σ_s, as in the case of the fluid, is given by a constitutive law. The momentum conservation reads

$$D_t \int_{\widehat{\omega}} J\rho_s \frac{\partial \widehat{\mathbf{d}}_s}{\partial t} \ d\widehat{\Omega}^s - \int_{\widehat{\omega}} J\nabla_x \cdot \sigma_s \ d\widehat{\Omega}^s = \int_{\widehat{\omega}} J\mathbf{f}_s \ d\widehat{\Omega}^s. \tag{1.5.2}$$

Using (1.5.1), the localization argument and the fact that the domain $\widehat{\omega}$ is fixed we obtain

$$\widehat{\rho}_s \frac{\partial^2 \widehat{\mathbf{d}}_s}{\partial t^2} - J\nabla_x \cdot \sigma_s = \mathbf{f}_s. \tag{1.5.3}$$

However this form is still hybrid, since the divergence is taken with respect to x. Using the Piola transform (1.1.10) we obtain

$$\widehat{\rho}_s \frac{\partial^2 \widehat{\mathbf{d}}_s}{\partial t^2} - \nabla_{\widehat{x}} \cdot \mathbf{\Pi} = \mathbf{f}_s, \tag{1.5.4}$$

where $\mathbf{\Pi} = J\mathbf{F}^{-1}\sigma_s$ is the first Piola–Kirchhof tensor.

$\mathbf{\Pi}$ is non-symmetric. To write the constitutive relation with respect to a symmetric tensor we introduce the second Piola–Kirchhof tensor

$$\mathbf{\Sigma} = \mathbf{F}^{-1}\mathbf{\Pi}. \tag{1.5.5}$$

Instead of using the Cauchy–Green strain tensor in the constitutive relation we rather use the *Green–Lagrange strain tensor*

$$\mathbf{E} = \frac{1}{2}(\mathbf{C} - I). \tag{1.5.6}$$

This tensor is null when there is no deformation, and from the properties of the tensor $\mathbf{C}$ we have

$$\frac{1}{2}(\|dx\|^2 - \|d\widehat{x}\|^2) = d\widehat{x}\mathbf{E}d\widehat{x}.$$

A large variety of materials can be chosen to model the arterial wall. The latter is *pre-stressed*, i.e., even when the structure is not loaded, the stress is different from zero (this can be seen when the artery at rest is cut longitudinally or transversally: in the former case it tends to open, in the latter case it tends to shrink [VV87]). The mechanical response of the large arteries wall to a given strain is mainly due to the *elastin* and *collagen* components. The former one is responsible for the elastic response in physiological conditions, while the latter activates when the strains reach a certain critical value and it is much stiffer. Furthermore the collagen component is made of fibers, which inhibit the elongation along the fiber direction. The arterial tissue is composed mainly by three layers which behave differently: these are, from the vessel lumen to the external wall, *intima*, *media* and *adventitia*. The intima is a thin layer in contact with blood, its mechanical properties can be neglected but it is responsible of the wall reaction (e.g. stiffening) to the blood flow (in terms of response to stresses or chemicals coming from the fluid). Media and adventitia are involved in the mechanical response. To accurately predict the mechanisms of the arterial wall one should take into account these characteristics in a constitutive law. Furthermore, as almost all biological tissues, the arterial wall is incompressible, which introduces another constraint. In literature accurate models for the arterial wall can be found in [HGO00, HSSB02] and more recently in [ZFDR08, RRDH08] We refer to [HO06, Bal06] for an overview of the mechanical properties and models.

Although some arteries show visco-elastic effects, they are usually negligible [Bal06], thus most of the times the arterial wall is modeled as an elastic material. If there exists a scalar valued strain energy function W depending on the strain tensor $\mathbf{E}$ and such that

$$\frac{\partial W}{\partial \mathbf{E}}(\mathbf{E}) = \mathbf{\Sigma}(\mathbf{E}), \tag{1.5.7}$$

then the material is called *hyperelastic*.

A constitutive law must be written in terms of objective quantities, in order to satisfy the frame indifference (or objectivity) principle. Let us consider a change of reference defined by the rotation $R \in SO(3)$ (the group of orthogonal matrices with determinant equal to one). A vector $\mathbf{v}$ is objective if in the new frame has the form $R^T\mathbf{v}$. A matrix M is objective if in the new frame of reference it reads R^TMR.

Let M and $\mathbf{v}$ be an objective matrix and vector respectively. A scalar function $f(M, \mathbf{v})$ is called *isotropic* if it is invariant with respect to rotations, i.e., if $f(M, \mathbf{v}) = f(R^TMR, R^T\mathbf{v})$. A vector function $\mathbf{f}(M, \mathbf{v})$ is isotropic if $R\mathbf{f}(M, \mathbf{v}) = \mathbf{f}(R^TMR^T, R\mathbf{v})$, while a matrix function $\mathcal{F}(M, \mathbf{v})$ is isotropic if $R^T\mathcal{F}(M, \mathbf{v})R = \mathcal{F}(R^TMR, R^T\mathbf{v})$.

If we neglect the collagen fiber orientation, then it is possible to describe the elasticity strain energy using an isotropic function. This has the advantage that thanks to the *representation theorem* (see e.g. [Kor90]) every isotropic function can be expressed in terms of the scalar *invariants* of the argument tensors.

We can consider a generic strain energy function $W(\mathbf{E})$ which depends on the Green–Lagrange strain tensor. Then equation (1.5.7) can be rewritten, exploiting the representation theorem, as

$$\mathbf{\Sigma} = \frac{\partial W}{\partial I_1}\frac{\partial I_1}{\partial \mathbf{E}} + \frac{\partial W}{\partial I_2}\frac{\partial I_2}{\partial \mathbf{E}} + \frac{\partial W}{\partial I_3}\frac{\partial I_3}{\partial \mathbf{E}},$$

where the invariants are $I_1 = \mathrm{tr}(\mathbf{E})$, $I_2 = \frac{[\mathrm{tr}(\mathbf{E})]^2 - \mathrm{tr}(\mathbf{E}^2)}{2}$ and $I_3 = |\mathbf{E}|$. It is possible to compute explicitly the derivative of the invariants:

- The derivative of the fist invariant, since the trace is a linear operator, gives $\frac{\partial I_1}{\partial \mathbf{E}} = I$.
- The derivative of the second invariant gives, from direct calculation,

$$\frac{\partial I_2}{\partial \mathbf{E}} = \mathrm{tr}(\mathbf{E})I - \mathbf{E}.$$

- The derivative of the third invariant comes from the Jacobi's formula for the derivative of a determinant (see e.g. [MN99])

$$\frac{\partial I_3}{\partial \mathbf{E}} = \det(\mathbf{E})(\mathbf{E})^{-1} = \mathrm{cof}\,(\mathbf{E}),$$

  where $\mathrm{cof}\,(\mathbf{E})$ is the cofactor matrix of $\mathbf{E}$.

It is now possible to write explicitly the form of the second Piola–Kirchhof tensor for hyperelastic isotropic materials

$$\mathbf{\Sigma} = \frac{\partial W}{\partial I_1} + \frac{\partial W}{\partial I_2}\mathrm{tr}(\mathbf{E}) - \frac{\partial W}{\partial I_2}\mathbf{E} + \frac{\partial W}{\partial I_3}\mathrm{cof}\,(\mathbf{E}). \tag{1.5.8}$$

A popular isotropic strain energy function is the one defining the Saint Venant–Kirchhof model:

$$W(\mathbf{E}) = \frac{L_1}{2}(\mathrm{tr}\mathbf{E})^2 + L_2\mathrm{tr}(\mathbf{E}^2), \tag{1.5.9}$$

where L_1 and L_2 are the *Lamé coefficients* defining the characteristics of the material. Taking the derivative we obtain

$$\mathbf{\Sigma} = L_1(\mathrm{tr}\mathbf{E})I + 2L_2\mathbf{E}. \tag{1.5.10}$$

Being expression (1.5.10) linear, it can be written in the more general form

$$\mathbf{\Sigma} = H : \mathbf{E},$$

where H is a fourth-order tensor.

The St. Venant–Kirchhof materials are often characterized by the *Young modulus* ϵ and the *Poisson coefficient* ν instead of the Lamé coefficients. The following relations hold between the two sets of coefficients:

$$\epsilon = L_2\frac{3L_1 + 2L_2}{L_1 + L_2}; \qquad \nu = \frac{1}{2}\frac{L_1}{L_1 + L_2};$$
$$L_1 = \frac{\epsilon\nu}{(1-2\nu)(1+\nu)}; \qquad L_2 = \frac{\epsilon}{2(1+\nu)};$$

Note that the constitutive equation for the St. Venant–Kirchhof material is nonlinear in the displacement $\widehat{\mathbf{d}}_s$, because both the tensors $\mathbf{E}$ and $\mathbf{\Sigma}$ are nonlinear in $\mathbf{F}$. A further simplification of the St. Venant–Kirchhof constitutive equation, which consists in neglecting the terms of order higher than one in the definitions of $\mathbf{E}$ and $\mathbf{\Sigma}$, leads to the linear elasticity equation. In particular, if we consider small deformations,

$$\mathbf{E} = \frac{\nabla_{\widehat{x}}\widehat{\mathbf{d}}_s + (\nabla_{\widehat{x}}\widehat{\mathbf{d}}_s)^T + \nabla_{\widehat{x}}\widehat{\mathbf{d}}_s(\nabla_{\widehat{x}}\widehat{\mathbf{d}}_s)^T}{2} \approx \frac{\nabla_{\widehat{x}}\widehat{\mathbf{d}}_s + (\nabla_{\widehat{x}}\widehat{\mathbf{d}}_s)^T}{2} = \mathbf{D},$$

which is the symmetric part of the displacement gradient, and

$$L_1(I_1)I + 2L_2\mathbf{E} = \mathbf{\Sigma} = \mathbf{F}^{-1}\mathbf{\Pi} \approx \mathbf{\Pi}.$$

With these simplifications the Venant–Kirchhof constitutive equation is also called isotropic generalized *Hooke's law*.

The equation of linear elasticity, substituting in (1.5.4), reads

$$\widehat{\rho}_s \frac{\partial^2 \widehat{\mathbf{d}}_s}{\partial t^2} - \nabla_{\widehat{x}} \cdot (L_1(\mathrm{tr}\mathbf{D})I + 2L_2\mathbf{D}) = \mathbf{f}_s. \tag{1.5.11}$$

This equation is close to the nonlinear model under the hypothesis of small deformations. We consider this model in the applications reported in Part III. The implementation of more general materials is currently under development.

Modeling Fluid–Structure Interaction Problems

After setting up the continuum mechanics models in the previous chapter, we can introduce the Fluid–Structure Interaction (FSI) coupling. We describe some of the common approaches to couple and discretize, in both time and space, the physical problems introduced in Sections §1.4 and §1.5. Theoretical considerations about the properties of the coupled problem are discussed in Section §2.8. In Section §2.9 we give an overview of the geometrical multiscale and reduced order models, the coupling between 3D, 1D and 0D models for arteries.

2.1 Coupling Conditions

Notations: *we distinguish in the following between the* Fluid–Structure (FS) interface Γ, *which is the internal boundary separating the fluid from the solid, and the* boundaries, *which consist of the external boundaries of each (fluid or structure) physical domain. Since we focus on hemodynamic problems, we name the fluid boundaries according to the direction of the blood flow: we call* Γ_f^{in} *the inlet boundaries and* Γ_f^{out} *the outlet boundaries. Concerning the structure we call* Γ_s^{in} *and* Γ_s^{out} *the termination* rings *corresponding respectively to* Γ_f^{in} *and* Γ_f^{out}, *while we call* Γ_s^{ext} *the external wall in contact with the tissue surrounding the artery. A schematic picture showing the notations is provided in Figure 2.1. We keep the same notations introduced in the previous chapter, so that all the quantities with the hat* $\hat{}$ *are intended in the reference configuration while those without hat are taken in the current configuration (and therefore depend also on time). Furthermore we introduce the variable* $\mathbf{u}_f = (\mathbf{u}, p)$ *grouping the fluid velocity and pressure.*

We represent the FSI problem in an Arbitrary Lagrangian Eulerian (ALE) frame (cf. Section §1.2). The construction of the ALE map is clarified in the next section.

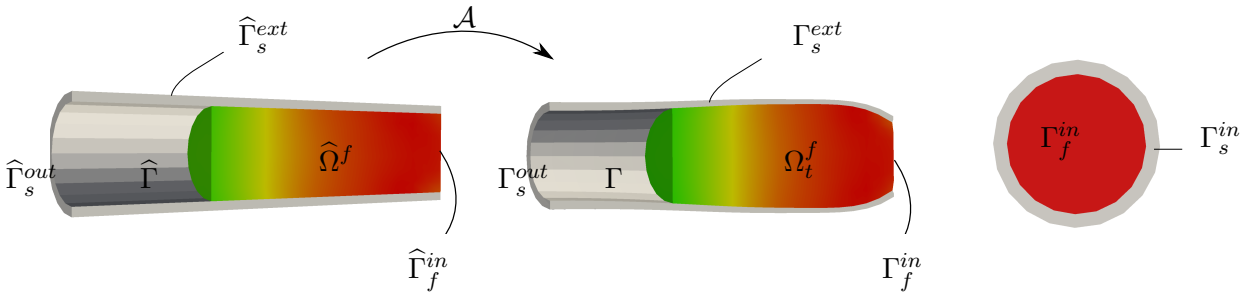


Figure 2.1: Notation for the boundaries and FS interface.

On the FS interface Γ the coupling is expressed through:

- the continuity of the velocity;
- the continuity of the stresses;
- the continuity of the domain displacement (geometric adherence).

The last coupling condition is at a certain extent artificial, since it is introduced by the ALE formulation of the fluid equation. It expresses the constraint on the fluid domain to follow the material particles of the solid on the interface Γ. The system of equations with the coupling conditions reads:

$$\begin{aligned}
\rho_f \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\widetilde{x}} - \nabla_x{\cdot}\sigma_f + \rho_f(\mathbf{u}-\mathbf{w})\cdot\nabla_x\mathbf{u} &= \mathbf{f}_f && \text{in } \Omega_t^f \\
\nabla_x{\cdot}\mathbf{u} &= 0 && \text{in } \Omega_t^f \\
\widehat{\rho}_s \frac{\partial^2 \widehat{\mathbf{d}}_s}{\partial t^2} - \nabla_{\widehat{x}}{\cdot}\boldsymbol{\Pi} &= \mathbf{f}_s && \text{in } \widehat{\Omega}^s \\
\frac{\partial \widehat{\mathbf{d}}_s}{\partial t} &= \mathbf{u}\circ\mathcal{A}_t && \text{on } \widehat{\Gamma} \qquad (2.1.1) \\
\widehat{\sigma}_f\widehat{\mathbf{n}}^f + \boldsymbol{\Pi}\widehat{\mathbf{n}}^s &= 0 && \text{on } \widehat{\Gamma} \qquad (2.1.2) \\
\frac{\partial \widehat{\mathbf{d}}_f}{\partial t} &= \mathbf{w}\circ\mathcal{A}_t && \text{on } \widehat{\Gamma}, \qquad (2.1.3)
\end{aligned}$$

where often the body forces $\mathbf{f}_f$ and $\mathbf{f}_s$ are set to zero in hemodynamic applications (they could account e.g. for the gravity force). We recall that from the Piola transform (1.1.10) we have $\widehat{\sigma}_f = J\mathbf{F}^{-1}\sigma_f$. The last three equations (2.1.1), (2.1.2), and (2.1.3) represent the three coupling conditions. The last one, as anticipated, is the geometric adherence constraint on the ALE map stating that the fluid domain must follow the material FS interface.

2.2 Three Fields Formulation

At every time $t > 0$ the coupled FSI problem can be refounded as being made of three coupled sub-problems. A *fluid problem*

$$\mathsf{F}(\mathbf{u}_f, \widehat{\mathbf{d}}_s, \widehat{\mathbf{d}}_f) = 0, \qquad (2.2.1)$$

describing the physics of the fluid through the equations introduced in Section §1.4. This problem consists in finding the fluid velocity and pressure $\mathbf{u}_f = (\mathbf{u}, p)$ [3] given the solid displacement $\widehat{\mathbf{d}}_s$ (or traction vector $\boldsymbol{\Pi}\widehat{\mathbf{n}}^s$, or a combination of both) and the domain displacement $\widehat{\mathbf{d}}_f$. We do not bind the notation for the fluid problem F to a specific interface condition, neither to a specific form (conservative or not) of the momentum conservation equation. When needed we distinguish the cases of Dirichlet, Neumann, or Robin coupling conditions by calling the fluid problem F_D, F_N, or F_R respectively.

A *solid problem*

$$\mathsf{S}(\mathbf{u}_f, \widehat{\mathbf{d}}_s) = 0, \qquad (2.2.2)$$

[3] to light the notation in the following pages we group the fluid velocity and pressure fields

representing the structure equation introduced in Section §1.5. This problem consists in finding the solid displacement $\widehat{\mathbf{d}}_s$ given the fluid velocity $\mathbf{u}$ (or traction vector $\sigma_f \mathbf{n}^f$, or a combination of both). The solid problem, as the fluid one, is not bound to a specific interface condition (which anyway has to be different from the one chosen for the fluid problem, in order for the coupled problem to be properly defined). Also in this case we distinguish the three possible coupling conditions, Dirichlet, Neumann and Robin, with appropriate notations, S_D, S_N and S_R respectively.

A *geometry problem*

$$\mathsf{G}(\widehat{\mathbf{d}}_s, \widehat{\mathbf{d}}_f) = 0, \tag{2.2.3}$$

defining the ALE map $\mathcal{A}_t$, and thus the fluid computational domain. This problem is coupled to the solid problem through the geometric adherence condition (2.1.3), that we can write equivalently as

$$\widehat{\mathbf{d}}_f = \widehat{\mathbf{d}}_s \text{ on } \widehat{\Gamma}. \tag{2.2.4}$$

The arbitrariness of the ALE approach also reflects in the arbitrary choice of the geometry problem. Our choice, which is quite common in this context, consists in describing the fluid domain displacement $\widehat{\mathbf{d}}_f$ as a harmonic extension of the solid displacement $\widehat{\mathbf{d}}_s|_{\widehat{\Gamma}}$ from the FS interface $\widehat{\Gamma}$ to the interior of the fluid reference domain $\widehat{\Omega}^f \subset \mathbb{R}^3$:

$$\begin{cases} -\Delta \widehat{\mathbf{d}}_f &= 0 & \text{in } \widehat{\Omega}^f \\ \widehat{\mathbf{d}}_f &= \widehat{\mathbf{d}}_s & \text{on } \widehat{\Gamma}, \\ \nabla_{\widehat{x}} \widehat{\mathbf{d}}_f \cdot \widehat{\mathbf{n}}^f &= \mathbf{0} & \text{on } \partial\widehat{\Omega}^f \setminus \widehat{\Gamma}, \end{cases} \tag{2.2.5}$$

The ALE mapping is then defined as

$$\begin{aligned} \mathcal{A}_t : \widehat{\Omega}^f &\rightarrow \Omega_t^f \\ \widehat{x} &\mapsto \mathcal{A}_t(\widehat{x}) = \widehat{x} + \widehat{\mathbf{d}}_f(\widehat{x}), \end{aligned}$$

so that the current domain Ω_t^f is defined as $\Omega_t^f = \mathcal{A}_t(\widehat{\Omega}^f)$.

In literature many other approaches to update the mesh displacement have been proposed (see e.g. [DGH82, STB03]). The goal of these methods is to be computationally cheap and to keep bounded the mesh elements aspect ratio even under large displacements.

In this work we frequently group the fluid and geometry problems, in which case we note the *Fluid–Geometry* problem FG:

$$\mathsf{FG}(\mathbf{u}_f, \widehat{\mathbf{d}}_s) = 0.$$

Notations: *we call Γ_D^f and Γ_D^s the Dirichlet boundaries (without the interface Γ) of Ω_t^f and $\widehat{\Omega}^s$ respectively, while the Neumann boundaries are noted Γ_N^f and Γ_N^s. Furthermore the Dirichlet boundary of $\widehat{\Omega}^f$ with respect to the mesh motion equation* (2.2.5)*, which is the fluid boundary where the domain motion is prescribed (e.g. when a portion of the fluid domain boundary is fixed), will be noted Γ_{Fixed}^f. The conditions imposed on the boundaries are problem dependent. We denote with $\mathbf{g}_f^D : \Gamma_D^f \rightarrow \mathbb{R}^3$, $\mathbf{g}_s^D : \Gamma_D^s \rightarrow \mathbb{R}^3$, $\mathbf{g}_m^D : \Gamma_{Fixed}^f \rightarrow \mathbb{R}^3$ the Dirichlet boundary data for the fluid, solid and geometry problems respectively, while $\mathbf{g}_f^N : \Gamma_N^f \rightarrow \mathbb{R}^3$ and $\mathbf{g}_s^N \in \Gamma_N^s \rightarrow \mathbb{R}^3$ represent the Neumann boundary data for the fluid and the solid problems.*

2.3 Time Discretization for the Structure Problem

The structure model is described by a PDE which is second order in time. Suitable and popular methods available in literature for the time discretization of such problem are the mid point scheme, Newmark (see [Nob01, Ch.4.8]) or the generalized-α scheme [RSFW09, DP07, BCHZ08]. The latter is a generalization of the Newmark scheme, while the mid point scheme can be viewed as a Newmark scheme with a particular choice of the coefficients [Nob01, Ch.4.8]. We report here the mid point scheme for a generic second-order ODE which can represent e.g. the momentum conservation for the solid problem (1.5.4)

$$\mathbf{M}\left(\frac{\partial^2 \widehat{\mathbf{d}}_s}{\partial t^2}, \frac{\partial \widehat{\mathbf{d}}_s}{\partial t}, \widehat{\mathbf{d}}_s\right) = 0. \tag{2.3.1}$$

The mid point scheme reads

$$\begin{cases} \dfrac{\widehat{\mathbf{d}}_s^{n+1} - \widehat{\mathbf{d}}_s^{n}}{\delta t} = \dfrac{\dot{\widehat{\mathbf{d}}}_s^{n+1} + \dot{\widehat{\mathbf{d}}}_s^{n}}{2} \\ \mathbf{M}\left(\dfrac{\dot{\widehat{\mathbf{d}}}_s^{n+1} - \dot{\widehat{\mathbf{d}}}_s^{n}}{\delta t}, \dfrac{\dot{\widehat{\mathbf{d}}}_s^{n+1} + \dot{\widehat{\mathbf{d}}}_s^{n}}{2}, \dfrac{\widehat{\mathbf{d}}_s^{n+1} + \widehat{\mathbf{d}}_s^{n}}{2}\right) = 0 \end{cases} \tag{2.3.2}$$

These are two equations in the unknowns $\widehat{\mathbf{d}}_s^{n+1}$ and $\dot{\widehat{\mathbf{d}}}_s^{n+1}$ to be solved at each time step.

We report below a discretization of (2.3.1) with the generalized-α method:

$$\begin{aligned} 0 &= \mathbf{M}(\ddot{\widehat{\mathbf{d}}}_s^{n+\alpha_m}, \dot{\widehat{\mathbf{d}}}_s^{n+\alpha_f}, \widehat{\mathbf{d}}_s^{n+\alpha_f}) \\ \widehat{\mathbf{d}}_s^{n+\alpha_f} &= \widehat{\mathbf{d}}_s^{n} + \alpha_f(\widehat{\mathbf{d}}_s^{n+1} - \widehat{\mathbf{d}}_s^{n}) \\ \dot{\widehat{\mathbf{d}}}_s^{n+\alpha_f} &= \dot{\widehat{\mathbf{d}}}_s^{n} + \alpha_f(\dot{\widehat{\mathbf{d}}}_s^{n+1} - \dot{\widehat{\mathbf{d}}}_s^{n}) \\ \ddot{\widehat{\mathbf{d}}}_s^{n+\alpha_f} &= \ddot{\widehat{\mathbf{d}}}_s^{n} + \alpha_m(\ddot{\widehat{\mathbf{d}}}_s^{n+1} - \ddot{\widehat{\mathbf{d}}}_s^{n}) \\ \widehat{\mathbf{d}}_s^{n+1} &= \widehat{\mathbf{d}}_s^{n} + \delta t \dot{\widehat{\mathbf{d}}}_s^{n} + \frac{(\delta t)^2}{2}\left[(1-2\beta)\ddot{\widehat{\mathbf{d}}}_s^{n} + 2\beta \ddot{\widehat{\mathbf{d}}}_s^{n+1}\right] \qquad (2.3.3) \\ \dot{\widehat{\mathbf{d}}}_s^{n+1} &= \dot{\widehat{\mathbf{d}}}_s^{n} + \delta t[(1-\gamma)\ddot{\widehat{\mathbf{d}}}_s^{n} + \gamma \ddot{\widehat{\mathbf{d}}}_s^{n+1}], \qquad (2.3.4) \end{aligned}$$

where α_f, α_m, β, γ, are arbitrary real parameters defining the method, while δt is the time step. It is a system of 6 equations in the unknowns $\widehat{\mathbf{d}}_s^{n+1}, \dot{\widehat{\mathbf{d}}}_s^{n+1}, \ddot{\widehat{\mathbf{d}}}_s^{n+1}, \widehat{\mathbf{d}}_s^{n+\alpha_f}, \dot{\widehat{\mathbf{d}}}_s^{n+\alpha_f}, \ddot{\widehat{\mathbf{d}}}_s^{n+\alpha_f}$. With $\alpha_f = \alpha_m = 1$ we have the Newmark scheme, and if also $\gamma = \frac{1}{2}$ and $\beta = \frac{1}{4}$ we obtain the mid point scheme.

Remark 2.3.1. *To obtain the mid point scheme from the generalized-α it is sufficient to substitute the coefficients α_f, α_m, β, and γ in* (2.3.4)

$$\delta t\left[\frac{1}{2}\ddot{\widehat{\mathbf{d}}}_s^{n} + \frac{1}{2}\ddot{\widehat{\mathbf{d}}}_s^{n+1}\right] = \dot{\widehat{\mathbf{d}}}_s^{n+1} - \dot{\widehat{\mathbf{d}}}_s^{n},$$

then substituting in (2.3.3) *we obtain directly*

$$\frac{\widehat{\mathbf{d}}_s^{n+1} - \widehat{\mathbf{d}}_s^{n}}{\delta t} = \frac{\dot{\widehat{\mathbf{d}}}_s^{n+1} + \dot{\widehat{\mathbf{d}}}_s^{n}}{2},$$

which is the first equation in (2.3.2).

Chung and Hulbert [CH93] showed that second order accuracy is attained if

$$\gamma = \frac{1}{2} - \alpha_f + \alpha_m,$$

and

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

The scheme is unconditionally stable if $\alpha_m \geq \alpha_f \geq \frac{1}{2}$. Furthermore in [CH93] optimal choices for the parameters α_f and α_m are retrieved:

$$\alpha_m^c = \frac{2 - \rho_\infty}{1 + \rho_\infty},$$

$$\alpha_f^c = \frac{1}{1 + \rho_\infty},$$

where the parameter ρ_∞ is the limit spectral radius of the amplification matrix for infinite time steps. If $\rho_\infty = 1$ then $\alpha_f = \alpha_m = \frac{1}{2}$ and there is no damping of the high frequency components.

A similar set of coefficients was devised by Jansen et al. in [JWH00] for the first order equations. We report for completeness the corresponding values of α_f and α_m in this case

$$\alpha_m^c = \frac{1}{2}\frac{3 - \rho_\infty}{1 + \rho_\infty},$$

$$\alpha_f^c = \frac{1}{1 + \rho_\infty}.$$

Notice that in [DP07, BCHZ08] the generalized-α scheme is used for both the fluid and structure fields.

Also the structure equation can be nonlinear, the only nonlinearity coming from the constitutive law relating the stress and the strain tensors. As in the fluid–geometry case there is the option of considering part (or all) of the nonlinear term explicitly, or to linearize using Newton or fixed-point. The discussion about the last two strategies is reported in the next chapter.

The time advancing scheme used in Part III for the numerical simulations is the mid point rule (2.3.2).

2.4 Time Discretization for the Fluid–Geometry Problem

A large variety of time discretizations of the incompressible Navier–Stokes equations on moving domains can be found in literature, depending on the targeted application. However using the ALE formulation imposes some constraints. In fact the choice of the time discretization of the convective term $(\boldsymbol{\beta} \cdot \nabla_x)\mathbf{u}$ plays a fundamental role in determining the order of the time advancing scheme and its stability properties (through the *Geometric Conservation Law*, cf. Section §2.8). For instance one may chose a time advancing scheme of second order and obtain only a first order approximation in time, see e.g. [FvdZG06]. The stability issue is discussed more in detail in Section §2.8, while here we just describe some of the possible choices adopted in literature.

The approaches more frequently used in hemodynamics are one-step implicit methods like the implicit Euler scheme (first order accurate), the generalized alpha scheme (first or second order accurate), Crank–Nicholson (second order accurate) or sometimes multistep BDF schemes (see [QSS00]).

The implicit Euler scheme consists in considering all the quantities at the current time step. For instance the discretization of the generic non conservative form of an ALE field reported in equation (1.3.9), i.e.,

$$\left.\frac{\partial \boldsymbol{\alpha}}{\partial t}\right|_{\widetilde{x}} = \nabla_x\cdot(\Theta - \boldsymbol{\alpha}\otimes\mathbf{u}) + (\mathbf{w}\cdot\nabla_x)\boldsymbol{\alpha} + \mathbf{b},$$

reads:

$$\frac{\boldsymbol{\alpha}^{n+1}}{\delta t} - \nabla_x\cdot(\Theta^{n+1} - \boldsymbol{\alpha}^{n+1}\otimes\mathbf{u}^{n+1}) - \mathbf{w}^\star\nabla_x\boldsymbol{\alpha}^{n+1} = \frac{\boldsymbol{\alpha}^n}{\delta t} + \mathbf{b}^{n+1}, \tag{2.4.1}$$

where $\boldsymbol{\alpha}$ and Θ are a vectorial and tensorial quantities introduced in (1.3.9).

The approximation of the domain velocity $\mathbf{w}^\star$ is independent of the time approximation chosen for the equation, indeed it defines how the coupling between the problems F and G is handled. If these two problems are not strongly coupled the approximation of the fluid domain velocity $\mathbf{w}^\star$ affects the order of the time discretization.

The generalized-α scheme was first introduced for structural dynamic applications in [CH93] for second order ODEs. It was then adapted to first order ODEs in [JWH00]. In FSI context the generalized-α has been used e.g. in [DP07, BCHZ08]. The Crank–Nicholson scheme applied to the fluid problem in FSI has been used in e.g. [HT06, Nob01], while in [RSFW09, FvdZG06] the BDF2 scheme is used. All these choices can lead to a globally second order in time FSI scheme, when the structure solver and coupling conditions are also second order accurate. The first order time discretizations (e.g. implicit Euler) for the FG problem are more common, however we anticipate that using the conservative formulation of the momentum equation, implicit Euler time discretization and choosing $\mathbf{w}^\star = \mathbf{w}^{n+1}$ an important condition (the DGCL condition, cf. Section §2.8) is not fulfilled, which can lead to the instability of the scheme [Nob01, Ch.4.7].

We remark also that the applications can dictate constraints on the timestep (e.g. when simulating turbulent fluids through DNS), in these cases an explicit time discretization can be employed for the fluid problem (see e.g. [BA09, FL00]).

With the aim of keeping a general notation that applies to every time discretization, as well as to every form of the conservation equations and interface conditions, in what follows we denote the time-discretized fluid and solid problems by F^n and S^n respectively.

The coupled fluid–structure problem has several levels of nonlinearities, some of them being related to the fact that we chose to write the equations in ALE form:

- the convective term of the Navier–Stokes equations in fixed domains

$$(\mathbf{u}\cdot\nabla_x)\mathbf{u}; \tag{2.4.2}$$

- the advection term introduced by the ALE formulation

$$(\mathbf{w}\cdot\nabla_x)\mathbf{u}; \tag{2.4.3}$$

- the dependence of the fluid domain displacement $\widehat{\mathbf{d}}_f$ on the displacement of the structure.

At every time step they can be solved by using a Newton or fixed point algorithm, as it is discussed in the next chapter. By properly choosing the time discretization the FSI problem at each time step may result to be linear.

We call *Fully Implicit* (FI) time discretization (see e.g. [BCHZ08, TSS06, HHB08, KGF$^+$09, BC10a, DP07, Hro01, MNS06, GV03, LM01, KW08b]) the implicit treatment of all the nonlinearities introduced above as well as of the coupling with the solid problem. It leads in general to a more accurate and robust time discretization, but it is also the most expensive approach among those discussed here, since the solution of the nonlinearities needs extra computational efforts.

Different approaches can be obtained by modifying the way the nonlinear terms are discretized in the FI approach. For instance being $n+1$ the current time step, the nonlinearity (2.4.2) can be discretized in time, given an extrapolation $\mathbf{u}^\star$ of the fluid velocity from the previous time steps, in the following way

$$(\mathbf{u}^\star \cdot \nabla_x)\mathbf{u}^{n+1}. \tag{2.4.4}$$

With this choice the term (2.4.4) becomes linear. This is a suitable choice when the characteristic Reynolds number of the fluid flow is not very high, and thus the nonlinearity of the convective term is not predominant (condition usually fulfilled in hemodynamics). The simplest approximation corresponds to choosing $\mathbf{u}^\star = \mathbf{u}^n$. It introduces an error of the first order in time and can be easily modified if a time advancing scheme of higher order is used. This approximation for the convective term of the fluid problem is used e.g. in [FM05, DDFQ06, Nob01, BQQ08b, Dep04].

Also the nonlinearity (2.4.3) can be discretized in time as

$$(\mathbf{w}^\star \cdot \nabla_x)\mathbf{u}^{n+1}, \tag{2.4.5}$$

where $\mathbf{w}^\star$ is an explicit extrapolation of the fluid domain velocity. Together with the term (2.4.4) it leads to a linearized convective term [CDFQ11].

The nonlinearity due to the moving domain is the most difficult to handle implicitly (cf. Chapter 3). Considering the fluid domain at the previous time step is a suitable approximation when the displacements are not very large. Its explicit treatment, combined with (2.4.4) and (2.4.5), leads to the *Geometry–Convective Explicit* (GCE) time discretization (see e.g. [CDFQ11, BQQ08a]), in which the fluid problem is linear. In Tezduyar et al., see e.g. [TSS06], some different coupling strategies are also investigated. In particular the quasi–direct coupling in their convention corresponds to a fully coupled scheme where the geometry is considered explicitly, while the direct coupling corresponds to the fully implicit discretization introduced above.

Besides the time discretization of the nonlinear terms, frequently the methods differ with respect to the time discretization of the coupling between fluid and structure. However, since this subject has been extensively studied and since it is critical for the stability of the method, we chose to leave it for Section §2.8, where the stability issue is addressed.

2.5 Fully Implicit and Convective Explicit Schemes

Using the fully implicit approach for the fluid–geometry problem, the coupled time-discrete system of equations at the time level $t = t_{n+1}$ reads

$$\begin{aligned} \mathsf{F}^n({\mathbf{u}_f}^{n+1}, \widehat{\mathbf{d}}_s^{n+1}, \widehat{\mathbf{d}}_f^{n+1}) &= 0 \\ \mathsf{S}^n({\mathbf{u}_f}^{n+1}, \widehat{\mathbf{d}}_s^{n+1}) &= 0 \\ \mathsf{G}^n(\widehat{\mathbf{d}}_s^{n+1}, \widehat{\mathbf{d}}_f^{n+1}) &= 0 \end{aligned} \tag{2.5.1}$$

We denote the first order discrete time derivative for the fluid problem as δ_t, while the discretization of the second derivative in time of the structure equation is represented by δ_{tt}. The whole system of equations in strong form, considering the fluid momentum conservation equation in non-conservative form, reads

$$\begin{aligned} \rho_f \delta_t \mathbf{u}^{n+1} + \rho_f \left((\mathbf{u}^{n+1} - \mathbf{w}^{n+1}) \cdot \nabla_x \right) \mathbf{u}^{n+1} - \nabla_x \cdot {\sigma_f}^{n+1} - {\mathbf{f}_f}^{n+1} &= 0 && \text{in } {\Omega_t^f}_{n+1} \\ \nabla_x \cdot \mathbf{u}^{n+1} &= 0 && \text{in } {\Omega_t^f}_{n+1} \\ \widehat{\rho}_s \delta_{tt} \widehat{\mathbf{d}}_s^{n+1} - \nabla_{\widehat{x}} \cdot \sigma_{os}^{n+1} - {\mathbf{f}_s}^{n+1} &= 0 && \text{in } \widehat{\Omega}^s \\ -\Delta \widehat{\mathbf{d}}_f^{n+1} &= 0 && \text{in } \widehat{\Omega}^f \\ \mathbf{\Pi}^{n+1} \widehat{\mathbf{n}}^s + \widehat{\sigma}_f^{n+1} \cdot \widehat{\mathbf{n}}^f &= 0 && \text{on } \widehat{\Gamma} \\ \mathbf{u}^{n+1} \circ \mathcal{A}_{t_{n+1}} - \delta_t \widehat{\mathbf{d}}_s^{n+1} &= 0 && \text{on } \widehat{\Gamma} \\ \widehat{\mathbf{d}}_f^{n+1} - \widehat{\mathbf{d}}_s^{n+1} &= 0 && \text{on } \widehat{\Gamma} \end{aligned} \tag{2.5.2}$$

+conditions on the external boundaries.

The solution of the geometry problem defines the ALE map $\mathcal{A}_{t_{n+1}}$ and the fluid domain at time t_{n+1}, ${\Omega_t^f}_{n+1} = \mathcal{A}_{t_{n+1}}(\widehat{\Omega}^f)$. Given (at least) $\widehat{\mathbf{d}}_f^{n+1}$ and $\widehat{\mathbf{d}}_f^n$ the fluid domain velocity can be computed as

$$\mathbf{w}^{n+1} = \delta_t \widehat{\mathbf{d}}_f^{n+1} \circ \mathcal{A}_{t_{n+1}}^{-1}. \tag{2.5.3}$$

When the convective term in the fluid momentum equation is discretized using both (2.4.4) and (2.4.5), while the fluid geometry is still considered implicitly, we call the corresponding time discretization of the fluid–geometry problem *Convective Explicit* (CE).

2.6 Geometry–Convective Explicit Scheme

In the *Geometry–Convective Explicit* (GCE) time discretization the nonlinear term (2.4.3) is linearized using (2.4.5) and the coupling of the fluid and the geometry problems is explicit. The GCE counterpart of (2.5.1) reads

$$\begin{aligned} \mathsf{F}^n({\mathbf{u}_f}^{n+1}, \widehat{\mathbf{d}}_s^{n+1}, \widehat{\mathbf{d}}_f^{n+1}) &= 0 \\ \mathsf{S}^n({\mathbf{u}_f}^{n+1}, \widehat{\mathbf{d}}_s^{n+1}) &= 0 \\ \mathsf{G}^n(\widehat{\mathbf{d}}_s^{n}, \widehat{\mathbf{d}}_f^{n+1}) &= 0. \end{aligned} \tag{2.6.1}$$

Although the convective term is linearized using an explicit extrapolation of the fluid velocity $\mathbf{u}^\star$, with an abuse of notation we use the same symbol for the fluid problem F^n as in the fully implicit case.

We write explicitly below the time discretized equations in strong form defining the GCE scheme. The geometry equation,

$$\mathsf{G}^n(\widehat{\mathbf{d}}_s^n, \widehat{\mathbf{d}}_f^{n+1}) = 0,$$

is solved separately once per time step and the fluid domain velocity $\mathbf{w}^{n+1}$ is computed using (2.5.3).

The time discrete fluid–structure problem reads (the fluid momentum equation being written in non-conservative form):

$$\begin{aligned}
\rho_f \delta_t \mathbf{u}^{n+1} + \rho_f \left((\mathbf{u}^n - \mathbf{w}^{n+1}) \cdot \nabla_x\right)\mathbf{u}^{n+1} - \nabla_x \cdot {\sigma_f}^{n+1} &= {\mathbf{f}_f}^{n+1} && \text{in } {\Omega_t^f}_{n+1}, \\
\nabla_{\widehat{x}} \cdot \mathbf{u}^{n+1} &= 0 && \text{in } {\Omega_t^f}_{n+1}, \\
\widehat{\rho}_s \delta_{tt} \widehat{\mathbf{d}}_s^{n+1} - \nabla_{\widehat{x}} \cdot (\mathbf{\Pi})^{n+1} &= {\mathbf{f}_s}^{n+1} && \text{in } \widehat{\Omega}^s, \\
\mathbf{u}^{n+1} \circ \mathcal{A}_{t_{n+1}} - \delta_t \widehat{\mathbf{d}}_s^{n+1} &= 0 && \text{on } \widehat{\Gamma}, \\
\mathbf{\Pi}^{n+1} \widehat{\mathbf{n}}^s + (\widehat{\sigma}_f)^{n+1} \cdot \widehat{\mathbf{n}}^f &= 0 && \text{on } \widehat{\Gamma},
\end{aligned} \tag{2.6.2}$$

+conditions on the external boundaries.

As already mentioned this problem has the advantage of being linear when the structure equation is linear, at any given time step. In applications, as shown in Sections §5.4 and §6.2, this time discretization also leads to a stable and robust FSI solver and can be efficiently implemented in parallel.

2.7 Space Discretization

In this section we write the weak form and describe the space finite elements discretization of the FSI system. We refer to [LM01, Nob01] for details.

Recalling the notations introduced in Section §2.2 we can define the following functional spaces

$$\begin{aligned}
U^f &= \{\mathbf{v} = \hat{\mathbf{v}} \circ {\mathcal{A}_t}^{-1} | \hat{\mathbf{v}} \in H^1(\widehat{\Omega}^f)^3\}, \\
U^p &= \{q = \hat{q} \circ {\mathcal{A}_t}^{-1} | \hat{q} \in L^2(\widehat{\Omega}^f)\}, \\
U^s &= H^1(\widehat{\Omega}^s)^3, \\
U^g &= H^1(\widehat{\Omega}^f)^3, \\
U^{\widehat{\Gamma}} &= H^{1/2}(\widehat{\Gamma})^3, \\
V^f &= \{\mathbf{v} = \hat{\mathbf{v}} \circ {\mathcal{A}_t}^{-1} | \hat{\mathbf{v}} \in H^1(\widehat{\Omega}^f)^3,\ \hat{\mathbf{v}} = 0 \text{ on } \Gamma_D^f\}, \\
V^s &= \{\mathbf{v} \in H^1(\widehat{\Omega}^s)^3 |\ \mathbf{v} = 0 \text{ on } \Gamma_D^s\}, \\
V^g &= \{\mathbf{v} \in H^1(\widehat{\Omega}^f)^3 | \mathbf{v} = 0 \text{ on } \Gamma_{Fixed}^f\}.
\end{aligned}$$

If Ω_t^f and $\widehat{\Omega}^f$ are bounded and with Lipschitz continuous boundary, $\mathcal{A}_t \in W^{1,\infty}(\widehat{\Omega}^f)^3$ and ${\mathcal{A}_t}^{-1} \in W^{1,\infty}(\Omega_t^f)^3$, then $U^f \equiv H^1(\Omega_t^f)^3$. Furthermore, if also $\mathcal{A} \in H^1(T \subset \mathbb{R}; W^{1,\infty}(\widehat{\Omega}^f)^3)$ holds, then, for $\widehat{\mathbf{u}} \in H^1(T; H^1(\widehat{\Omega}^f)^3)$, $\mathbf{u} = \widehat{\mathbf{u}} \circ \mathcal{A}^{-1} \in H^1(T; U^f) \equiv H^1(T; H^1(\Omega_t^f)^3)$ [Nob01, Ch.1.3].

We report hereafter the weak formulation of the FSI system in non-conservative and conservative form. We first introduce the way in which the stress continuity coupling condition

is enforced (in a weak sense), and show that enforcing this coupling condition amounts to choosing matching test functions across the interface.

While the velocity continuity coupling condition is usually imposed in its strong form, the continuity of stresses through the interface can be imposed in a weak way by equating the variational residuals of the two momentum conservation equations restricted to the interface. In fact if we write the variational formulation of the fluid momentum equation, considering an arbitrary $\mathbf{v}_f \in V^f$, we have

$$\int_{\Omega_t^f} \left(\rho_f \left. \frac{\partial \mathbf{u}}{\partial t} \right|_{\widetilde{x}} \cdot \mathbf{v}_f + (\rho_f(\mathbf{u}-\mathbf{w})\cdot\nabla_x)\mathbf{u}\cdot\mathbf{v}_f + \sigma_f : \nabla_x \mathbf{v}_f - \mathbf{f}_f \cdot \mathbf{v}_f \right) \, d\Omega_t{}^f + \\ - \int_{\Gamma_N^f} \mathbf{g}_f^N \cdot \mathrm{tr}_\Gamma(\mathbf{v}_f)\, d\gamma = \int_\Gamma \sigma_f \mathbf{n}^f \cdot \mathrm{tr}_\Gamma(\mathbf{v}_f)\, d\gamma. \quad (2.7.1)$$

The same can be done for the structure equation for $\mathbf{v}_s \in V^s$, leading to

$$\int_{\widehat{\Omega}^s} \left(\widehat{\rho}_s \frac{\partial^2 \mathbf{d}_s}{\partial t^2} \cdot \mathbf{v}_s - (\nabla_{\widehat{x}} \cdot \mathbf{\Pi})\mathbf{v}_s - \mathbf{f}_s \mathbf{v}_s \right) \, d\widehat{\Omega}^s - \int_{\Gamma_N^s} \mathbf{g}_s^N \cdot \mathrm{tr}_{\widehat{\Gamma}}(\mathbf{v}_s)\, d\gamma = \int_{\widehat{\Gamma}} \mathbf{\Pi}\widehat{\mathbf{n}}^s \cdot \mathrm{tr}_{\widehat{\Gamma}}(\mathbf{v}_s)\, d\widehat{\gamma}. \quad (2.7.2)$$

We introduce now the two linear continuous lift operators

$$\begin{aligned} \mathcal{L}_f : \; U^{\widehat{\Gamma}} &\rightarrow V^f, \\ \mathcal{L}_s : \; U^{\widehat{\Gamma}} &\rightarrow V^s. \end{aligned}$$

Expressing the stress tensors in the reference configuration, we can write the stress continuity equation on the interface in a weak form,

$$\int_{\widehat{\Gamma}} \mathbf{\Pi}\widehat{\mathbf{n}}^s \cdot \psi \, d\widehat{\gamma} + \int_{\widehat{\Gamma}} \widehat{\sigma}_f \widehat{\mathbf{n}}^f \cdot \psi \, d\widehat{\gamma} = 0 \quad \forall \psi \in U^{\widehat{\Gamma}}, \quad (2.7.3)$$

as

$$\int_{\widehat{\Omega}^s} \left(\widehat{\rho}_s \frac{\partial^2 \mathbf{d}_s}{\partial t^2} \cdot \mathcal{L}_s(\psi) \right) \, d\widehat{\Omega} + \int_{\widehat{\Omega}^s} \left(\mathbf{\Pi} : \nabla_{\widehat{x}} \mathcal{L}_s(\psi) - \mathbf{f}_s \cdot \mathcal{L}_s(\psi) \right) \, d\widehat{\Omega} - \int_{\Gamma_N^s} \mathbf{g}_s^N \cdot \mathcal{L}_s(\psi)\, d\gamma \\ + \int_{\widehat{\Omega}^f} \left(\rho_f \left. \frac{\partial \mathbf{u} \circ \mathcal{A}_t{}^{-1}}{\partial t} \right|_{\widetilde{x}} \cdot \mathcal{L}_f(\psi) + \rho_f \left((\mathbf{u} \circ \mathcal{A}_t{}^{-1} - \mathbf{w} \circ \mathcal{A}_t{}^{-1}) \cdot \nabla_{\widehat{x}} \right) \mathbf{u} \circ \mathcal{A}_t{}^{-1} \cdot \mathcal{L}_f(\psi) + \right. \\ \left. + \widehat{\sigma}_f : \nabla_{\widehat{x}} \cdot \mathcal{L}_f(\psi) - \widehat{\mathbf{f}}_f \cdot \mathcal{L}_f(\psi) \right) \, d\widehat{\Omega} - \int_{\Gamma_N^f} \mathbf{g}_f^N \cdot \mathcal{L}_f(\psi)\, d\gamma = 0 \quad \forall \psi \in U^{\widehat{\Gamma}}. \quad (2.7.4)$$

Thus the weak imposition of the stress continuity can be achieved by simply choosing test functions $\mathbf{v}_f \in V^f$ and $\mathbf{v}_s \in V^s$ which are matching on the FS interface. The weak imposition of the stresses leads sometimes to more stable numerical results [FLT98, LM01].

Following [LM01] and [Nob01] we consider a global weak formulation of the FSI problem. To this end we define the space $W = \{(\mathbf{v}_f, \mathbf{v}_s) \in V^f \times V^s | \mathrm{tr}_{\widehat{\Gamma}}(\mathbf{v}_f \circ \mathcal{A}_t) = \mathrm{tr}_{\widehat{\Gamma}}(\mathbf{v}_s)\}$. Imposing the weak stress continuity coupling condition, as previously stated, is obtained automatically by choosing the test functions in this space, i.e, supposing that the test functions restricted to the interface are matching.

We recall the notation for the Dirichlet boundary data for the three problems: $\mathbf{g}_f^D : \Gamma_D^f \to \mathbb{R}^2$, $\mathbf{g}_s^D : \Gamma_D^s \to \mathbb{R}^2$, $\mathbf{g}_m^D : \Gamma_{Fixed}^f \to \mathbb{R}^2$. The weak form of the equations (1.4.6), (1.4.2), (1.5.4) and (2.2.5) reads: for almost every $t \in T$ find $\mathbf{u} : T \to U^f$ such that $t \mapsto \mathbf{u}(t)$ and $\mathbf{u} = \mathbf{g}_f^D$ on Γ_D^f , $p : T \to U^p$ such that $t \mapsto p(t)$, $\widehat{\mathbf{d}}_s : T \to U^s$ such that $t \mapsto \widehat{\mathbf{d}}_s(t)$ and $\widehat{\mathbf{d}}_s = \mathbf{g}_s^D$ on Γ_D^s, $\widehat{\mathbf{d}}_f : T \to U^g$ such that $t \mapsto \widehat{\mathbf{d}}_f(t)$ and $\widehat{\mathbf{d}}_f = \mathbf{g}_m^D$ on Γ_{Fixed}^f satisfying

$$\begin{aligned}
&\int_{\Omega_t^f} \rho_f \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\widetilde{x}} \cdot \mathbf{v}_f + (\rho_f(\mathbf{u}-\mathbf{w})\cdot\nabla_x)\mathbf{u}\cdot\mathbf{v}_f + \sigma_f : \nabla_x \mathbf{v}_f \; d\Omega_t{}^f + \\
&+ \int_{\widehat{\Omega}^s} \widehat{\rho}_s \frac{\partial^2 \widehat{\mathbf{d}}_s}{\partial t^2}\cdot\mathbf{v}_s + \mathbf{\Pi} : \nabla_x \mathbf{v}_s \; d\widehat{\Omega}^s = \int_{\widehat{\Omega}^s} \mathbf{f}_s \cdot \mathbf{v}_s \; d\widehat{\Omega}^s - \int_{\Gamma_N^s} \mathbf{g}_s^N \cdot \mathbf{v}_s \, d\widehat{\gamma} + \\
&\qquad + \int_{\Omega_t^f} \mathbf{f}_f \cdot \mathbf{v}_f \; d\Omega_t{}^f - \int_{\Gamma_N^f} \mathbf{g}_f^N \cdot \mathbf{v}_f \, d\gamma, \qquad \forall (\mathbf{v}_f, \mathbf{v}_s) \in W,
\end{aligned} \tag{2.7.5}$$

$$\int_{\Omega_t^f} \mathbf{u}\cdot\nabla_x q \; d\Omega_t{}^f = 0 \qquad \forall q \in U^p,$$

$$\int_{\widehat{\Omega}^f} \nabla_{\widehat{x}} \widehat{\mathbf{d}}_f : \nabla_{\widehat{x}} \mathbf{v}_m \; d\widehat{\Omega}^f = 0 \qquad \forall \mathbf{v}_m \in V^g.$$

$$\mathbf{u} \circ \mathcal{A}_t = \frac{\partial \widehat{\mathbf{d}}_s}{\partial t} \qquad on \;\; \widehat{\Gamma}$$

$$\widehat{\mathbf{d}}_f = \widehat{\mathbf{d}}_s \qquad on \;\; \widehat{\Gamma}.$$

To write the conservative form it is sufficient to replace the fluid momentum contribution in (2.7.5) with

$$D_t \int_{\Omega_t^f} \rho_f \mathbf{u}\cdot\mathbf{v}_f \, d\Omega_t + \int_{\Omega_t^f} \left[(\sigma_f - \rho_f \mathbf{u}\otimes\boldsymbol{\beta}) : \nabla_x \mathbf{v}_f - \mathbf{f}_f \cdot \mathbf{v}_f\right] \; d\Omega_t{}^f - \int_{\Gamma_N^f} \sigma_f \mathbf{n}^f \cdot \mathbf{v}_f \, d\gamma = 0. \tag{2.7.6}$$

The Dirichlet boundary conditions are usually enforced *strongly*, i.e., discretizing directly equation (2.1.1), while the Neumann boundary conditions are usually imposed *weakly*, i.e., given the boundary data $\mathbf{g}_f^N \in (H^{\frac{1}{2}}(\Gamma_N^f))^3$ and $\mathbf{g}_s^N \in (H^{\frac{1}{2}}(\Gamma_N^s))^3$ enforcing the Neumann boundary conditions is obtained by substituting the boundary terms in (2.7.5).

Remark 2.7.1. *If the geometric adherence coupling condition holds, then, on the FS interface, the ALE map equals the deformation map ϕ_t, and the fluid domain velocity equals the fluid velocity $\mathbf{w} = \mathbf{u}$. Since the description of the boundary for both the fluid and solid problems is Lagrangian no interpolation is required when enforcing the coupling (if the finite elements are conforming across the interface).*

Remark 2.7.2. *The test functions in V^f are piecewise constant in time, so that*

$$\left.\frac{\partial \mathbf{v}_f}{\partial t}\right|_{\widetilde{x}} = 0 \; \forall \mathbf{v}_f \in V^f. \tag{2.7.7}$$

Due to (2.7.7) *the test functions $\mathbf{v}_f$ can always be taken out of the ALE derivatives. In particular this implies the following equality*

$$D_t \int_{\Omega_t^f} \mathbf{u}\cdot\mathbf{v}_f \; d\Omega_t{}^f = \int_{\Omega_t^f} \left.\frac{\partial \mathbf{u}}{\partial t}\right|_{\widetilde{x}} \cdot \mathbf{v}_f + \mathbf{u}(\nabla_x \cdot \mathbf{w})\cdot\mathbf{v}_f \; d\Omega_t{}^f. \tag{2.7.8}$$

The Finite Element (FE) space discretization is obtained from the variational formulation by approximating the functional spaces previously defined using finite dimensional spaces, that we denote with the label $_h$, spanned by a basis of *shape functions* (see e.g. [QV94]) defined in the reference domain, that we note $\{\psi_i\}_{1\leq i\leq N}$, where N is the dimension of the space U_h^f. For instance the fluid velocity $\mathbf{u}$ in the moving domain Ω_t^f is approximated as

$$\mathbf{u}(x,t) \approx \mathbf{u}_h(x,t) = \sum_{i=1}^{N} \mathbf{u}_i(t)\psi_i({\mathcal{A}_{h,t}}^{-1}(x)). \tag{2.7.9}$$

We remark that due to the linear combination above, the accuracy in space of the discrete solution of the fluid problem depends on the degree n of the approximation in space, the degree k of the FE parametric map and the degree l of the ALE approximation. To avoid a loss of accuracy the correct order for the ALE map should be the same as for the FE parametric map (see [Nob01, Ch.1.5] for a broader discussion). The differential problem for the ALE map should then be discretized with isoparametric finite elements of order k. In our case isoparametric linear finite elements are used for the discretization of all the equations, while for instance Bazilevs et al. [BCHZ08] use isoparametric NURBS (Non Uniform Rational B-Splines) shape functions for all the fields, which allows to describe the domain, the solution, and the ALE map as NURBS, and leads in principle to arbitrary order approximation in space and spectral convergence when increasing the degree of the rational polynomials and when the solution is analytic. A drawback of high order in space is that the polynomial order of (2.7.9) is the sum of the order of the shape functions ψ_i plus the order of the ALE map, which can lead to a large number of quadrature points necessary to integrate exactly the discrete solution. To avoid this waste of computational time in [PP10, PPQ10] a high order polynomial for the ALE map is used only for the elements touching the boundary, while a linear approximation is used on the other elements.

Notations: *we distinguish in the algebraic system between the degrees of freedom lying on the FS interface (adding the label* $_\Gamma$ *to the variables) and those which are internal to the domains* Ω_t^f *and* $\widehat{\Omega}^s$ *(for which we keep the same names as for the continuous case). The right-hand sides including the terms generated by the time discretization of the fluid and solid momentum conservation equations are denoted* $\mathbf{r}_f$ *and* $\mathbf{r}_s$ *respectively.*

We report hereafter the fully discrete nonlinear system in matrix form. Supposing that we chose a suitable FE discretization space for the three fields we can write the discrete system in a 4×4 block matrix form. Delimiting the blocks using straight lines the nonlinear system can be written in matrix form as

$$\left(\begin{array}{cc|cc|c|cc} C_{ff}(\diamond) & C_{f\Gamma}(\diamond) & 0 & 0 & 0 & 0 & 0 \\ C_{\Gamma f}(\diamond) & C_{\Gamma\Gamma}(\diamond) & 0 & 0 & I & 0 & 0 \\ \hline 0 & 0 & N_s(\diamond) & N_{s\Gamma}(\diamond) & 0 & 0 & 0 \\ 0 & 0 & N_{\Gamma s}(\diamond) & N_{\Gamma\Gamma}(\diamond) & -I & 0 & 0 \\ \hline 0 & I & 0 & -I/\delta t & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & H_{ff} & H_{f\Gamma} \\ 0 & 0 & 0 & -I & 0 & 0 & I \end{array}\right) \left(\begin{array}{c} {\mathbf{u}_f}^{n+1} \\ \hline {\mathbf{u}_{f\Gamma}}^{n+1} \\ \hline \widehat{\mathbf{d}}_s^{n+1} \\ \hline \widehat{\mathbf{d}}_{s\Gamma}^{n+1} \\ \hline \lambda^{n+1} \\ \hline \widehat{\mathbf{d}}_f^{n+1} \\ \widehat{\mathbf{d}}_{f\Gamma}^{n+1} \end{array}\right) = \left(\begin{array}{c} \mathbf{r}_f^{n+1}(\diamond) \\ \hline \mathbf{r}_{f\Gamma}^{n+1}(\diamond) \\ \hline \mathbf{r}_s^{n+1} \\ \hline \mathbf{r}_{s\Gamma}^{n+1} \\ \hline -I/\delta t\widehat{\mathbf{d}}_{s\Gamma}^{n} \\ \hline 0 \\ 0 \end{array}\right), \tag{2.7.10}$$

where the nonlinearity is expressed in the matrix entries with the symbol $(\diamond)$ (it expresses the dependence of the blocks on the solution variables). The $(1,1)$ block represents the

discrete fluid problem equations, the block $(2,2)$ represents the discrete solid problem, while block $(4,4)$ represents the discretization of the geometry problem. The off-diagonal entries enforce the coupling conditions between the three fields. In this discrete formulation we have supposed a first order time discretization of the velocity continuity coupling condition. The right hand sides $\mathbf{r}_f^{n+1}$ and $\mathbf{r}_s^{n+1}$ are composed by the volume forces and the terms of the time discretization which depend on the previous time steps. The multiplier λ is introduced to enforce the coupling conditions across the FS interface. When the coupling is enforced introducing a new variable we say that the linear system is written in *augmented form*. In fact from (2.7.4) we can write the weak stress continuity as

$$C_{\Gamma f}\mathbf{u}_f + C_{\Gamma\Gamma}\mathbf{u}_f + N_{\Gamma s}\widehat{\mathbf{d}}_s + N_{\Gamma\Gamma}\widehat{\mathbf{d}}_{s\Gamma} = 0.$$

We can rewrite the previous equation as

$$\begin{aligned} C_{\Gamma f}\mathbf{u}_f + C_{\Gamma\Gamma}\mathbf{u}_f &= \lambda, \\ N_{\Gamma s}\widehat{\mathbf{d}}_s + N_{\Gamma\Gamma}\widehat{\mathbf{d}}_{s\Gamma} + \lambda &= 0. \end{aligned} \tag{2.7.11}$$

Thus the third block-column of the matrix in (2.7.10) enforces the stress continuity coupling, while the third block-row represents the velocity continuity.

Note that we defined the multiplier λ at the algebraic level, which means that it consists of a vector with the dimension of the degrees of freedom sitting on the FS interface. However due to (2.7.11) it can be interpreted as the FE discretization of the integral of the surface traction on the interface: $\int_\Gamma \sigma_f \mathbf{n}^f \, d\gamma$.

The weak form of the GCE system discretized in space leads to the following linear systems: the mesh motion equation can be solved separately because in the geometry–explicit time discretization the fluid domain and the mesh velocity are extrapolated from the interface displacement of the previous time step. We have

$$\begin{pmatrix} H_{ff} & H_{f\Gamma} \\ 0 & I \end{pmatrix} \begin{pmatrix} \widehat{\mathbf{d}}_f^{n+1} \\ \widehat{\mathbf{d}}_{f\Gamma}^{n+1} \end{pmatrix} = \begin{pmatrix} 0 \\ \widehat{\mathbf{d}}_{s\Gamma}^{n} \end{pmatrix}.$$

The new domain Ω_t^f at time t_{n+1} is then recovered by means of the nodes of the mesh following $\widehat{\mathbf{d}}_f^{n+1}$. The integrals relative to the FE discretization of the fluid are computed with respect to the new mesh, yielding the following linear system:

$$\left(\begin{array}{cc|cc|c} C_{ff} & C_{f\Gamma} & 0 & 0 & 0 \\ C_{\Gamma f} & C_{\Gamma\Gamma} & 0 & 0 & I \\ \hline 0 & 0 & N_s & N_{s\Gamma} & 0 \\ 0 & 0 & N_{\Gamma s} & N_{\Gamma\Gamma} & -I \\ \hline 0 & I & 0 & -I/\delta t & 0 \end{array}\right) \left(\begin{array}{c} \mathbf{u}_f{}^{n+1} \\ \mathbf{u}_{f\Gamma}{}^{n+1} \\ \hline \widehat{\mathbf{d}}_s^{n+1} \\ \widehat{\mathbf{d}}_{s\Gamma}^{n+1} \\ \hline \lambda^{n+1} \end{array}\right) = \left(\begin{array}{c} \mathbf{r}_f^{n+1} \\ \mathbf{r}_{f\Gamma}^{n+1} \\ \hline \mathbf{r}_s^{n+1} \\ \mathbf{r}_{s\Gamma}^{n+1} \\ \hline -I/\delta t \mathbf{d}_{s\Gamma}^n \end{array}\right). \tag{2.7.12}$$

Since the GCE problem is linear, the symbol $(\diamond)$ is removed from the blocks in (2.7.10).

The general Jacobian system for CE or FI is reported in the next chapter, while the parallel solution and preconditioning of the various Jacobian systems is addressed in Chapter 4 in which the blocks structure of the matrix is exploited.

Notations: *in the following chapters we distinguish between three Jacobian matrices coming from three different choices for the time discretization: we call J_{FI} the Jacobian*

matrix of the system (2.7.10)*, J_{CE} the Jacobian of* (2.7.10) *when the convective term is linearized using* (2.4.4) *and* (2.4.5) *(but considering implicitly the geometric nonlinearity) and J_{GCE} the matrix in* (2.7.12)*.*

2.7.1 Interior Penalty Stabilization

For our simulations we discretize the Navier–Stokes equations by P1–P1 elements stabilized by interior penalty (IP) [BF07, BFH04, PB05].

The IP stabilization consists of adding to the Galerkin discrete formulation of the fluid problem some consistent terms that penalize the jumps of given quantities across the boundary Γ_e of each element. We denote with $[\![\cdot]\!]$ the jumps across the element boundaries; the penalization terms added to the fluid momentum equation are the following, for $\mathbf{v}_f \in V^f$:

- $j_\beta(\mathbf{u}, \mathbf{v}_f) = \int_{\Gamma_e} \gamma_\beta \frac{h^2}{\|\mathbf{u}-\mathbf{w}\|_{L^\infty}} [\![(\mathbf{u}-\mathbf{w})\cdot\nabla_x\mathbf{u}]\!][\![(\mathbf{u}-\mathbf{w})\cdot\nabla_x\mathbf{v}_f]\!]\, d\gamma$, where γ_β is a constant to be chosen. This term penalizes the jump of the convective term across the interface of the finite elements, and it is used to stabilize convection dominated flows.
- $j_{div}(\mathbf{u}, \mathbf{v}_f) = \int_{\Gamma_e} \gamma_{div} h^2 \|\mathbf{u}-\mathbf{w}\|_{L^\infty} [\![\nabla_x\mathbf{u}]\!][\![\nabla_x\mathbf{v}_f]\!]\, d\gamma$, where γ_{div} is a constant. This term penalizes the jumps of the velocity gradient.

Another term to stabilize the pressure is added to the continuity equation in weak form. This penalization term allows to circumvent the inf–sup condition and to use equal order finite elements for velocity and pressure fields. For $q \in U^p$, and a constant parameter γ_p,

$$j_p(p,q) = \int_{\Gamma_e} \gamma_p \frac{h^3}{\max\{h\|\mathbf{u}-\mathbf{w}\|_{L^\infty}, \mu_f\}} [\![\nabla_x p]\!][\![\nabla_x q]\!]\, d\gamma.$$

This term penalizes the jump of the pressure gradient across the elements and may lead to an inconsistency when the exact solution features a discontinuous pressure gradient.

Note that although these terms are converging to zero with the grid size h, and they are identically null in the continuous case, they are in general different from zero when discretized in space. This implies in particular that the discrete continuity equation is not satisfied exactly.

2.8 Analysis of the Coupled Problem

In this section we report results about the well posedness of the FSI problem and we discuss the stability of some of the different time discretization schemes introduced above. We introduce the concept of geometric conservation laws and the associated conditions for a numerical scheme. To conclude the section the stability of the coupling between the fluid–geometry and structure problems is investigated.

2.8.1 Analysis of Simplified Coupled Models

We refer in this section mainly to Y. Maday [FQV09, Ch.8], where it is reported an overview of recent results and methodologies to prove the well posedness of the FSI system.

In the usual definitions of stability, a system is stable when its energy is bounded by a constant which depends on the forcing terms and on the boundary conditions. A priori energy estimates are fundamental to prove the stability of a differential problem, just as they are

important for the stability of a numerical method. A classical Faedo–Galerkin approach used e.g. in [Lio69] allows to prove the existence of a weak solution in the FSI context when the fluid domain motion is neglected and the structure is visco-elastic. The solution is proved to be unique in the 2D case (see [Lio69, FQV09] for the proofs).

Starting from the variational formulation, the Faedo–Galerkin technique consists in defining a sequence of finite dimensional test spaces converging to the original infinite dimensional space. Then the existence and uniqueness of the solution for the finite dimensional problems is proved. Thanks to proper a priori estimates for all the terms in the variational formulation, there exists a subsequence for each term which converges weakly in the infinite dimensional space. For the FSI case considered in [Lio69] a compactness theorem is needed for the nonlinear convective term to pass to the limit.

An analysis by Chambolle et al. [CDEG05, FQV09] generalizes the previous strategy to FSI with moving fluid domain. The fact that the domain motion depends on the solution adds a nonlinearity which is a major complication for the analysis. In that work the solid is modeled as a visco-elastic plate. We give here an idea of the method used for the analysis.

Definition 2.8.1 (Compact operator)**.** *Let X and Y be two Banach spaces, and $T : X \to Y$ an operator. T is* compact *iff*

- *it is continuous;*
- *it maps bounded sets into relatively compact sets.*

The *Schauder Fixed point theorem* which can be found e.g. in [Zei91], is used to complete the proof of the existence.

Theorem 2.8.1 (Schauder Fixed-Point theorem)**.** *Let M be a nonempty, closed, bounded, convex subset of a Banach space X, and suppose that $T : M \to M$ is a compact operator. Then T has a fixed point.*

This theorem is frequently used to prove the existence of solutions for nonlinear equations in a Banach space. The proof of the existence of a weak solution for the FSI problem follows the main steps listed below.

The fluid and solid problems are substituted by two regularized problems, such that the solution is defined on a regularized current domain depending on a parameter ϵ.

The equations are linearized through fixed point, considering known the fluid domain displacement and the fluid velocity in the convective term. Given the solution $(\mathbf{u}^o, \widehat{\mathbf{d}}_s{}^o)$, we call $\widetilde{\mathcal{S}}(\mathbf{u}^o, \widehat{\mathbf{d}}_s{}^o)$ the solution of the linearized system.

The linearized equations are recast back to the original configuration. A basis of eigenfunctions for the Stokes problem is then used to generate a sequence of finite dimensional test spaces, and the Faedo–Galerkin technique is used, with steps analogous to those in [Lio69] recalled above. Using the a-priori estimates it is possible to conclude that the linearized continuous problem has a unique solution.

Let us consider the image of the solver operator $\widetilde{\mathcal{S}}$ of the linearized problem. Due to the estimates and to classical theorems (Aubin's lemma), this operator is relatively compact. Thus the Schauder's theorem guarantees the existence of a fixed point for $\widetilde{\mathcal{S}}$ (i.e., the solution of the nonlinear regularized problem).

Eventually it is proved that the limit for $\epsilon \to 0$ converges to a solution of the original problem.

Beirao da Veiga [BdV04, FQV09] provides a proof for the existence of a strong solution in 2D, where the structure is modeled through a 1D *generalized string* model [FQV09, Ch.3.4]. As in the previous analysis, the problem is formulated as a fixed point problem and the Schauder's theorem is employed to show the existence. However the equations are written on an ALE reference domain, and the a priori estimates are performed there. The final requirements on the regularity of the solution in this case are more restrictive than in the previous case.

Finally the existence of a strong solution for a 3D FSI problem is derived by Cheng et al. in [ACCS06]. The structure is modeled by a nonlinear Koiter shell model without the inertial term. As in the previous work the analysis is performed in the reference configuration. In this analysis a uniqueness result is obtained under appropriate compatibility conditions between initial and boundary data.

2.8.2 Stability and Geometric Conservation Law

The stability of the numerical schemes for FSI in ALE form is currently subject of research. As mentioned in Section §2.4 the unconditional stability on fixed domains does not imply that the same method used on moving domain is unconditionally stable as well. Usually to guarantee stability of a method it is necessary to find a bound for the energy of the discretized system. This is achieved by means of a-priori error estimates (see [Nob01]). However this approach is not always possible (or easy), thus other types of stability can be devised [FGG01].

A well established condition that a discretization method should satisfy is the *Discrete Geometric Conservation Law* (DGCL). A numerical scheme is said to satisfy the DGCL if it is capable of reproducing constant (in space and time) solutions in the absence of forcing terms and with proper boundary conditions. This is not always true for problems in moving domains adopting an ALE frame, and it depends on the time discretization chosen. The role of DGCL in the stability of numerical schemes for problems in moving domains was investigated in the last years, in particular in Nobile [Nob01] it leads to an unconditional stability result for the implicit Euler scheme, which will be recalled below, while in Farhat et al. [FGG01] DGCL in some cases is found to be equivalent to a discrete maximum principle.

The fulfillment of DGCL is automatic for any time discretization if the equations are written in non-conservative form. In fact, taking (2.7.1) and substituting a constant solution we obtain an identity independently of the time discretization employed.

On the other hand rewriting the conservative form (2.7.6) for a constant solution and without forcing term, we end up with the following equation

$$D_t \int_{\omega_{\mathcal{A}}} \rho_f \mathbf{u} \cdot \mathbf{v}_f \; d\Omega_t = \int_{\omega_{\mathcal{A}}} (\rho_f \mathbf{u} \cdot \nabla_x \mathbf{w}) \cdot \mathbf{v}_f \;\; d{\Omega_t}^f, \tag{2.8.1}$$

which is the weak form of the Leibnitz transport formula (1.2.7) for a constant solution. Integrated in time it gives

$$\int_{{\omega_{\mathcal{A}}}^{n+1}} \rho_f \mathbf{u} \cdot \mathbf{v}_f \; d\Omega_t - \int_{{\omega_{\mathcal{A}}}^{n}} \rho_f \mathbf{u} \cdot \mathbf{v}_f \; d\Omega_t = \mathcal{INT}_{t_n}^{t_{n+1}} \left[\int_{\omega_{\mathcal{A}}} (\rho_f \mathbf{u} \cdot \nabla_x \mathbf{w}) \cdot \mathbf{v}_f \; d\Omega_t \right], \tag{2.8.2}$$

where the term $\mathcal{INT}_{t_n}^{t_{n+1}}$ denotes a chosen numerical integration in time from t_n to t_{n+1}. While condition (2.8.1) is referred to as *Geometric Conservation Law* (GCL) condition and

it is a conservation equation, (2.8.2) is the *discrete geometric conservation law* and its form depends on the choice of the time integrator.

Thus the DGCL condition consists in choosing a time integration scheme which allows to satisfy exactly (1.2.7) for a constant solution. It can also be interpreted as a discretization of Jacobi's formula(1.2.5): if DGCL holds then the discrete counterpart of (1.2.5) holds exactly. We see from this relation that the choice of the integration rule for the part of the convective term related to the ALE description determines whether the scheme satisfies or not the DGCL. A different time integration can be chosen for the term $\mathbf{u} \cdot \nabla_x \mathbf{w}$ than for the other terms, leading to a scheme satisfying the DGCL. A rule to choose the time integration in order to satisfy (2.8.1) consists in taking an integration scheme of order $d \cdot s - 1$, where d is the number of spatial dimensions and s is the polynomial order of the nodal displacement. A proof is given in [Nob01, Ch.1.8].

An a-priori error estimate for an FSI scheme, for which unconditional stability is proved, is derived in [Nob01, Ch.4.7]. The equations are discretized in time using an implicit Euler scheme satisfying the DGCL (i.e., using a mid point scheme to discretize the fluid domain velocity) for the fluid and BDF1 for the structure. A piecewise linear in time ALE map is used. In the energy estimates the DGCL comes into play when the term

$$\int_{t_n}^{t_{n+1}} \int_{\Omega_t^f} \mathbf{u}_h^{n+1} \nabla_x \cdot \mathbf{w}^\star \; d{\Omega_t}^f$$

has to be bounded. In fact due to (2.8.2) (taking $\mathbf{u}$ as test function) the following equation holds:

$$\mathcal{INT}_{t_n}^{t_{n+1}} \int_{\Omega_t^f} \|\mathbf{u}_h^{n+1}\|^2 \nabla_x \cdot \mathbf{w}^\star \; d{\Omega_t}^f = \int_{{\Omega_t^f}^{n+1}} \|\mathbf{u}_h^{n+1}\|^2 \; d{\Omega_t}^f - \int_{{\Omega_t^f}^{n}} \|\mathbf{u}_h^{n+1}\|^2 \; d{\Omega_t}^f. \quad (2.8.3)$$

Substituting in the energy equation this identity allows to devise estimates which are independent of the mesh velocity $\mathbf{w}$. Such results have not been found in case of the non-conservative formulation, for which only conditional stability results have been proven.

In [FGG01] the importance of DGCL in the stability of numerical methods for nonlinear equations in moving domains is investigated. The analysis is applied to finite volumes schemes. The stability is defined in the sense of a *discrete maximum principle*, stating that there exist two constants M and m such that the space average $\bar{u}^n$ of the numerical solution in a cell has to satisfy $m \leq \bar{u}^n \leq M \; \forall n$. The DGCL is shown to be necessary and sufficient condition in order to satisfy this maximum principle in the case of theta-methods.

Also in [BG04] the influence of DGCL is addressed. The result found is that satisfy DGCL is neither necessary nor sufficient to guarantee the stability (in a sense specified in [BG04]) of a numerical scheme. In fact a condition on the time step is derived for which an implicit Euler scheme that does not satisfy DGCL is stable. However also in [BG04] the importance of using a scheme which satisfies DGCL is stressed, since e.g. the implicit Euler satisfying DGCL is unconditionally stable, as discussed above.

In [LM01] it is also shown that not satisfying DGCL introduces for a conservative scheme an error in the energy conservation which depends on the time discretization for the fluid employed. Furthermore it is shown that using the non conservative formulation the error introduced in the energy balance is proportional to $\nabla_x \cdot \boldsymbol{\beta}$. Thus choosing ALE maps which satisfy $\nabla_x \cdot \mathbf{w} = 0$ can reduce the energy dissipation in non conservative schemes.

2.8.3 Strong versus Weak Coupling

The choice of the time discretization introduces further distinctions among the methods. Besides the characterization of the methods as *Fully Implicit*, *Geometry–Convective Explicit*, etc., another classification concerns the way in which the coupling conditions are discretized in time. We can devise three main classes of methods regarding the coupling strategies:

- Strongly (implicitly) coupled schemes. The coupling condition is enforced exactly at each time-step (e.g. by means of an extra loop). As a result they can require a variable amount of *coupling iterations*, depending on the algorithm used and on physical parameters [KW08b, MNS06, BNV08, BQQ08a].

- Fractional step (semi-implicit coupling) schemes[4]. They involve a splitting of the system in two problems, a solution of both and a successive correction of the fluid velocity. These schemes require in general a time step restriction to converge [FGG07, ACF09, QQ07, BQQ08b].

- Weakly (explicitly) coupled, or staggered, schemes. These methods are very cheap since they require just one solution of each subproblem per time step. In some cases they are proved to be unconditionally unstable (due to the *added mass effect*, as explained below), however recent studies show that this instability can be overcome, at the expense of introducing suitable dissipation terms [BF09]. Weakly coupled schemes were recently addressed also in e.g. [GGCC09, BRvdV09, JGLJ11].

Added Mass Effect

In this section we report the analysis performed in Causin et al. [CGN05], where the source of instability for a weakly coupled 2D simplified FSI models is identified. The model considered for the fluid is linear, incompressible, and inviscid. The fluid domain Ω_t^f consists in a rectangle, with Neumann boundary conditions imposed on the lateral edges Γ_N^f, Dirichlet slip conditions on the bottom edge Γ_D^f and the velocity continuity coupling condition on the top edge, which represents the FS-interface Γ. The domain can be interpreted as the surface generating a 3D pipe by rotating around the bottom axis, although 3D axisymmetry is indeed neglected. The fluid domain is kept fixed and the Eulerian frame is used. To be consistent with the previous chapter we keep the same notations, although everything in this case is two dimensional. The equations for the fluid read: find the fluid velocity $\mathbf{u}$ and the pressure p such that

$$\begin{aligned} \rho_f \frac{\partial \mathbf{u}}{\partial t} + \nabla_x p &= 0 \quad && \text{in } \Omega_t^f, \\ \nabla_x \cdot \mathbf{u} &= 0 \quad && \text{in } \Omega_t^f, \\ p &= \bar{p} \quad && \text{on } \Gamma_N^f, \\ \mathbf{u} \cdot \mathbf{n} &= 0 \quad && \text{on } \Gamma_D^f, \\ \mathbf{u} \cdot \mathbf{n} &= w \quad && \text{on } \Gamma, \end{aligned} \tag{2.8.4}$$

[4]The splitting that occurs in these methods is usually not obtained through domain decomposition because the solid displacement is strongly coupled with the fluid pressure (as it will be clarified at the end of the section)

where $\bar{p}$ and w are given functions on Γ. Taking the divergence of the first equation of system (2.8.4) we obtain after some substitutions

$$\begin{aligned} -\Delta_x p &= 0 && \text{in } \Omega_t^f, \\ p &= \bar{p} && \text{on } \Gamma_N^f, \\ \nabla_x p \cdot \mathbf{n} &= 0 && \text{on } \Gamma_D^f, \\ \nabla_x p \cdot \mathbf{n} &= -\rho_f \tfrac{\partial w}{\partial t} && \text{on } \Gamma. \end{aligned} \tag{2.8.5}$$

The structure equation is modeled by a one dimensional generalized string model. Calling d_s the scalar solid displacement, h_s the thickness of the wall and b the stiffness, we have

$$\rho_s h_s \frac{\partial^2 d_s}{\partial t^2} + a d_s - b \frac{\partial^2 d_s}{\partial x^2} = p, \tag{2.8.6}$$

where a is a parameter related to the constitutive equation and to the characteristic dimensions of the wall.

We refrain here from discussing the variational formulation and the functional settings of the problem. We just recall that the solution p of the fluid problem is in the space $H^1(\Omega_t^f)$, that the ALE map is the identity I since the fluid domain is fixed, and that the solid solution is in $U^{\widehat{\Gamma}}$, as well as w. Then it is possible to define a linear operator $R : U^{\widehat{\Gamma}} \to U^p$, such that $R\tau$ satisfies

$$\begin{aligned} -\Delta_x R\tau &= 0 && \text{in } \Omega_t^f, \\ R\tau &= 0 && \text{on } \Gamma_N^f, \\ \nabla_x R\tau \cdot \mathbf{n} &= 0 && \text{on } \Gamma_D^f, \\ \nabla_x R\tau \cdot \mathbf{n} &= -\rho_f \tau && \text{on } \Gamma. \end{aligned} \tag{2.8.7}$$

The *added mass operator* $\mathcal{M}_\mathcal{A}$ is the trace of R on Γ, i.e.,

$$\begin{aligned} \mathcal{M}_\mathcal{A} : \quad U^{\widehat{\Gamma}} &\to U^{\widehat{\Gamma}} \\ \tau &\mapsto \operatorname{tr}_\Gamma (R\tau). \end{aligned}$$

Exploiting the linearity, we can rewrite the system (2.8.5) as sum of two contributions, one coming from (2.8.7) (considering $\tau = \frac{\partial^2 d_s}{\partial t^2}$), and the other given by

$$\begin{aligned} -\Delta_x p^\star &= 0 && \text{in } \Omega_t^f, \\ p^\star &= \bar{p} && \text{on } \Gamma_N^f, \\ \nabla_x p^\star \cdot \mathbf{n} &= 0 && \text{on } \Gamma_D^f, \\ \nabla_x p^\star \cdot \mathbf{n} &= 0 && \text{on } \Gamma, \end{aligned} \tag{2.8.8}$$

so that

$$p = p^\star - \rho_f R \frac{\partial^2 d_s}{\partial t^2} \text{ in } \Omega_t^f.$$

Restricting to the FS-interface we get

$$p_\Gamma = \operatorname{tr}_\Gamma (p^\star) - \rho_f \mathcal{M}_\mathcal{A} \frac{\partial^2 d_s}{\partial t^2} \text{ on } \Gamma. \tag{2.8.9}$$

Substituting then into the structure equation (2.8.6) we can see that the quantity $\rho_f \mathcal{M}_\mathcal{A} \frac{\partial^2 d_s}{\partial t^2}$ plays the role of an extra mass.

The eigenvalues of the added mass operator $\mathcal{M}_\mathcal{A}$ for this simplified problem can be found explicitly as functions of the characteristic dimensions of the wall (radius, thickness, and length).

In [CGN05] explicit and implicit coupling algorithms are analyzed in the case of null stiffness coefficient $b = 0$ and using an implicit Euler and Leap-Frog time discretization for the fluid and structure respectively.

The strong form of the explicitly coupled scheme reads

$$\begin{aligned}
\rho_f \frac{\mathbf{u}^n - \mathbf{u}^{n-1}}{\delta t} + \nabla_x p^n &= 0 && \text{in } \Omega_t^f, \\
\nabla_x \cdot \mathbf{u}^n &= 0 && \text{in } \Omega_t^f, \\
p^n &= \bar{p}^n && \text{on } \Gamma_N^f, \qquad (2.8.10) \\
\mathbf{u}^n \cdot \mathbf{n} &= 0 && \text{on } \Gamma_D^f, \\
\mathbf{u}^n \cdot \mathbf{n} &= \tfrac{\partial {d_s}^n - {d_s}^{n-1}}{\partial t} && \text{on } \Gamma, \\
\rho_s h_s \frac{{d_s}^{n+1} - 2{d_s}^n + {d_s}^{n-1}}{\delta t^2} + a{d_s}^n &= p^n && \text{on } \Gamma. \qquad (2.8.11)
\end{aligned}$$

Equation (2.8.11) after the substitution of $\mathcal{M}_\mathcal{A}$ reads

$$\rho_s h_s \frac{{d_s}^{n+1} - 2{d_s}^n + {d_s}^{n-1}}{\delta t^2} + \rho_f \mathcal{M}_\mathcal{A} \frac{{d_s}^n - 2{d_s}^{n-1} + {d_s}^{n-2}}{\delta t^2} + a{d_s}^n = p_{ext}^n, \qquad (2.8.12)$$

where $p_{ext}^n = \mathrm{tr}_\Gamma\,(p^{\star n})$. Computing the spectral radius of this scheme gives a condition for the instability: the scheme is unconditionally unstable if

$$\frac{\rho_s h_s}{\rho_f \mu_{max}} < 1,$$

where μ_{max} is the maximum eigenvalue of the added mass operator.

This instability is seen in hemodynamic applications, for which $\rho_f \approx \rho_s$, while e.g. in aerodynamics usually $\rho_f \ll \rho_s$ and the explicit coupling is stable.

An extension to this analysis is derived in Förster et al. [FWR07], where an approximation of the added mass operator is obtained at the algebraic level for a three dimensional explicitly coupled FSI problem. Different time discretizations are compared and stability conditions are derived. All the schemes tested showed instabilities for a critical mass density ratio. Interestingly by decreasing the time-step or increasing the order of the scheme (i.e., reducing the numerical dissipation) does increase the instability.

The algebraic expression for the added mass operator is found by considering a linearized version of the fluid equation in which both the dependence of the domain on the solid displacement and the convective term are neglected. Furthermore the stiffness matrix is also neglected, since when considering very small time steps the main contribution is given by the mass term, which dominates the others. The algebraic system is written with respect to the acceleration (i.e., taking the time derivative of the fluid momentum equation) and it is reduced formally to the interface by means of a Schur complement. Then substituting in the solid equation it is shown that the fluid force contribution on the interface corresponds entirely to an added mass.

In [BA09] the added mass effect is investigated numerically on a benchmark 3D geometry. The FSI problem is explicitly coupled using an energy conserving scheme. The structure equation is discretized in time using the generalized-α scheme, while the fluid is solved explicitly. A smaller timestep is used for the fluid field with respect to the solid. The limiting mass ratio for stability is found to depend on the number of sub-cycles performed by the fluid solver: increasing the number of sub-cycles the instability increases. This result confirms the fact that reducing the energy dissipation the system instability increases.

Fractional–Step and Explicit Couplings

An option to avoid the implicit coupling consists in the fractional-step (semi-implicit) coupling algorithms. In these schemes only the equation for the fluid pressure is strongly coupled with the structure dynamic. Indeed these methods introduce a splitting which is not the traditional one between the fluid and the solid variables.

A fractional step algorithm for FSI is introduced in the original paper by Fernández et al. [FGG07]. There the differential Chorin–Temam method is extended to the FSI system. It consists, for each time level, in solving part of the fluid equation, obtaining an approximated velocity which is not divergence free, and then in solving the solid equations strongly coupled with the remaining part of the fluid equations, linearized using the approximated velocity previously computed (*projection* step).

In [FGG07] the analysis on a simplified problem allows to devise a conditional stability result for the implicit Euler time discretization for both fluid and structure (where the fluid momentum equation is written in non conservative form). It is interesting to observe that the stability condition found still depends on the added mass effect through the mass density ratio.

In Quaini et al. [QQ07] a similar approach is represented at the algebraic level as an inexact factorization of the system. In particular the Yosida method is generalized to the algebraic FSI system. Three equations are solved sequentially at each time step:

- the fluid momentum and the geometry equations are solved, using an extrapolation for the solid displacement and for the fluid pressure;
- the pressure-solid Schur complement equation is solved (projection step), approximating the Schur complement using the zeroth order term of the Neumann expansion of the fluid momentum equation's matrix;
- the velocity is corrected solving again the momentum equation, using the pressure found in the step above.

This method (Fluid–Structure Yosida, FSY) can be interpreted as an inexact block LU factorization of the system matrix $A \approx \tilde{L}U$. In this approach the upper block triangular factor U in the inexact factorization is not approximated.

In Badia et al. [BQQ08b] another method is proposed in which both the factors are approximated in the inexact factorization. This method is called Pressure-Interface Correction (PIC). The FSY and PIC schemes are conditionally stable, while the *splitting error* is of order one in time. The two algebraic inexact factorization methods (FSY and PIC) are detailed in [Qua09], where the differences with the method introduced in [FGG07] are highlighted.

Another modification of the original algorithm reported in [FGG07] is introduced in Astorino et al. [ACF09]. There the Dirichlet and Neumann coupling conditions assigned to the

sub-problems are replaced with Robin-type boundary conditions derived from the Nitsche's interface method [Han05, BF09]. With this modification it has been possible to prove a conditional stability result, which is independent of the mass density ratio (and thus of the added mass effect).

In Burman et al. [BF09] the coupling between the fluid and solid problems is also enforced through Nitsche's interface conditions, which allow to impose weakly the essential boundary conditions (in our case the velocity continuity). The scheme proposed in [BF09] can be seen as a *Dirichlet–Robin* explicitly coupled scheme, since the splitting in a fluid and structure subproblems associates a (weak) Dirichlet coupling condition to the fluid problem and a Robin coupling to the structure. We remark that in [BF09] neither the velocities nor the test functions are constrained to match at the FS interface. However as stated in [BF09] the Nitsche's coupling is not sufficient alone to overcome the instability due to the added mass effect, and to recover an energy estimate independent of the mass density ratio (and thus of the added mass effect, as shown above). Thus a consistent penalization term is introduced, that reduces the order of convergence in time (which should be one for the chosen time discretization) to $\frac{1}{2}$, but allows to obtain for a simplified problem a CFL-like condition independent of the added mass. A strategy is also proposed to recover the first order convergence by means of a correction step.

2.9 Geometrical Multiscale

The computational cost of the full FSI simulations is sometimes prohibitive, e.g. when we aim at the simulation of an entire arterial tree. Furthermore the inlet-outlet boundary conditions for the arterial segment of interest are not always available, and the arterial network can influence these values (e.g. through backward travelling pressure waves). For these reasons reduced models for the cardiovascular network have been devised, which allow a fast and reliable simulation of some macroscopic quantities (like pressure, flux, area of a transversal section). In particular one and zero dimensional models have been studied.

In this section we illustrate an implicit algorithm to couple a 3D FSI model with a reduced order model. Since we are more focused on the coupling we do not describe the details and the derivation of the reduced order model, but we just retrieve the model and the assumptions which are at the basis of the approximation.

A classical 1D model for the arteries is the one described in T. J. R. Hughes [HL73, Hug74]. Recently in [RMP+09] a similar model, accounting for the whole cardiovascular system through a network of more than 100 different branches, was validated against in-vivo measurements. All the models that we recall here, plus other more complicated reduced order models, are described more extensively in J. Peiró, A. Veneziani [FQV09, Chap 10].

The 1D models used in [HL73, Hug74] can be obtained by integrating and averaging the fluid momentum and mass conservation equations in moving domains, under the hypothesis of a straight vessel. We call x the axial coordinate. The averaged quantities considered in the model are the flux over a radial section $Q(x)$, the area of the section $A(x)$ and the mean pressure over the section $p(x)$. The effect of the structure is taken into account through an algebraic pressure-area relationship in the simplest case, or through an ordinary differential equation (0D model) which relates pressure and mean radial displacement and can account also for visco-elastic or inertial effects. The 1D model relies on some assumptions (like the straightness of the vessel) which are often not fulfilled in real geometries. Taking

the space average on the section S of the term representing the viscous stress (in 3D $\mu_f \Delta_x \mathbf{u}$ for incompressible Newtonian fluids) proportional to the fluid mean axial velocity, so that $\frac{1}{A}\int_S \nabla_x \cdot \sigma_f \; dS \approx -\frac{dp}{dx} + \frac{K_R}{A} Q$, leads to the following hyperbolic equation representing the fluid momentum conservation

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\alpha \frac{Q^2}{A}\right) + \frac{A}{\rho_f}\frac{\partial p}{\partial x} + K_R \frac{Q}{A} = 0. \tag{2.9.1}$$

This coupled with the mass conservation equation

$$\frac{\partial A}{\partial t} + \partial_x Q = 0, \tag{2.9.2}$$

and the structure model, represented by

$$p = \Phi(A, \dot{A}, \ddot{A}), \tag{2.9.3}$$

gives the 1D system for an artery segment in the variables A, p, Q (where either p or A is usually eliminated using (2.9.3)). The eigenvalues of this system are usually real and have different signs for hemodynamic applications, thus the system is (nonlinear) hyperbolic and subcritical. It is possible to diagonalize the differential system by a change of variables. The new variables obtained are the *characteristic variables*

$$W_1 = Q/A + 4A^{\frac{1}{4}}\sqrt{\frac{\beta}{2\rho_f}}, \tag{2.9.4}$$

$$W_2 = Q/A - 4A^{\frac{1}{4}}\sqrt{\frac{\beta}{2\rho_f}}, \tag{2.9.5}$$

where $\beta > 0$ is a coefficient related to the mechanical properties of the vessel wall. Using the characteristic variables is particularly useful when we want to impose an absorbing boundary condition (i.e., a condition which avoids numerical reflections).

This 1D model can be improved to handle the pressure loss when a bifurcation occurs, accounting for the bifurcation angle. Other 1D models accounting for curved vessels are described in [GN93, GNS93], where the equations are recast to a local frame of reference and the metric change is taken into account.

The absorbing boundary condition at the termination of the 1D segment models the continuation of the branch as if the segment length was infinite. Other boundary conditions can model the presence of valves, smaller arterial trees and so on. In particular they consist in zero dimensional models coupled with the 1D artery. The absorbing boundary conditions are first order ordinary differential equations which under some assumptions (in particular the convective term is neglected and the area A is fixed) can be considered linear. In the latter case they have the same form as the equations for electric circuits, where Q represents the current, p the voltage. The inertia of blood is modeled by an inductance, the capacity is related to the compliance of the artery (represented by A or p). The resistance is related to the blood viscosity coefficient μ_f, i.e., to the coefficient K_R in (2.9.1). There are many options to model the effect of the peripheral circulation on the termination of an artery by means of 0D model. Usually a *Windkessel model* is used. It consists in expressing the *transfer function* $\zeta(\omega)$ (in the frequency domain) as a function of the chosen values for the resistance, capacity and inductance. 0D models of the valves are the analogue of diodes in electrical circuits.

Many different 0D models have been devised for the heart, see e.g. [KVCF$^+$09], which usually depend on an *elastance function* E_v that has to be tuned, expressing the relationship between the pressure and the volume of the ventricle.

Coupling these reduced order dynamics with a 3D model can be achieved in several ways. The values of pressure, area or flux coming from the reduced model can be assigned to the 3D model pointwise or in a *defective* fashion. The first methodology consists in either assuming a constant value of the coupling variable on the 3D boundary section (strategy usually adopted for the pressure), or to assign a 1-parameter family of possible profiles (the velocity is usually imposed assuming a parabolic or Womersley profile). The second methodology consists in assigning the integral of the coupling variable over the boundary section by means of a Lagrange multiplier, see e.g. [FGNQ02a, Ver10]. For instance imposing the flux Q on a boundary γ_{flux} of a 3D FSI model is achieved by enforcing the constraint

$$\int_{\gamma_{flux}} \mathbf{u}\, d\gamma = Q. \tag{2.9.6}$$

We call this boundary condition of *defective flux* type. Imposing (2.9.6) through Lagrange multipliers adds one line and one column to the matrix in (2.7.10), thus it constrains only one degree of freedom on the boundary γ_{flux}, while no profile is imposed.

The stability of the coupling between one and three dimensional FSI models was investigated e.g. in [FGNQ01]. The coupling conditions assigned on the interface between the different models may be:

1. the continuity of the area,
2. the continuity of the flux,
3. the continuity of the mean pressure,
4. the continuity of the characteristic variable (expressed as a function of Q and A becomes a nonlinear Robin type condition, see [NV08]).

There are several algorithms that allow to enforce the coupling. The easiest are probably explicit of Gauss–Seidel type. More robust algorithms are implicit and require nonlinear iterations (see e.g. [MBDQ11] and Section §3.3). In Section §3.3 we define the tangent problem which is used in this situation.

Remark 2.9.1. *Because of the subcritical hyperbolic nature of the 1D model only one boundary condition per termination is required. The average value coming from the 1D can be coupled with either the fluid or the solid (or geometry) problems in 3D. However since both these 3D problems require a condition on the 3D-1D interface it is possible to assign to the 3D problem uncoupled with the 1D model a condition devised from the 1D solution. For instance, if the coupling condition is the flux continuity, we can take the value of the area from the 1D to assign the structure displacement on the 3D-1D interface. This choice has been considered e.g. in [FGNQ01].*

In our simulations we do not use this strategy, mainly because it would be not trivial to implement it for arbitrary geometries, and because we did not observe spurious reflections by simply taking a Neumann homogeneous boundary condition for the terminations of the vessels structure.

Part II

Solution Strategies for the FSI problem

Nonlinearities and Newton Method

In this chapter we focus on some different formulations of the nonlinear problem (2.7.10) arising from the discretization of the FSI system in time and space (in fully implicit form). We present some of the most popular methodologies usually adopted in literature to solve the various nonlinearities. Throughout this chapter we omit the time-step index n. Furthermore we abbreviate the total and partial derivatives of a functional $F(\mathbf{X})$ with respect to a vector field $\mathbf{X}$ with $D_{\mathbf{X}}(\cdot)$ or $\partial_{\mathbf{X}}(\cdot)$ respectively and we denote with $|_{\mathbf{X}}$ the evaluation of a function at the point $\mathbf{X}$.

In the first two sections we recall the formulation of the Newton method, we describe different fixed point formulations of the FSI problem, and we discuss the Newton method applied to the fixed point problems introduced. In the third section we explain how to compute the derivatives of the fluid problem with respect to the domain motion. Those derivatives have a non trivial form and are often neglected, they can be viewed as *shape derivatives*. In the fourth section we describe the Newton method adopted to solve the geometrical multiscale coupled system introduced in Section §2.9. In particular we express the coupled problems by exploiting the notations and the framework introduced in Section §3.2.

3.1 Newton Method

For a generic nonlinear problem $M(\mathbf{X}) = 0$, the Newton–Rhapson method can be represented by the following steps. Given the initial solution $\mathbf{X}^o$ (and starting with $k = 0$):

- Compute the residual $M(\mathbf{X}^k)$ and check the stopping criterium. The latter is usually based on the residual and have the following form

  $$M(\mathbf{X}^k) \leq \max\{\epsilon^r \|M(\mathbf{X}^o)\|, \epsilon^a\},$$

  where $\epsilon^r, \epsilon^a \in \mathbb{R}$ are the relative and absolute tolerances respectively.

- Compute the (possibly approximated) Jacobian matrix $J(\mathbf{X}^k) = D_{\mathbf{X}} M(\mathbf{X})|_{\mathbf{X}^k}$, and solve the Jacobian system

  $$J(\mathbf{X}^k)\delta\mathbf{X} = -M(\mathbf{X}^k).$$

- Update the solution $\mathbf{X}^{k+1} = \mathbf{X}^k + \omega\delta\mathbf{X}$, where ω is a relaxation parameter (often $\omega = 1$) and increase k.

- Proceed to the next iteration.

Advantages and disadvantages of this approach are often discussed in the FSI literature. The quadratic convergence of the Newton method is an appealing feature, however acceleration (Aitken) techniques can be devised in order to improve the convergence

of the fixed point algorithms as well [KW08b, MW01, MWR01, DBHV08, Dep04]. Furthermore the computation of the Jacobian is expensive both in terms of coding and computational time. The Newton algorithm is often sensible to the initial guess for the solution, thus the fixed point method can be used for the first nonlinear iterations, or a *coarse nonlinear problem* can be solved, to obtain a better initial solution at every time iteration [BC10b]. The Jacobian matrix is often approximated e.g. by neglecting some or all of the shape derivatives terms [MS03, Tez04, Hei04, BC09, BCHZ08], by using simplified models [GV03] or using techniques based on extrapolation from the previous nonlinear iterations [KGF$^+$09, DBV09, DAV10]. These inexact Newton strategies are often coupled with a linesearch technique which is meant to select an optimal relaxation parameter ω [KW08b, KGF$^+$09].

Several ways to compute the complete Jacobian for the Newton scheme in FSI are reviewed in literature. In particular if a matrix–free method like GMRES is used to solve the Jacobian system there is no need to build explicitly the Jacobian matrix J, it is sufficient the Jacobian–vector product. This product (or a part of it) can be approximated e.g. by finite differences (see e.g. [TSS06, KGF$^+$09]):

$$J\delta\mathbf{X} = \frac{M(\mathbf{X} + \epsilon\delta\mathbf{X}) - M(\mathbf{X})}{\epsilon}, \tag{3.1.1}$$

for a suitable (small enough) ϵ. However such approximation is costly, since it requires an evaluation of the nonlinear problem operator $M(\mathbf{X} + \epsilon\delta\mathbf{X})$ at each Jacobian–vector product. Furthermore it is not evident how to choose a proper value for the parameter ϵ. Another method to compute the Jacobian is used in Hron [Hro01]. The idea consists of approximating a column j of the Jacobian matrix as finite difference, substituting to the vector $\mathbf{X}$ in (3.1.1) the vector $\mathbf{e}_j$ of the canonical basis and computing the finite difference for all the degrees of freedom. The form of the analytic Jacobian–vector product for a particular formulation (cf. Section §3.2) is described in detail in [FM05] while the complete Jacobian matrix is assembled analytically in e.g. [CDFQ11, BC10a](cf. Section §3.4).

Let us consider the full system (2.7.10). Being $\mathbf{X} = (\mathbf{u}_f, \widehat{\mathbf{d}}_s, \widehat{\mathbf{d}}_f)^T$ and $\delta\mathbf{X} = (\delta\mathbf{u}_f, \delta\widehat{\mathbf{d}}_s, \delta\widehat{\mathbf{d}}_f)^T$, the iteration $(k+1)$ of the Newton scheme reads:

$$\begin{pmatrix} D_{\mathbf{u}_f}\mathsf{F} & D_{\widehat{\mathbf{d}}_s}\mathsf{F} & D_{\widehat{\mathbf{d}}_f}\mathsf{F} \\ D_{\mathbf{u}_f}\mathsf{S} & D_{\widehat{\mathbf{d}}_s}\mathsf{S} & D_{\widehat{\mathbf{d}}_f}\mathsf{S} \\ 0 & D_{\widehat{\mathbf{d}}_s}\mathsf{G} & D_{\widehat{\mathbf{d}}_f}\mathsf{G} \end{pmatrix} \begin{pmatrix} \delta\mathbf{u}_f \\ \delta\widehat{\mathbf{d}}_s \\ \delta\widehat{\mathbf{d}}_f \end{pmatrix} = - \begin{pmatrix} \mathsf{F}({\mathbf{u}_f}^k, \widehat{\mathbf{d}}_s^k, \widehat{\mathbf{d}}_f^k) \\ \mathsf{S}({\mathbf{u}_f}^k, \widehat{\mathbf{d}}_s^k, \widehat{\mathbf{d}}_f^k) \\ \mathsf{G}({\mathbf{u}_f}^k, \widehat{\mathbf{d}}_s^k, \widehat{\mathbf{d}}_f^k) \end{pmatrix}, \tag{3.1.2}$$

$$\mathbf{X}^{k+1} = \mathbf{X}^k + \delta\mathbf{X}.$$

3.2 Fixed Point Formulations

We introduce here the concept of *nonlinear Schwarz preconditioners* (see e.g. Cai and Keyes [CK02]), which is not directly related to the linear preconditioner one. We call $\widehat{\Omega}$ a generic domain where the differential problem is defined and take a possibly overlapping partition $\{\Omega_i\}_{1\leq i\leq N}$ of the domain $\widehat{\Omega}$. Let us consider the problem $M(\mathbf{X}) = 0$ written in weak form (following the notation used in e.g. [FM05]). We call U the functional space where the global solution $\mathbf{X}$ is defined, chosen accordingly with the differential problem to be solved, while V represents the space of the test functions. We call $U_i(\Omega_i)$ some functional spaces

where the local solutions $\mathbf{X}|_{\Omega_i}$ are searched, and $V_i(\Omega_i)$ the corresponding test space. With the aim of representing the weak form in a compact way we introduce the notation $< \cdot, \cdot >$ for the usual inner product in L^2. We introduce the nonlinear problem operator $M(\cdot) : U \to V'$ such that $< M(\mathbf{X}), \psi >= 0 \; \forall \psi \in V$.

We define the restriction $R_i : U \to U_i$ and prolongation $R_i^T : U_i \to U$ operators such that $R_i(\mathbf{X}) = \mathbf{X}|_{\Omega_i}$ and

$$R_i^T(\mathbf{X}) = \begin{cases} \mathbf{X}(\widehat{x}) & \text{if } \widehat{x} \in \Omega_i, \\ 0 & \text{otherwise.} \end{cases}$$

For simplicity we use the same notation also for the restriction-prolongation operators acting on the test functions.

We define the *subdomain problem operators* $T_i : U \to (V_i)'$ such that

$$< T_i(\mathbf{X}), \psi_i > =< M(\mathbf{X}), R_i^T \psi_i > \qquad \forall \psi_i \in V_i. \tag{3.2.1}$$

The operators T_i are well defined if the subproblems are well posed.

From now on the equalities must be interpreted in weak sense (in $(V_i)'$ or $(V)'$), though we refrain from writing it explicitly to light the notation.

We define the *solver operators* $\mathcal{T}_i : U \to U_i$ such that $\mathcal{T}_i(\mathbf{X}) = R_i(\mathbf{X} - R_i^T T_i(\mathbf{X}))$.

The nonlinear additive Schwarz preconditioned system reads

$$\mathcal{M}^{AS}(\mathbf{X}) = \sum_{i=1}^{N} R_i^T(T_i(\mathbf{X})) = 0. \tag{3.2.2}$$

The analogy with the linear additive Schwarz preconditioners is evident (see [CK02]) and partially explains why the term *preconditioners* is used also in the nonlinear case. In [CK02] it is proved the equivalence of the nonlinear additive Schwarz preconditioned formulation (3.2.2) with the original one $M(\mathbf{X}) = 0$.

The multiplicative counterpart $\mathcal{M}_N^{MS}(\mathbf{X})$ of (3.2.2) can be defined as

$$\mathcal{M}_N^{MS}(\mathbf{X}) = R_N \mathbf{X} - \mathcal{T}_N \circ \sum_{i=1}^{N-1} \mathcal{M}_i^{MS}(\mathbf{X}) = 0, \tag{3.2.3}$$

where the quantities $\mathcal{M}_i^{MS}$ are defined recursively by

$$\begin{cases} \mathcal{M}_1^{MS}(\mathbf{X}) = \mathbf{X} - R_1^T T_1(\mathbf{X}), \\ \mathcal{M}_j^{MS}(\mathbf{X}) = R_j^T \mathcal{T}_j \circ \sum_{i=1}^{j-1} \mathcal{M}_i^{MS}(\mathbf{X}). \end{cases}$$

The additive and multiplicative nonlinear Schwarz preconditioners thus define a set of nonlinear sub-problems which have to be solved in order to obtain the solution of the global problem. A fixed point method applied to these problem formulations can lead to the classical *Schwarz methods*, which are particular cases of *Domain Decomposition* (DD) *methods* [TW05, QV99].

In the FSI context many algorithms have been devised which take advantage of the multiphysics nature of the problem by splitting the domain into the fluid and solid parts, or into fluid, solid and FS interface parts. These algorithms usually start from a formulation of the

problem which can be described by means of the nonlinear multiplicative Schwarz preconditioners with no overlap. The popular algorithms in FSI just cover a small fraction of the possibilities, which leaves room for further investigations.

A natural way to linearize expressions (3.2.3) and (3.2.2) is by means of fixed point iterations, or, if the derivatives of the operators T_i are available, Newton iterations. In [CK02] a method to compute the derivatives of T_i as functions of the Jacobian of the full problem $M(\mathbf{X})$ is provided, and the Newton method applied to problem (3.2.2) is described.

Here to be consistent with the previous notations, and since the methods that we are going to describe deal with a non overlapping partition of the domain into a fluid and a solid part, we discard the more general concept of nonlinear Schwarz preconditioner, though keeping in mind that all the formulations that we are going to describe could be reinterpreted in such framework.

3.2.1 Dirichlet–Neumann Formulation

The fluid–geometry and solid problem operators of the fully coupled system (2.5.1) (omitting the time level index and grouping the fluid and geometry problems) read

$$\begin{aligned} \mathsf{FG}(\mathbf{u}_f, \widehat{\mathbf{d}}_s) &= 0, \\ \mathsf{S}(\mathbf{u}_f, \widehat{\mathbf{d}}_s) &= 0. \end{aligned} \tag{3.2.4}$$

This leads to the definition of the *fluid–geometry solver operator*

$$\begin{aligned} \mathcal{F} : U^s &\rightarrow U^f \times U^p \\ \widehat{\mathbf{d}}_s &\mapsto \mathbf{u}_f, \end{aligned}$$

such that $\mathsf{FG}(\mathcal{F}(\widehat{\mathbf{d}}_s), \widehat{\mathbf{d}}_s) = 0$. Similarly we define the *solid solver operator*

$$\begin{aligned} \mathcal{S} : U^f \times U^p &\rightarrow U^s \\ \mathbf{u}_f &\mapsto \widehat{\mathbf{d}}_s, \end{aligned}$$

such that $\mathsf{S}(\mathbf{u}_f, \mathcal{S}(\mathbf{u}_f)) = 0$.

The system written as in (3.2.3) reads

$$\widehat{\mathbf{d}}_s - \mathcal{S}(\mathcal{F}(\widehat{\mathbf{d}}_s)) = 0. \tag{3.2.5}$$

A fixed point strategy applied to this formulation reads

$$\widehat{\mathbf{d}}_s^{k+1} = \mathcal{S}(\mathcal{F}(\widehat{\mathbf{d}}_s^k)).$$

When the fluid problem is endowed with Dirichlet coupling condition (the velocity continuity) and the solid problem is endowed with a Neumann one (the stress continuity), this strategy corresponds to the classical *Dirichlet–Neumann* (DN) approach. It can be viewed as a nonlinear block Gauss–Seidel iterative method, or as an undamped nonlinear Richardson method. In practice a relaxation is used to improve the stability and convergence of the method. A relaxed DN iteration reads

$$\widehat{\mathbf{d}}_s^{k+1} = \omega \mathcal{S}(\mathcal{F}(\widehat{\mathbf{d}}_s^k)) + (1-\omega)\widehat{\mathbf{d}}_s^k. \tag{3.2.6}$$

Methods to automatically adapt the relaxation parameter (such as the Aitken acceleration technique [KW08b, MW01, MWR01, DBHV08, Dep04]) can be used. The DN scheme depends heavily on the added mass effect, see Section §2.8.3. In particular as the mass density ratio $\frac{\widehat{\rho}_s}{\rho_f}$ decreases the fixed point iterations required to reach a given tolerance increase and the algorithm can become very inefficient. In the case of hemodynamic the mass density ratio is almost one and this fixed point method requires a large number of iterations to satisfy the strong coupling between the two problems.

Other choices for the coupling conditions assigned to each subproblem have been investigated in literature to avoid this issue. The same scheme (3.2.6) when the Neumann coupling condition is assigned to the fluid and the Dirichlet one to the structure is called *Neumann–Dirichlet* (ND) method. The convergence properties of this method are investigated numerically in e.g. Causin et al. [CGN05], where the behavior is found to be worse than the DN's one. One possible option which solves the problem of the added mass effect is to replace the fluid Dirichlet condition of the DN scheme with a Robin condition (a linear combination between the two coupling conditions). This choice leads to the definition of the *Robin–Neumann* (RN) scheme. The behavior of this scheme is influenced by the choice of the coefficient introduced in the linear combination, thus in Badia et al. [BNV08] a method to obtain this coefficient from a simplified model is devised. Recent advances regarding this topic have been achieved in Gerardo Giorda et al. [GGNV10]; substituting also the Neumann condition on the structure with a Robin one leads to the *Robin–Robin* (RR) scheme and brings in another coefficient to be estimated.

Since fixed point iterations often feature slow convergence, it is preferable to adopt a Newton method (see e.g. [FM05, KGF$^+$09, BC10a, GV03]) for the systems (2.7.10) or (3.2.5).

The derivatives of the fluid equations with respect to the fluid domain displacement are non trivial and will be addressed in the next section. Implementing the Newton scheme for the system (2.7.10), besides the difficulties of computing the aforementioned derivatives, does not immediately allow the reuse of existing codes which can already solve the sub-problems. In Fernández and Moubachir [FM05] the authors applied the Newton method to (3.2.5), which leads to the following Newton iterations (see also e.g. [DDQ06, KGF$^+$09, DAV10])

$$\left[I - D_{\mathbf{u}_f}\mathcal{S}|_{\mathcal{F}(\widehat{\mathbf{d}}_s^k)} D_{\widehat{\mathbf{d}}_s}\mathcal{F}_{\widehat{\mathbf{d}}_s^k}\right]\delta\widehat{\mathbf{d}}_s = -\widehat{\mathbf{d}}_s^k + \mathcal{S}(\mathcal{F}(\widehat{\mathbf{d}}_s^k)), \tag{3.2.7}$$

$$\widehat{\mathbf{d}}_s^{k+1} = \widehat{\mathbf{d}}_s^k + \delta\widehat{\mathbf{d}}_s.$$

To implement this method we need to evaluate the derivatives of the solver operators in the points ${\mathbf{u}_f}^k = \mathcal{F}(\widehat{\mathbf{d}}_s^k)$ and $\widehat{\mathbf{d}}_s^k$. This is achieved in [FM05] by taking the derivatives of the equations

$$\mathsf{FG}(\mathcal{F}(\widehat{\mathbf{d}}_s^k), \widehat{\mathbf{d}}_s^k) = 0,$$
$$\mathsf{S}({\mathbf{u}_f}^k, \mathcal{S}({\mathbf{u}_f}^k)) = 0.$$

Defining $\theta_s = (\mathcal{F}(\widehat{\mathbf{d}}_s^k), \widehat{\mathbf{d}}_s^k)$ and $\theta_f = ({\mathbf{u}_f}^k, \mathcal{S}({\mathbf{u}_f}^k))$, we have the two linear systems

$$\partial_{\mathbf{u}_f}\mathsf{FG}|_{\theta_s}\partial_{\widehat{\mathbf{d}}_s}\mathcal{F}|_{\widehat{\mathbf{d}}_s^k}\delta\widehat{\mathbf{d}}_s + \partial_{\widehat{\mathbf{d}}_s}\mathsf{FG}|_{\theta_s}\delta\widehat{\mathbf{d}}_s = 0, \tag{3.2.8}$$

$$\partial_{\widehat{\mathbf{d}}_s}\mathsf{S}|_{\theta_f}\partial_{\mathbf{u}_f}\mathcal{S}|_{{\mathbf{u}_f}^k}\delta\mathbf{u}_f + \partial_{\mathbf{u}_f}\mathsf{S}|_{\theta_f}\delta\mathbf{u}_f = 0, \tag{3.2.9}$$

where the unknown variables are $\partial_{\widehat{\mathbf{d}}_s}\mathcal{F}|_{\widehat{\mathbf{d}}_s^k}\delta\widehat{\mathbf{d}}_s$ and $\partial_{\mathbf{u}_f}\mathcal{S}|_{\mathbf{u}_f{}^k}\delta\mathbf{u}_f$. Formally inverting these problems we obtain a representation of the Newton method (3.2.7) in terms of the Jacobians of the fluid and solid problems (instead of the solver operators):

$$\left[I + \left(\partial_{\widehat{\mathbf{d}}_s}\mathsf{S}|_{\theta_f}\right)^{-1}\partial_{\mathbf{u}_f}\mathsf{S}|_{\theta_f}\left(\partial_{\mathbf{u}_f}\mathsf{FG}|_{\theta_s}\right)^{-1}\partial_{\widehat{\mathbf{d}}_s}\mathsf{FG}|_{\theta_s}\right]\delta\widehat{\mathbf{d}}_s = -\widehat{\mathbf{d}}_s^k + \mathcal{S}(\mathcal{F}(\widehat{\mathbf{d}}_s^k)). \tag{3.2.10}$$

This Jacobian is available if the Jacobians of the subproblems S and FG are.

Remark 3.2.1. *The residual evaluation involves the evaluation of the fluid and solid solver operators (which are used also for the computation of the derivatives $\partial_{\widehat{\mathbf{d}}_s}\mathsf{S}$ and $\partial_{\mathbf{u}_f}\mathsf{FG}$ in θ_f and θ_s). This means that each nonlinear subdomain problem needs to be solved at each Newton iteration. In our case however for the fluid problem the domain is fixed (since it depends only on the structure displacement), thus the only nonlinearity for the fluid comes from the convective term, which can be treated e.g. using* (2.4.4) *for Reynolds numbers which are not too large. The structure equation can be linear or not, depending on the constitutive law chosen. We remark also that if a nested Newton method is used, then the derivative terms computed at the last iteration can be reused in the computation of the Jacobian problem* (3.2.10).

Remark 3.2.2. *We can notice that the shape derivatives are embedded in the cross derivatives $\partial_{\widehat{\mathbf{d}}_s}\mathsf{FG}|_{\theta_s}$ and $\partial_{\mathbf{u}_f}\mathsf{S}|_{\theta_f}$ of* (3.2.8) *and* (3.2.9)*, thus they go to the right hand side when solving these linear problems. This is not the case when dealing with* (3.1.2).

3.2.2 Steklov–Poincaré Formulation

Another possible fixed point formulation is discussed in the remaining part of this section. Since now we consider the interface displacement separately, for convenience we express the dependence of the problems on the variables $\mathbf{u}_f, \eta, \widehat{\mathbf{d}}_s$, where $\eta = \widehat{\mathbf{d}}_{s\Gamma}$ is the FS interface displacement.

We suppose to apply a Dirichlet interface condition, $\mathbf{u}\circ\mathcal{A}_t = \eta$ and $\partial_t\widehat{\mathbf{d}}_s = \eta$ to both the FG and S problems respectively, and denote them FG_D and S_D. To close the system the stress continuity (2.1.2) equation is expressed by a further *interface problem* that we call N. We have the following system of equations

$$\begin{aligned}\mathsf{FG}_D(\mathbf{u}_f,\eta) &= 0,\\ \mathsf{S}_D(\widehat{\mathbf{d}}_s,\eta) &= 0,\\ \mathsf{N}(\mathbf{u}_f,\widehat{\mathbf{d}}_s,\eta) &= 0.\end{aligned} \tag{3.2.11}$$

The variable η introduced corresponds to the interface displacement, however we prefer not to denote it with $\widehat{\mathbf{d}}_{s\Gamma}$ or $\widehat{\mathbf{d}}_{f\Gamma}$, indeed it represents a Lagrange multiplier to enforce the stress continuity constraint.

We can define the three following solver operators

$$\begin{aligned}\mathcal{F}:&\quad U^{\widehat{\Gamma}} &&\to\quad U^f,\\ \mathcal{S}:&\quad U^{\widehat{\Gamma}} &&\to\quad U^s,\\ \mathcal{N}:&\quad U^f\times U^s &&\to\quad U^{\widehat{\Gamma}},\end{aligned}$$

such that

$$\mathsf{FG}_D(\mathcal{F}(\eta), \eta) = 0, \tag{3.2.12}$$

$$\mathsf{S}_D(\mathcal{S}(\eta), \eta) = 0, \tag{3.2.13}$$

$$\mathsf{N}(\mathbf{u}_f, \widehat{\mathbf{d}}_s, \mathcal{N}(\mathbf{u}_f, \widehat{\mathbf{d}}_s)) = 0. \tag{3.2.14}$$

As in the previous case, these operators are well defined if the three problems are individually well-posed.

From (3.2.12), (3.2.13) and (3.2.14) we obtain the equations

$$\begin{cases} \mathbf{u}_f - \mathcal{F}(\eta) & = 0 \\ \widehat{\mathbf{d}}_s - \mathcal{S}(\eta) & = 0 \\ \eta - \mathcal{N}(\mathbf{u}_f, \widehat{\mathbf{d}}_s) & = 0 \end{cases}. \tag{3.2.15}$$

The equation

$$\eta - \mathcal{N}(\mathcal{F}(\eta), \mathsf{S}(\eta)) = 0, \tag{3.2.16}$$

is yet another fixed point formulation of the problem (3.2.4). The advantage of this formulation is that the solutions of the FG_D and S_D problems can be computed in parallel. As the previous formulation (3.2.16) can be solved by means of relaxed fixed point iterations,

$$\eta^{k+1} = \omega\mathcal{N}(\mathcal{F}(\eta^k), \mathcal{S}(\eta^k)) + (1-\omega)\eta^k. \tag{3.2.17}$$

An alternative to the fixed point algorithm to solve the nonlinear problem (3.2.16) can be Newton method, as it will be discussed next.

It is possible to rewrite equation (3.2.16) in an equivalent form separating the contributions from the fluid and solid problems as

$$\mathsf{N}(\mathcal{F}(\eta), \mathcal{S}(\eta), \eta) = \Sigma_f(\eta) + \Sigma_s(\eta) = 0, \tag{3.2.18}$$

where the operators Σ_s and Σ_f are the *Steklov–Poincaré* operators [DDQ06]

$$\begin{array}{rccc} \Sigma_f: & U^{\widehat{\Gamma}} & \to & U^{\widehat{\Gamma}} \\ & \eta & \mapsto & \widehat{\sigma}_f \widehat{\mathbf{n}}^f \\ \Sigma_s: & U^{\widehat{\Gamma}} & \to & U^{\widehat{\Gamma}} \\ & \eta & \mapsto & \boldsymbol{\Pi}\widehat{\mathbf{n}}^s. \end{array} \tag{3.2.19}$$

Notice that by formally applying ${\Sigma_s}^{-1}$ to equation (3.2.18) we have

$$\eta + {\Sigma_s}^{-1}(\Sigma_f(\eta)) = 0, \tag{3.2.20}$$

which subtracted to (3.2.16) gives

$${\Sigma_s}^{-1}(\Sigma_f(\eta)) = \mathcal{N}(\mathcal{F}(\eta), \mathcal{S}(\eta)). \tag{3.2.21}$$

This equation establishes the relation between the Steklov–Poincaré operators and the solver operators.

This formulation is extensively described in Deparis et al. [DDQ06] for FSI applications, where many choices for the preconditioner of the linearized system are proposed, while in [DDFQ06] numerical simulations and comparisons are presented.

The same considerations made for the Dirichlet–Neumann scheme apply in this case: one could think of substituting the Dirichlet condition on the problems FG_D and S_D with Neumann ones (FG_N, S_N). In this case the Lagrange multiplier λ would have a similar meaning as in (2.7.10) (i.e., it would represent the traction vector $\lambda = \widehat{\sigma}_f \widehat{\mathbf{n}}^f = -\boldsymbol{\Pi}\widehat{\mathbf{n}}^s$).

Remark 3.2.3. *In this section we used the fluid–geometry problem operator FG instead of F, thus the development is independent of the use of an ALE formulation for the fluid problem. Notice also that the implementation of the fixed point methods presented allows the use of fluid and solid solvers as black boxes. This feature is called* modularity*.*

We concentrate now on the Newton method applied to (3.2.18). In particular we report a derivation for the Jacobian of (3.2.16) obtained by generalizing the steps performed in [FM05] (and reported in the previous subsection).

We proceed in the derivation above by replacing (3.2.5) by (3.2.16). The essential difference is the nature of the variable considered in the two fixed point problems; $\widehat{\mathbf{d}}_s$ lives in $\widehat{\Omega}^s$ while η lives on the interface $\widehat{\Gamma}$. We take the derivatives of equation (3.2.14). Given the starting solution $\theta_\Gamma \in U^{\widehat{\Gamma}}$ we define $\theta = (\mathcal{F}(\theta_\Gamma), \mathcal{S}(\theta_\Gamma), \theta_\Gamma) \in U^f \times U^s \times U^{\widehat{\Gamma}}$ We have

$$D_{\mathbf{u}_f}\mathsf{N}|_\theta \delta\mathbf{u}_f = \partial_{\mathbf{u}_f}\mathsf{N}|_\theta \delta\mathbf{u}_f + \partial_\eta \mathsf{N}|_\theta (\partial_{\mathbf{u}_f}\mathcal{N}|_{(\mathcal{F}(\theta_\Gamma),\mathcal{S}(\theta_\Gamma))}\delta\mathbf{u}_f) = 0,$$

$$D_{\widehat{\mathbf{d}}_s}\mathsf{N}|_\theta \delta\widehat{\mathbf{d}}_s = \partial_{\widehat{\mathbf{d}}_s}\mathsf{N}\delta\widehat{\mathbf{d}}_s + \partial_\eta \mathsf{N}|_{\theta_\Gamma} (\partial_{\widehat{\mathbf{d}}_s}\mathcal{N}|_{(\mathcal{F}|_{\theta_\Gamma},\mathcal{S}(\theta_\Gamma))}\delta\widehat{\mathbf{d}}_s) = 0.$$

We can then write formally the derivatives of the solver operator $\mathcal{N}$ in terms of the derivatives of the problem operator N:

$$\partial_{\mathbf{u}_f}\mathcal{N}|_{(\mathcal{F}|_{\theta_\Gamma},\mathcal{S}|_{\theta_\Gamma})}\delta\mathbf{u}_f = (\partial_\eta \mathsf{N}|_\theta)^{-1}\left(-\partial_{\mathbf{u}_f}\mathsf{N}|_\theta\right)\delta\mathbf{u}_f,$$

$$\partial_{\widehat{\mathbf{d}}_s}\mathcal{N}|_{(\mathcal{F}|_{\theta_\Gamma},\mathcal{S}|\theta_\Gamma)}\delta\widehat{\mathbf{d}}_s = (\partial_\eta \mathsf{N}|_\theta)^{-1}\left(-\partial_{\widehat{\mathbf{d}}_s}\mathsf{N}|_\theta\right)\delta\widehat{\mathbf{d}}_s.$$

Notice that the partial derivatives of the problem operator N are usually easy to compute in closed form. They consist in the derivatives of the stress continuity coupling condition.

Remark 3.2.4. *As pointed out in remark 3.2.1, also here the evaluation of the operator N (and of its derivatives) in $\theta = (\mathcal{F}(\theta_\Gamma), \mathcal{S}(\theta_\Gamma), \theta_\Gamma)$ is crucial. As in the previous case we can have two nested Newton iterations if the fluid and solid equations are nonlinear. However again the geometrical nonlinearity of the fluid equation is not accounted for by the fluid problem FG_D, in fact in the fluid solver operator $\mathcal{F}(\theta_\Gamma)$ the geometry is fixed by the interface Dirichlet condition for the domain displacement $\widehat{\mathbf{d}}_{f\Gamma} = \theta_\Gamma$.*

We can express the Jacobian system of (3.2.16) at θ_Γ as

$$J|_{\theta_\Gamma}\delta\eta = (I - \partial_{\mathbf{u}_f}\mathcal{N}|_{(\mathcal{F}|_{\theta_\Gamma},\mathcal{S}|_{\theta_\Gamma})}\partial_\eta\mathcal{F}|_{\theta_\Gamma} - \partial_{\widehat{\mathbf{d}}_s}\mathcal{N}|_{(\mathcal{F}|_{\theta_\Gamma},\mathcal{S}|_{\theta_\Gamma})}\partial_\eta\mathcal{S}|_{\theta_\Gamma})\delta\eta, \tag{3.2.22}$$

$$J|_{\theta_\Gamma}\delta\eta = \delta\eta - (\partial_\eta \mathsf{N}|_\theta)^{-1}\left(-\partial_{\widehat{\mathbf{d}}_s}\mathsf{N}|_\theta\partial_\eta\mathcal{S}|_{\theta_\Gamma}\right)\delta\eta - (\partial_\eta \mathsf{N}|_\theta)^{-1}\left(-\partial_{\mathbf{u}_f}\mathsf{N}|_\theta\partial_\eta\mathcal{F}|_{\theta_\Gamma}\right)\delta\eta. \tag{3.2.23}$$

Notice that now the Jacobian is split in two parts containing the Jacobian of the two physical problems. To simplify this expression we can apply on both sides the linear operator $\partial_\eta \mathsf{N}|_\theta$

$$\partial_\eta \mathsf{N}|_\theta J|_{\theta_\Gamma}\delta\eta = \partial_\eta \mathsf{N}|_\theta\delta\eta - \left(-\partial_{\widehat{\mathbf{d}}_s}\mathsf{N}|_\theta\partial_\eta\mathcal{S}|_{\theta_\Gamma}\right)\delta\eta - \left(-\partial_{\mathbf{u}_f}\mathsf{N}|_\theta\partial_\eta\mathcal{F}|_{\theta_\Gamma}\right)\delta\eta. \tag{3.2.24}$$

This expression still contains the derivatives of the solver operators $\mathcal{S}$ and $\mathcal{F}$. To rewrite it only in terms of problem operators some further manipulation is required. Recalling the equations for the solver operators we can express their derivatives (as done in the previous case) in terms of the Jacobians of the problem operators FG_D, S_D and N. In fact, the derivative with respect to η, e.g. of (3.2.12), reads

$$\left(\partial_{\mathbf{u}_f}\mathsf{FG}_D|_\theta\right)\partial_\eta\mathcal{F}|_{\theta_\Gamma}\delta\eta = -\partial_\eta\mathsf{FG}_D|_\theta\delta\eta, \tag{3.2.25}$$

which is a linear system in the unknown $\partial_\eta\mathcal{F}|_{\theta_\Gamma}\delta\eta$. The same expression can be written for the solid operator. Thus the evaluation of the solver operator derivatives involves the solution of two linear systems. We write formally

$$\partial_\eta\mathcal{F}|_{\theta_\Gamma} = -\left(\partial_{\mathbf{u}_f}\mathsf{FG}_D|_\theta\right)^{-1}\partial_\eta\mathsf{FG}_D|_\theta, \tag{3.2.26}$$

$$\partial_\eta\mathcal{S}|_{\theta_\Gamma} = -\left(\partial_{\widehat{\mathbf{d}}_s}\mathsf{S}_D|_\theta\right)^{-1}\partial_\eta\mathsf{S}_D|_\theta. \tag{3.2.27}$$

One eventually obtains substituting in (3.2.24)

$$\begin{aligned}\partial_\eta\mathsf{N}|_\theta J|_{\theta_\Gamma}\delta\eta =& \partial_\eta\mathsf{N}|_\theta\delta\eta - \partial_{\widehat{\mathbf{d}}_s}\mathsf{N}|_\theta\left(\left(\partial_{\widehat{\mathbf{d}}_s}\mathsf{S}_D|_\theta\right)^{-1}\partial_\eta\mathsf{S}_D|_\theta\right)\delta\eta + \\ &- \partial_{\mathbf{u}_f}\mathsf{N}|_\theta\left(\left(\partial_{\mathbf{u}_f}\mathsf{FG}_D|_\theta\right)^{-1}\partial_\eta\mathsf{FG}_D|_\theta\right)\delta\eta.\end{aligned} \tag{3.2.28}$$

Remark 3.2.5. *Also in this case the shape derivatives are taken into account in the term $\partial_\eta\mathsf{FG}_D|_\theta$ of* (3.2.26)*, thus they go to the right hand side of this linear system.*

Remark 3.2.6. *Notice that this derivation is formally the same if we assign the stress continuity condition to both the fluid and solid problems and solve separately the velocity continuity equation. In this way we obtain two Neumann problems FG_N and S_N for the fluid and for the structure.*

For the considerations made in Section §2.7 we can write the variational form of the equation (3.2.18) as

$$< \Sigma_f(\eta), \mathbf{v}_{\widehat{\Gamma}} >_{U^{\widehat{\Gamma}}} + < \Sigma_s(\eta), \mathbf{v}_{\widehat{\Gamma}} >_{U^{\widehat{\Gamma}}} = 0 \ \ \forall \mathbf{v}_{\widehat{\Gamma}} \in U^{\widehat{\Gamma}},$$

where $< \cdot, \cdot >$ represents the usual inner product in L^2. This can be rewritten as

$$< \mathcal{R}_f, \mathcal{L}_f\mathbf{v}_{\widehat{\Gamma}} >_{U^f} + < \mathcal{R}_s, \mathcal{L}_s\mathbf{v}_{\widehat{\Gamma}} >_{U^s} = 0 \ \ \forall \mathbf{v}_{\widehat{\Gamma}} \in U^{\widehat{\Gamma}},$$

where $\mathcal{R}_f$ and $\mathcal{R}_s$ are the residuals of the momentum equations of the fluid and the solid problems, while $\mathcal{L}_f$ and $\mathcal{L}_s$ are the continuous lift operators introduced in Section §2.7. Thus we can define the Steklov–Poincaré operators as the trace of the residuals of the two momentum equations

$$\begin{array}{rccc} \Sigma_f: & U^{\widehat{\Gamma}} & \to & U^{\widehat{\Gamma}} \\ & \eta & \mapsto & \mathrm{tr}_{\widehat{\Gamma}}\left(\mathcal{R}_f|_{(\mathcal{F}(\eta),\eta)}\right) \\ \Sigma_s: & U^{\widehat{\Gamma}} & \to & U^{\widehat{\Gamma}} \\ & \eta & \mapsto & \mathrm{tr}_{\widehat{\Gamma}}\left(\mathcal{R}_s|_{(\mathcal{S}(\eta),\eta)}\right). \end{array} \tag{3.2.29}$$

The Jacobian reads, by the chain rule and using the expression for the derivatives of the solver operators (3.2.26) and (3.2.27),

$$\begin{aligned} J^{sp}_{\theta_\Gamma}\delta\eta &= \left[D_\eta \Sigma_s|_{\theta_\Gamma}\right]\delta\eta + \left[D_\eta \Sigma_f|_{\theta_\Gamma}\right]\delta\eta = \\ &= \left[\mathrm{tr}_{\widehat{\Gamma}}\left(\partial_\eta \mathcal{R}_s|_{(\mathcal{S}(\theta_\Gamma),\theta_\Gamma)}\right) - \mathrm{tr}_{\widehat{\Gamma}}\left(\partial_{\widehat{\mathbf{d}}_s}\mathcal{R}_s|_{(\mathcal{S}(\theta_\Gamma),\theta_\Gamma)}\left(\partial_{\widehat{\mathbf{d}}_s}\mathsf{S}\right)^{-1}\partial_\eta \mathsf{S}\right)\right]\delta\eta + \\ &\quad \left[\mathrm{tr}_{\widehat{\Gamma}}\left(\partial_\eta \mathcal{R}_f|_{(\mathcal{F}(\theta_\Gamma),\theta_\Gamma)}\right) - \mathrm{tr}_{\widehat{\Gamma}}\left(\partial_{\mathbf{u}_f}\mathcal{R}_f|_{(\mathcal{F}(\theta_\Gamma),\theta_\Gamma)}\left(\partial_{\mathbf{u}_f}\mathsf{FG}\right)^{-1}\partial_\eta \mathsf{FG}\right)\right]\delta\eta. \end{aligned} \tag{3.2.30}$$

Recalling that in weak (formal) sense $\mathsf{N} = \mathrm{tr}_{\widehat{\Gamma}}\left(\mathcal{R}_f + \mathcal{R}_s\right) = \Sigma_s + \Sigma_f$, we see that the right-hand side of equation (3.2.30) is the same as that in (3.2.28). Thus the following relation holds

$$J^{sp}_{\theta_\Gamma} = \partial_\eta \mathsf{N}|_\theta J|_{\theta_\Gamma}.$$

3.3 Newton Method for Geometrical Multiscale

In this section we introduce a framework to describe the coupling between 3D FSI and reduced order models (cf. Section §2.9). The method used to describe the coupled system is a generalization of the one used in Section §3.2 and the representation of the Newton algorithm can be derived in a similar way.

We assume in the following without loss of generality that the flux continuity is assigned to the 3D problem, while the mean pressure continuity is assigned to the 1D. We can then define the solver operators for the 3D and 1D problems (in a similar way as for the fluid and structure solver operators used in Section §3.2)

$$\begin{array}{rccc} \mathcal{F}_{3D}: & \mathbb{R} & \to & \mathbb{R} \\ & Q & \mapsto & p \\ \mathcal{F}_{1D}: & \mathbb{R} & \to & \mathbb{R} \\ & p & \mapsto & Q, \end{array} \tag{3.3.1}$$

where p and Q are respectively the mean pressure and the flux over the 3D-1D interface. These operators are such that $\mathcal{F}_{3D}(Q)$ involves the solution of the 3D FSI model with defective flux boundary condition on the boundary coupled with the 1D model, while $\mathcal{F}_{1D}(p)$ involves the solution of the 1D model with pressure p imposed on the same boundary. These operators are well defined for both the 1D and 3D models as far as the relative problems with the corresponding boundary condition are well posed.

The values Q and p at the coupling interface are the solution of the system

$$Q - \mathcal{F}_{1D}(p) = 0 \tag{3.3.2}$$

$$p - \mathcal{F}_{3D}(Q) = 0. \tag{3.3.3}$$

The Newton method can be used to solve this system. It consists in several iterations of the type

$$\begin{pmatrix} Q_{k+1} \\ p_{k+1} \end{pmatrix} = \begin{pmatrix} Q_k \\ p_k \end{pmatrix} + \begin{pmatrix} \delta Q \\ \delta p \end{pmatrix}$$

where the vector $(\delta Q, \delta p)^T$ is the solution of the Jacobian system

$$\begin{pmatrix} 1 & \partial_p \mathcal{F}_{1D}(p_k) \\ \partial_Q \mathcal{F}_{3D}(Q_k) & 1 \end{pmatrix} \begin{pmatrix} \delta Q \\ \delta p \end{pmatrix} = - \begin{pmatrix} Q_k - \mathcal{F}_{1D}(p_k) \\ p_k - \mathcal{F}_{3D}(Q_k) \end{pmatrix}. \tag{3.3.4}$$

As in Section §3.2 the derivatives in (3.3.4) can be computed as functions of the derivatives of the *problem operators*, which in this case are the full 3D FSI operator and the 1D model operator. Since Remark 3.2.4 applies also to this case we need to solve the nonlinearity on the two problems separately, then we can use the last Jacobian matrix computed for both the problems to assemble the Jacobian of (3.3.4).

For each Newton iteration we need thus a solution of the two problems only to obtain the residual, and then the solution of the Jacobian linear system. However, how it is shown in the previous sections, the evaluation of the derivative of each solver operator implies the solution of the linear Jacobian system for each one of the subproblems. This as previously mentioned can be done in a relatively cheap way by reusing the last Jacobian matrix (and preconditioner) calculated during the residual evaluation. The generalization of the previous derivation to several subdomains is straightforward, details can be found e.g. in Malossi et al. [MBDQ11].

Remark 3.3.1. *Notice that if the time discretization for both the 1D and 3D models is such that the space discretized systems are linear, then equations* (3.3.2) *are linear too, and the only iterations required are those to solve the* 2×2 *system* (3.3.4).

Remark 3.3.2. *We remark also that a Newton scheme could be devised, in a similar way as in the previous sections, for a fixed point formulation of the problem like*

$$Q - \mathcal{F}_{1D}(\mathcal{F}_{3D}(Q)) = 0. \tag{3.3.5}$$

The newton kth iteration reads, taking $p^k = \mathcal{F}_{3D}(Q^k)$,

$$(1 - \partial_p \mathcal{F}_{1D}|_{p^k} \partial_Q \mathcal{F}_{3D}|_{Q^k}) \delta Q = -Q^k + \mathcal{F}_{1D}(\mathcal{F}_{3D}(Q^k)).$$

The derivatives of the solver operators can be computed as in the previous case. This formulation has the advantage that the dimension of the Jacobian system is reduced to one, however the disadvantage is that the two solutions needed for the residual evaluation cannot be computed in parallel.

3.4 Computing the Exact Jacobian Matrix

We devote this section to the derivation of the Jacobian matrix for the Newton method in FSI. We describe in detail the form of the derivatives, with particular attention to the shape derivatives, i.e., the cross derivatives of the fluid problem with respect to the domain motion. Often in literature this derivation is omitted because it is quite technical, for this reason we felt the need to describe the steps leading to the Jacobian representation more in detail than other parts of the dissertation, for which detailed explanations can be found in literature. Furthermore, the derivation of the Jacobian terms are quite different, depending on the authors. We refer here mainly to the approach followed in [FM05, FFT00]; other references are e.g. [BC10a, BCHZ08].

We do not discuss in this chapter the Fréchet derivatives of the structure equation. In fact, although they can be more or less complex depending on the constitutive law, their derivation is rather standard thanks to the Lagrangian formulation.

The derivative of the convective term in the fluid momentum conservation equation with respect to the fluid velocity is straightforward, namely

$$D_{\mathbf{u}}\left[\int_{\Omega_t^f} \frac{\rho_f}{\delta t}\left((\mathbf{u}-\mathbf{w})\cdot\nabla_x\right)\mathbf{u}\cdot\mathbf{v}_f \;\; d{\Omega_t}^f\right]\delta\mathbf{u} =$$

$$= \int_{\Omega_t^f} \frac{\rho_f}{\delta t}\left[\left((\mathbf{u}-\mathbf{w})\cdot\nabla_x\right)\delta\mathbf{u}\cdot\mathbf{v}_f + \left(\delta\mathbf{u}\cdot\nabla_x\right)\mathbf{u}\cdot\mathbf{v}_f\right] \;\; d{\Omega_t}^f.$$

As anticipated the geometrical nonlinearity is more difficult to handle. In the next subsection we illustrate some technical tools that are necessary in the derivation described in Section §3.4.2.

3.4.1 Shape Derivatives of Domain Functionals

In this subsection we introduce an expression for the shape derivatives of generic integral functions. The derivatives in this form are used in Fernández and Moubachir [FM05] in the FSI context. A similar derivation is reported in Fanion et al. [FFT00], where the fluid equations in ALE form are linearized under the hypothesis of steady state and they are solved on a fixed grid, coupled with the structure through transpiration conditions. We follow Sokolowski and Zolesio [SZ92], adapting the notations to the framework introduced in the previous chapters.

Given an open set $D \subset \mathbb{R}^d$, with $d = 2, 3$, and two open sets $\widehat{\Omega}, \Omega_t \subset D$ we can define a diffeomorphism $\mathcal{A} : \widehat{\Omega} \times T \to \Omega_t$ as done in Section §1.2 between the two domains. The parameter t can represent time or any other scalar parameter the domain Ω_t depends on. For simplicity and consistency with the previous chapters we consider it to be the time, while the map $\mathcal{A}$ represents the ALE map or the motion map.

We recall the definition of the domain velocity $\mathbf{w}$ (written in the Lagrangian frame)

$$\partial_t \mathcal{A}(t,\widehat{x}) = \widehat{\mathbf{w}}(t,\widehat{x}) = \mathbf{w}(t,\mathcal{A}(t,\widehat{x})). \tag{3.4.1}$$

In this section for the sake of notations we express explicitly the dependence of the domain velocity on time. In the following we abbreviate the spatial derivatives with respect to the referential coordinates $\frac{\partial}{\partial \widehat{x}_i}$ with ∂_i.

We suppose that the admissible deformations $\mathcal{A}$ of the domain $\Omega_t = \mathcal{A}(t,\widehat{\Omega})$ satisfy the conditions $\mathcal{A}(t,\widehat{\Omega}) \in C^1(T; C^k(\widehat{\Omega}))$, $\mathcal{A}^{-1}(t,\Omega_t) \in C^1(T; C^k(\Omega_t))$, where k is the regularity of the coordinate functions describing the parametrization of the reference domain (see (1.1.1)). We recall from Lemma 1.1.1 that the change of measure for the volume in Ω_t involves the Jacobian determinant $J_{\mathcal{A}} = \det(\nabla_{\widehat{x}}\mathcal{A})$.

Let us consider first an infinitesimal transformation $\mathcal{A}(\epsilon,\cdot)$ and $d = 2$, so that $\mathcal{A}(\epsilon,\cdot) = (\mathcal{A}_1(\epsilon,\cdot), \mathcal{A}_2(\epsilon,\cdot))^T$. Then the Jacobian determinant may be written as

$$J_{\mathcal{A}}(\epsilon,\widehat{x}) = \partial_1\mathcal{A}_1(\epsilon,\widehat{x})\partial_2\mathcal{A}_2(\epsilon,\widehat{x}) - \partial_1\mathcal{A}_2(\epsilon,\widehat{x})\partial_2\mathcal{A}_1(\epsilon,\widehat{x}),$$

and its derivative with respect to t reads

$$\partial_t J_{\mathcal{A}}|_{(0,\widehat{x})} = \partial_1(\partial_t \mathcal{A}_1|_{(0,\widehat{x})})\partial_2\mathcal{A}_2(0,\widehat{x}) + \partial_1(\mathcal{A}_1(0,\widehat{x}))\partial_2(\partial_t\mathcal{A}_2|_{(0,\widehat{x})}) + \\ - \partial_1(\partial_t\mathcal{A}_2|_{(0,\widehat{x})})\partial_2\mathcal{A}_1(0,\widehat{x}) - \partial_1\mathcal{A}_2(0,\widehat{x})\partial_2(\partial_t\mathcal{A}_1|_{(0,\widehat{x})}). \quad (3.4.2)$$

Using (3.4.1) we get

$$\partial_t J_{\mathcal{A}}|_{(0,\widehat{x})} = \partial_1\widehat{\mathbf{w}}_1(0)\partial_2\mathcal{A}_2(0,\widehat{x}) + \partial_1\mathcal{A}_1(0,\widehat{x})\partial_2\widehat{\mathbf{w}}_2(0) - \partial_1\widehat{\mathbf{w}}_2(0)\partial_2\mathcal{A}_1(0,\widehat{x}) - \partial_1\mathcal{A}_2(0,\widehat{x})\partial_2\widehat{\mathbf{w}}_1(0).$$

The two last terms are infinitesimal because for $(\epsilon \to 0)$ also $\mathcal{A}(\epsilon,\cdot) \to I(\cdot)$ and thus $\partial_1\mathcal{A}_2(\epsilon,\cdot) \to \partial_1 0 = 0$ while e.g. $\partial_1\mathcal{A}_1(\epsilon,\cdot) \to \partial_1\widehat{x}_1 = 1$. Thus

$$\partial_t J_{\mathcal{A}}(0,\widehat{x}) = \nabla_{\widehat{x}}\cdot\widehat{\mathbf{w}}(0). \quad (3.4.3)$$

This result extends easily to $d = 3$.

We introduce the domain at time $t+\epsilon$

$$\Omega_{t+\epsilon} = \mathcal{A}(t+\epsilon,\widehat{\Omega}),$$

and the map $\mathcal{A}_\epsilon^t : \Omega_t \to \Omega_{t+\epsilon}$ such that

$$\mathcal{A}_\epsilon^t(\mathcal{A}(t,\widehat{x})) = \mathcal{A}(t+\epsilon,\widehat{x}).$$

The Lagrangian derivative of $\mathcal{A}_{t+\epsilon}$ reads

$$D\mathcal{A}(t+\epsilon,\widehat{x}) = \nabla_x\mathcal{A}_\epsilon^t(x)D\mathcal{A}(t,\widehat{x}), \quad (3.4.4)$$

where $x = \mathcal{A}(t,\widehat{x})$. The Lagrangian "time" derivative of the Jacobian matrix evaluated in $t = \bar{t}$ becomes, using the definition of derivative, the chain rule, and (3.4.1),

$$D_t\nabla_{\widehat{x}}\mathcal{A}|_{(\bar{t},\widehat{x})} = \lim_{\epsilon\to 0}\frac{1}{\epsilon}(\nabla_{\widehat{x}}\mathcal{A}(\bar{t}+\epsilon,\widehat{x}) - \nabla_{\widehat{x}}\mathcal{A}(\bar{t},\widehat{x})) = \lim_{\epsilon\to 0}\frac{1}{\epsilon}((\nabla_x\mathcal{A}_\epsilon^t(x)\nabla_{\widehat{x}}\mathcal{A}(\bar{t},\widehat{x})) - \nabla_{\widehat{x}}\mathcal{A}(\bar{t},\widehat{x})) = \\ = \lim_{\epsilon\to 0}\frac{1}{\epsilon}(\nabla_x\mathcal{A}_\epsilon^t(x) - I)\nabla_{\widehat{x}}\mathcal{A}(\bar{t},\widehat{x}) = \nabla_x\mathbf{w}(0)\nabla_{\widehat{x}}\mathcal{A}(\bar{t},\widehat{x}). \quad (3.4.5)$$

Also, with the same computation, using the chain rule and (3.4.3) we get the determinant derivative formula (1.2.5): we have, using Binet's theorem

$$D_t J_{\mathcal{A}}|_{(\bar{t},\widehat{x})} = \lim_{\epsilon\to 0}\frac{1}{\epsilon}(\det(\nabla_{\widehat{x}}\mathcal{A}(\bar{t}+\epsilon,\widehat{x})) - \det(\nabla_{\widehat{x}}\mathcal{A}(\bar{t},\widehat{x}))) = \\ = \lim_{\epsilon\to 0}\frac{1}{\epsilon}(\det(\nabla_x\mathcal{A}_\epsilon^t(x)\nabla_{\widehat{x}}\mathcal{A}(\bar{t},\widehat{x})) - \det(\nabla_{\widehat{x}}\mathcal{A}(\bar{t},\widehat{x}))) = \\ = \lim_{\epsilon\to 0}\frac{1}{\epsilon}(\det(\nabla_x\mathcal{A}_\epsilon^t(x)) - 1)\det(\nabla_{\widehat{x}}\mathcal{A}(\bar{t},\widehat{x})). \quad (3.4.6)$$

Now, remembering (3.4.3) we have

$$D_t J_{\mathcal{A}}|_{(\bar{t},\widehat{x})} = \lim_{\epsilon\to 0}\partial_t(\det(\nabla_x\mathcal{A}_\epsilon^t(x)))J_{\mathcal{A}}(\bar{t},\widehat{x}) = \nabla_x\cdot\mathbf{w}(0)J_{\mathcal{A}}(\bar{t},\widehat{x}). \quad (3.4.7)$$

Using the formulas derived up to now we are able to compute the Lagrangian derivative of a domain functional:

$$D_t\int_{\Omega_t} f\,d\Omega_t = D_t\int_{\widehat{\Omega}}\widehat{f}J_{\mathcal{A}}\,d\widehat{\Omega} = \int_{\widehat{\Omega}}[D_t\widehat{f}J_{\mathcal{A}} + \widehat{f}D_tJ_{\mathcal{A}}]\,d\widehat{\Omega} = \int_{\Omega_t}[D_tf + f\nabla_x\cdot(\mathbf{w})]\,d\Omega_t,$$

which corresponds to the Reynolds or Leibnitz Theorems 1.2.1 and 1.2.2.

We show now how to compute the derivative of an integral on the boundary γ of the domain Ω_t. Recalling the local coordinate system introduced in Section §1.1, using the cofactor matrix $\text{cof}\,(\nabla_\xi \mathbf{h}) = \det(\nabla_\xi \mathbf{h})(\nabla_\xi \mathbf{h})^{-T}$ we can write, from Nanson's formula derived in Proposition 1.1.2,

$$\int_{\widehat{\gamma}} f \, d\widehat{\gamma} = \int_{B_0 \subset \mathbb{R}^{n-1}} f \circ \mathbf{h} ||\text{cof}\,(\nabla_\xi \mathbf{h}) \mathbf{e}_3|| d\xi.$$

Analogously we have in this case that

$$\int_{\gamma} f \, d\gamma = \int_{\widehat{\gamma}} f \circ \mathcal{A}_t ||\text{cof}\,(\nabla_{\widehat{x}} \mathcal{A}_t) \widehat{\mathbf{n}}|| \, d\widehat{\gamma}.$$

Let us look for simplicity at time $t = 0$ (the case $t = \bar{t}$ is analogous). We define $\omega = \text{cof}\,(\nabla_{\widehat{x}} \mathcal{A}_t)\widehat{\mathbf{n}} = J_{\mathcal{A}}(t)(\nabla_{\widehat{x}} \mathcal{A}_t)^{-T}\widehat{\mathbf{n}}$. We aim at computing the derivative of the surface measure $\|\omega\|$. From the definition of the Euclidean norm in $\mathbb{R}^3$ we easily express this derivative in terms of $D_t\|\omega\|^2$:

$$D_t\|\omega\| = \frac{1}{2\|\omega\|} D_t\|\omega\|^2. \tag{3.4.8}$$

Then we have, from the previous definitions,

$$\begin{aligned} D_t\|\omega\|^2 &= D_t[{J_{\mathcal{A}}}^2(\nabla_{\widehat{x}} \mathcal{A}_t)^{-T}\widehat{\mathbf{n}} \cdot (\nabla_{\widehat{x}} \mathcal{A}_t)^{-T}\widehat{\mathbf{n}}] = \\ &= 2J_{\mathcal{A}} D_t J_{\mathcal{A}} (\nabla_{\widehat{x}} \mathcal{A}_t)^{-1}(\nabla_{\widehat{x}} \mathcal{A}_t)^{-T}\widehat{\mathbf{n}} \cdot \widehat{\mathbf{n}} + {J_{\mathcal{A}}}^2 D_t[(\nabla_{\widehat{x}} \mathcal{A}_t)^{-1}(\nabla_{\widehat{x}} \mathcal{A}_t)^{-T}\widehat{\mathbf{n}} \cdot \widehat{\mathbf{n}}]. \end{aligned}$$

Remembering that for $t \to 0$ from (3.4.5) $D_t \nabla_{\widehat{x}} \mathcal{A} \to \nabla_{\widehat{x}} \widehat{\mathbf{w}}(0)$,

$$\begin{aligned} 2\omega \cdot D_t\omega = 2J_{\mathcal{A}} D_t J_{\mathcal{A}} (\nabla_{\widehat{x}} \mathcal{A}_t)^{-1}(\nabla_{\widehat{x}} \mathcal{A}_t)^{-T}\widehat{\mathbf{n}} \cdot \widehat{\mathbf{n}} - {J_{\mathcal{A}}}^2 (\nabla_{\widehat{x}} \mathcal{A}_t)^{-2} \nabla_{\widehat{x}} \widehat{\mathbf{w}}(0) (\nabla_{\widehat{x}} \mathcal{A}_t)^{-T}\widehat{\mathbf{n}} \cdot \widehat{\mathbf{n}} \\ - {J_{\mathcal{A}}}^2 (\nabla_{\widehat{x}} \mathcal{A}_t)^{-1}({\mathcal{A}_t}^T)^{-2} \nabla_{\widehat{x}} \widehat{\mathbf{w}}(0)^T \widehat{\mathbf{n}} \cdot \widehat{\mathbf{n}}, \end{aligned}$$

and since $\nabla_{\widehat{x}} \mathcal{A}_t \to I$ for $t \to 0$,

$$2\omega \cdot D_t\omega = 2J_{\mathcal{A}} D_t J_{\mathcal{A}} \widehat{\mathbf{n}} \cdot \widehat{\mathbf{n}} - {J_{\mathcal{A}}}^2 (\nabla_x \widehat{\mathbf{w}}(0) + \nabla_x \widehat{\mathbf{w}}(0)^T) \, \widehat{\mathbf{n}} \cdot \widehat{\mathbf{n}}.$$

Then always taking the limit, as $J_{\mathcal{A}} \to 1$ (and $\omega \to J_{\mathcal{A}}\widehat{\mathbf{n}}$), using (3.4.3)

$$2D_t\omega \cdot \widehat{\mathbf{n}} = 2\nabla_{\widehat{x}} \cdot \widehat{\mathbf{w}}(0)\widehat{\mathbf{n}} \cdot \widehat{\mathbf{n}} - [\nabla_{\widehat{x}} \widehat{\mathbf{w}}(0) + (\nabla_{\widehat{x}} \widehat{\mathbf{w}}(0))^T]\widehat{\mathbf{n}} \cdot \widehat{\mathbf{n}}.$$

Substituting in (3.4.8) we obtain

$$D_t\|\omega\| = \frac{1}{2\|\omega\|} D_t\|\omega\|^2 = \frac{2D_t\omega \cdot \widehat{\mathbf{n}}}{2} = \nabla_{\widehat{x}} \cdot \widehat{\mathbf{w}}(0) - \frac{1}{2}[\nabla_{\widehat{x}} \widehat{\mathbf{w}}(0) + (\nabla_{\widehat{x}} \widehat{\mathbf{w}}(0))^T]\widehat{\mathbf{n}} \cdot \widehat{\mathbf{n}}.$$

Fixing $t = \bar{t}$ and performing similar substitutions, considering (3.4.5) and (3.4.7), performing the derivatives in the reference configuration and then recasting the integral in the current one, yields to the following expression for the derivative of a boundary shape functional

$$D_t \int_{\gamma} f \, d\gamma = \int_{\gamma} \nabla_x f \mathbf{w}(0) + f \left\{ \nabla_x \cdot (\mathbf{w}(0)) - \frac{1}{2}[\nabla_x \mathbf{w}(0) + (\nabla_x \mathbf{w}(0))^T]\mathbf{n} \cdot \mathbf{n} \right\} d\gamma. \tag{3.4.9}$$

3.4.2 Shape derivatives in FSI-ALE

Given the function $f \in W^{2,1}(\mathbb{R}^3)$ (i.e., with first and second derivatives in $L^1(\mathbb{R}^3)$), we consider the following functional

$$\mathcal{J}(\Omega_t) = \int_{\Omega_t} f(x)\, d\Omega_t.$$

In the previous section we have shown how to compute the "time" derivative of a domain functional, i.e., using the chain rule, $D_t\left(\mathcal{J} \circ \mathcal{A}\right) = D_{\widehat{\mathbf{d}}_f}\mathcal{J} D_t\mathcal{A} = D_{\widehat{\mathbf{d}}_f}\mathcal{J}\mathbf{w}$. However to implement the Newton method we need the derivative of the domain functional with respect to the deformation $\widehat{\mathbf{d}}_f$ of the domain. Given the increment $\delta\widehat{\mathbf{d}}_f$ we need the derivative $D_{\widehat{\mathbf{d}}_f}\mathcal{J}\delta\widehat{\mathbf{d}}_f$. These derivatives (due to the chain rule) can be computed using the formulae derived in the previous section, just substituting $\mathbf{w}$ with $\delta\widehat{\mathbf{d}}_f$.

Now we can use the expressions derived above to compute all the derivatives of the variational formulation of the fluid problem equations with respect to the fluid domain displacement $\mathbf{d}_f$. We consider here the case in which the fluid momentum equation is written in conservative formulation. The fluid equations are discretized in time with implicit Euler (FI scheme). As for the rest of this chapter we omit the index referring to time to light the notation. However all the variables explicitly appearing at the right hand side in the equations are intended at a time t_{n+1}. The variational formulation for the fluid problem consists of the following integrals (cf. Section §2.7):

$$\frac{1}{\delta t}\int_{\Omega_t^f} \rho_f \mathbf{u}\cdot\mathbf{v}_f \;\; d{\Omega_t}^f \tag{3.4.10a}$$

$$+\int_{\Omega_t^f} \rho_f \nabla_x\cdot[\mathbf{u}\otimes(\mathbf{u}-\mathbf{w})]\cdot\mathbf{v}_f \tag{3.4.10b}$$

$$+\int_{\Omega_t^f} \sigma_f : \nabla_x \mathbf{v}_f \;\; d{\Omega_t}^f \tag{3.4.10c}$$

$$-\int_{\Omega_t^f} \mathbf{f}^f\cdot\mathbf{v}_f \;\; d{\Omega_t}^f \tag{3.4.10d}$$

$$+\int_{\Gamma_N^f} \sigma_f\mathbf{n}^f\cdot\mathbf{v}_f \; d\gamma = -\int_{\Gamma_N^f}\mathbf{g}_f^N\cdot\mathbf{v}_f\; d\gamma + \int_{\Omega_t^f}\mathbf{f}_f\cdot\mathbf{v}_f \;\; d{\Omega_t}^f \qquad \forall \mathbf{v}_f \in V^f \tag{3.4.10e}$$

$$\int_{\Omega_t^f}\nabla_x\cdot\mathbf{u}q = 0 \qquad \forall q \in U^p. \tag{3.4.10f}$$

The shape derivative in the $\delta\widehat{\mathbf{d}}_f$ direction of the first term (3.4.10a) is simple to compute. In fact just using the identity (3.4.7) we get

$$D_{\widehat{\mathbf{d}}_f}\left[\int_{\Omega_t^f}\frac{\rho_f}{\delta t}\mathbf{u}\cdot\mathbf{v}_f \;\; d{\Omega_t}^f\right]\delta\widehat{\mathbf{d}}_f = \int_{\widehat{\Omega}^f}\frac{\rho_f}{\delta t}\widehat{\mathbf{u}}\cdot\widehat{\mathbf{v}}_f D_{\widehat{\mathbf{d}}_f}J_{\mathcal{A}}\delta\widehat{\mathbf{d}}_f \;\; d\widehat{\Omega}^f = \int_{\widehat{\Omega}^f}\frac{\rho_f}{\delta t}\widehat{\mathbf{u}}\cdot\mathbf{v}_f J_{\mathcal{A}}\nabla_x\cdot\delta\widehat{\mathbf{d}}_f \;\; d\widehat{\Omega}^f =$$
$$= \int_{\Omega_t^f}\frac{\rho_f}{\delta t}\mathbf{u}\cdot\mathbf{v}_f\nabla_x\cdot\delta\widehat{\mathbf{d}}_f \;\; d{\Omega_t}^f.$$

We then consider the term (3.4.10f). Rewriting the integral in the reference configuration,

recalling that (using Einstein's notations)

$$\nabla_x \cdot \mathbf{u} = \frac{\partial u^i}{\partial x_i} = \frac{\partial u^i}{\partial \widehat{x}_k}\frac{\partial \widehat{x}_k}{\partial x_i} = \frac{\partial u^i}{\partial \widehat{x}_k}\left(\frac{\partial x_i}{\partial \widehat{x}_k}\right)^{-1} = \nabla_{\widehat{x}}\mathbf{u} : (\mathbf{F}^T)^{-1}, \tag{3.4.11}$$

gives

$$D_{\widehat{\mathbf{d}}_f}\left[\int_{\Omega_t^f} q\nabla_x\cdot\mathbf{u}\ \ d{\Omega_t}^f\right]\delta\widehat{\mathbf{d}}_f = D_{\widehat{\mathbf{d}}_f}\left[\int_{\widehat{\Omega}^f} q\nabla_{\widehat{x}}\widehat{\mathbf{u}} : \mathbf{F}^{-T}J_{\mathcal{A}}\ \ d\widehat{\Omega}^f\right]\delta\widehat{\mathbf{d}}_f =$$
$$= \int_{\widehat{\Omega}^f} q\nabla_{\widehat{x}}\widehat{\mathbf{u}} : D_{\widehat{\mathbf{d}}_f}\mathbf{F}^{-T}\delta\widehat{\mathbf{d}}_f J_{\mathcal{A}} + q\nabla_{\widehat{x}}\widehat{\mathbf{u}} : \mathbf{F}^{-T}D_{\widehat{\mathbf{d}}_f}J_{\mathcal{A}}\delta\widehat{\mathbf{d}}_f\ \ d\widehat{\Omega}^f. \tag{3.4.12}$$

Now we use the identities derived in the previous section, namely

$$D_t J_{\mathcal{A}} = \nabla_x\cdot\mathbf{w}J_{\mathcal{A}},$$

and

$$D_t\mathbf{F}^{-T} = -(\mathbf{F}^{-T})^2 D_t\mathbf{F}^T = -(\mathbf{F}^{-T})^2(D_t\mathbf{F})^T = -(\mathbf{F}^{-T})^2\left(\nabla_x\mathbf{w}\mathbf{F}\right)^T = -\mathbf{F}^{-T}(\nabla_x\mathbf{w})^T. \tag{3.4.13}$$

Replacing $\mathbf{w}$ with $\delta\widehat{\mathbf{d}}_f$ and substituting into (3.4.12) we obtain

$$\int_{\widehat{\Omega}^f} -q\nabla_{\widehat{x}}\widehat{\mathbf{u}} : \mathbf{F}^{-T}(\nabla_x\delta\widehat{\mathbf{d}}_f)^T J_{\mathcal{A}} + q\nabla_{\widehat{x}}\widehat{\mathbf{u}} : \mathbf{F}^{-T}J_{\mathcal{A}}\nabla_x\cdot\delta\widehat{\mathbf{d}}_f\ \ d\widehat{\Omega}^f =$$

$$= \int_{\widehat{\Omega}^f} q\nabla_{\widehat{x}}\widehat{\mathbf{u}} : \mathbf{F}^{-T}[-(\nabla_x\delta\widehat{\mathbf{d}}_f)^T + I\nabla_x\cdot\delta\widehat{\mathbf{d}}_f]J_{\mathcal{A}}\ \ d\widehat{\Omega}^f.$$

Following the notation introduced in [FM05, FFT00] we call $\eta = [-(\nabla_x\delta\widehat{\mathbf{d}}_f)^T + I\nabla_x\cdot\delta\widehat{\mathbf{d}}_f]$. Then passing to the current configuration we obtain, using a relation analogous to (3.4.11),

$$D_{\widehat{\mathbf{d}}_f}\int_{\Omega_t^f} q\nabla_x\cdot\mathbf{u}\ \ d{\Omega_t}^f = \int_{\Omega_t^f} q\nabla_x\mathbf{u} : \eta\ \ d{\Omega_t}^f. \tag{3.4.14}$$

With the same strategy we compute the derivative of the convective term (3.4.10b). We have

$$D_{\widehat{\mathbf{d}}_f}\left[\int_{\Omega_t^f}\rho_f\nabla_x\cdot(\mathbf{u}\otimes(\mathbf{u}-\mathbf{w}))\cdot\mathbf{v}_f\ \ d{\Omega_t}^f\right]\delta\widehat{\mathbf{d}}_f =$$

$$= \int_{\widehat{\Omega}^f}\rho_f\nabla_{\widehat{x}}(\widehat{\mathbf{u}}\otimes(\widehat{\mathbf{u}}-\widehat{\mathbf{w}})) : \mathbf{F}^{-T}\eta\cdot\widehat{\mathbf{v}}_f J_{\mathcal{A}}\ \ d\widehat{\Omega}^f + \int_{\Omega_t^f}\frac{\rho_f}{\delta t}\nabla_x\cdot(\mathbf{u}\otimes(-\delta\widehat{\mathbf{d}}_f))\cdot\mathbf{v}_f\ \ d{\Omega_t}^f.$$

The second term corresponds to a standard derivative, coming form the fact that we consider implicitly the domain velocity $\mathbf{w}$ in the convective term (3.4.10b). We notice here the appearance of a third order tensor $\nabla_{\widehat{x}}(\widehat{\mathbf{u}}\otimes(\widehat{\mathbf{u}}-\widehat{\mathbf{w}}))$, whose *double dot* product with $\mathbf{F}$ produces a vector.

The first integral rewritten in the deformed domain simplifies to

$$\int_{\Omega_t^f}\frac{\rho_f}{\delta t}\nabla_x(\mathbf{u}\otimes(\mathbf{u}-\mathbf{w})) : \eta\cdot\mathbf{v}_f\ \ d{\Omega_t}^f.$$

For the numerical computation this integral can be split into different parts, exploiting the product differentiation rule.

For what concerns the stiffness integral (3.4.10c), considering that in our case $\sigma_f = \mu_f(\nabla_x \mathbf{u} + (\nabla_x \mathbf{u})^T) - pI$, we get

$$D_{\widehat{\mathbf{d}}_f}\left[\int_{\Omega_t^f} \sigma_f : \nabla_x \mathbf{v}_f \ d{\Omega_t}^f\right]\delta\widehat{\mathbf{d}}_f =$$

$$= D_{\widehat{\mathbf{d}}_f}\left[\int_{\Omega_t^f} \mu_f\left(\nabla_x \mathbf{u} + (\nabla_x \mathbf{u})^T\right) : \nabla_x \mathbf{v}_f \ d{\Omega_t}^f\right]\delta\widehat{\mathbf{d}}_f + \tag{3.4.15a}$$

$$- D_{\widehat{\mathbf{d}}_f}\left[\int_{\Omega_t^f} pI : \nabla_x \mathbf{v}_f \ d{\Omega_t}^f\right]\delta\widehat{\mathbf{d}}_f. \tag{3.4.15b}$$

First following the same steps as in (3.4.14) we can compute the term (3.4.15b) as

$$\int_{\Omega_t^f} pI : (\eta \nabla_x \mathbf{v}_f) \ \ d{\Omega_t}^f. \tag{3.4.16}$$

We consider then the integral (3.4.15a). Rewriting it in the reference configuration we get

$$\int_{\widehat{\Omega}^f} D_{\widehat{\mathbf{d}}_f}\left[\nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T} + (\nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T})^T\right]\delta\widehat{\mathbf{d}}_f : \mathbf{F}^{-T} J_{\mathcal{A}} \nabla_{\widehat{x}}\widehat{\mathbf{v}}_f \ \ d\widehat{\Omega}^f + \tag{3.4.17a}$$

$$+ \int_{\widehat{\Omega}^f} \nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T} + (\nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T})^T : D_{\widehat{\mathbf{d}}_f}\left[\mathbf{F}^{-T}\right]\delta\widehat{\mathbf{d}}_f J_{\mathcal{A}} \nabla_{\widehat{x}}\widehat{\mathbf{v}}_f \ \ d\widehat{\Omega}^f + \tag{3.4.17b}$$

$$+ \int_{\widehat{\Omega}^f} \nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T} + (\nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T})^T : \mathbf{F}^{-T} D_{\widehat{\mathbf{d}}_f}\left[J_{\mathcal{A}}\right]\delta\widehat{\mathbf{d}}_f \nabla_{\widehat{x}}\widehat{\mathbf{v}}_f \ \ d\widehat{\Omega}^f, \tag{3.4.17c}$$

and using (3.4.13) in (3.4.17a) and (3.4.17b)

$$- \int_{\widehat{\Omega}^f} \left[\nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T}(\nabla_x \delta\widehat{\mathbf{d}}_f)^T + (\nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T}(\nabla_x \delta\widehat{\mathbf{d}}_f)^T)^T\right] : \mathbf{F}^{-T} J_{\mathcal{A}} \nabla_{\widehat{x}}\widehat{\mathbf{v}}_f \ \ d\widehat{\Omega}^f +$$

$$- \int_{\widehat{\Omega}^f} \nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T} + (\nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T})^T : \mathbf{F}^{-T}(\nabla_x \delta\widehat{\mathbf{d}}_f)^T J_{\mathcal{A}} \nabla_{\widehat{x}}\widehat{\mathbf{v}}_f \ \ d\widehat{\Omega}^f +$$

$$+ \int_{\widehat{\Omega}^f} \nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T} + (\nabla_{\widehat{x}}\widehat{\mathbf{u}}\,\mathbf{F}^{-T})^T : \mathbf{F}^{-T} \nabla_x \cdot \delta\widehat{\mathbf{d}}_f J_{\mathcal{A}} \nabla_{\widehat{x}}\widehat{\mathbf{v}}_f \ \ d\widehat{\Omega}^f.$$

Writing the integrals back to the deformed configuration and exploiting (3.4.16) we have

$$D_{\widehat{\mathbf{d}}_f}\left[\int_{\Omega_t^f} \sigma_f : \nabla_x \mathbf{v}_f \ \ d{\Omega_t}^f\right]\delta\widehat{\mathbf{d}}_f = - \int_{\Omega_t^f} \nabla_x \mathbf{u}(\nabla_x \delta\widehat{\mathbf{d}}_f)^T + (\nabla_x \mathbf{u}(\nabla_x \delta\widehat{\mathbf{d}}_f)^T)^T : \nabla_x \mathbf{v}_f \ \ d{\Omega_t}^f +$$

$$+ \int_{\Omega_t^f} \sigma_f : \eta \nabla_x \mathbf{v}_f \ \ d{\Omega_t}^f.$$

The derivative of the term (3.4.10d) has to be added as well to the Jacobian if volume forces are present (and evaluated implicitly with respect to the domain displacement). However this contribute is treated in the same way as (3.4.10a), since the volume force does not depend on the domain displacement:

$$D_{\widehat{\mathbf{d}}_f}\left[\int_{\Omega_t^f} \mathbf{f} \cdot \mathbf{v}_f \ d\Omega_t\right]\delta\widehat{\mathbf{d}}_f = \int_{\Omega_t^f} \mathbf{f} \cdot \mathbf{v}_f \nabla_x \cdot \delta\widehat{\mathbf{d}}_f \ d\Omega_t.$$

Also the derivative of the term (3.4.10e) has to be taken into account if the Neumann boundary Γ_N^f is not fixed. Thus using (3.4.9)

$$D_{\widehat{\mathbf{d}}_f}\left[\int_{\Gamma_N^f}\sigma_f\mathbf{n}^f\cdot\mathbf{v}_f\,d\gamma\right]\delta\widehat{\mathbf{d}}_f = \\ =\int_{\Gamma_N^f}\nabla_x(\sigma_f\mathbf{n}^f\cdot\mathbf{v}_f)\delta\widehat{\mathbf{d}}_f+(\sigma_f\mathbf{n}^f\cdot\mathbf{v}_f)\left\{\nabla_x\cdot(\delta\widehat{\mathbf{d}}_f)-\frac{1}{2}[\nabla_x\delta\widehat{\mathbf{d}}_f+(\nabla_x\delta\widehat{\mathbf{d}}_f)^T]\mathbf{n}^f\cdot\mathbf{n}^f\right\}d\gamma. \tag{3.4.18}$$

We denote $\tilde{\eta}=\left\{\nabla_x\cdot(\delta\widehat{\mathbf{d}}_f)-\frac{1}{2}[\nabla_x\delta\widehat{\mathbf{d}}_f+(\nabla_x\delta\widehat{\mathbf{d}}_f)^T]\mathbf{n}^f\cdot\mathbf{n}^f\right\}$ to light the notations.

Remark 3.4.1. *Notice that the choice of the conservative form for the momentum conservation law instead of the non-conservative one introduces some differences in the terms to be computed. In particular in the non-conservative case the derivative*

$$D_{\widehat{\mathbf{d}}_f}\left[\int_{\Omega_t^f}\frac{\rho_f}{\delta t}\mathbf{u}^n\cdot\mathbf{v}_f\,d\Omega_t\right]\delta\widehat{\mathbf{d}}_f=\int_{\Omega_t^f}\frac{\rho_f}{\delta t}\mathbf{u}^n\cdot\mathbf{v}_f\nabla_x\cdot\delta\widehat{\mathbf{d}}_f\,d\Omega_t$$

needs to be computed as well, while for the convective term the derivative,

$$D_{\widehat{\mathbf{d}}_f}\left[\int_{\Omega_t^f}\rho_f((\mathbf{u}-\mathbf{w})\cdot\nabla_x)\mathbf{u}\cdot\mathbf{v}_f\;\;d{\Omega_t}^f\right]\delta\widehat{\mathbf{d}}_f$$

is needed. This derivative can be computed splitting the integral in two parts and using the same methods described for the other terms.

Remark 3.4.2. *Notice that the imposition of flux as defective boundary condition (cf. Section §2.9) on a fluid outlet involves a surface integral. Thus if the outlet is not clamped and the surface is changing in time an additional shape derivative term is required. The same holds for Robin boundary conditions or generalized Robin conditions, like the absorbing boundary conditions reported in [NV08] (if they are discretized implicitly in time).*

For instance we can express the derivative of the defective flux condition, which is made of two contributions (see e.g. [FGNQ02b]). The first one is the constraint, which reads, given the flux $Q(t):T\mapsto\mathbb{R}$ and the surface S_t depending on time,

$$\int_{S_t}\mathbf{u}\cdot\mathbf{n}^f\,d\gamma=Q.$$

Its derivative can be computed using (3.4.18)

$$D_{\widehat{\mathbf{d}}_f}\left[\int_{S_t}\mathbf{u}\cdot\mathbf{n}^f\,d\gamma\right]\delta\widehat{\mathbf{d}}_f=\int_{S_t}\nabla_x(\mathbf{u}\cdot\mathbf{n})\delta\widehat{\mathbf{d}}_f+\mathbf{u}\cdot\mathbf{n}\tilde{\eta}\,d\gamma.$$

The second is a penalization term added to the momentum conservation equation,

$$\int_{S_t}\mathbf{v}_f\cdot\mathbf{n}^f.$$

Its shape derivative trivially consists of

$$D_{\widehat{\mathbf{d}}_f}\int_{S_t}\mathbf{v}_f\cdot\mathbf{n}^f\delta\widehat{\mathbf{d}}_f=\int_{S_t}\mathbf{v}_f\cdot\mathbf{n}\tilde{\eta}\,d\gamma.$$

The final form of the shape derivatives for the fluid problem reads

$$D_{\widehat{\mathbf{d}}_f}\mathsf{F}(\mathbf{u}_f,\widehat{\mathbf{d}}_s) = \int_{\Omega_t^f} \frac{\rho_f}{\delta t}\mathbf{u}\cdot\mathbf{v}_f\nabla_x\cdot\delta\widehat{\mathbf{d}}_f \;\; d{\Omega_t}^f + \tag{3.4.19a}$$

$$+\int_{\Omega_t^f} \frac{\rho_f}{\delta t}\nabla_x(\mathbf{u}\otimes(\mathbf{u}-\mathbf{w})):\eta\cdot\mathbf{v}_f \;\; d{\Omega_t}^f + \int_{\Omega_t^f} \frac{\rho_f}{\delta t}\nabla_x\cdot(\mathbf{u}\otimes(-\delta\widehat{\mathbf{d}}_f))\cdot\mathbf{v}_f \;\; d{\Omega_t}^f + \tag{3.4.19b}$$

$$-\int_{\Omega_t^f}\nabla_x\mathbf{u}(\nabla_x\delta\widehat{\mathbf{d}}_f)^T + (\nabla_x\mathbf{u}(\nabla_x\delta\widehat{\mathbf{d}}_f)^T)^T:\nabla_x\mathbf{v}_f \;\; d{\Omega_t}^f +$$

$$+\int_{\Omega_t^f}\sigma_f:\nabla_x\mathbf{v}_f\eta \;\; d{\Omega_t}^f + \tag{3.4.19c}$$

$$-\int_{\Omega_t^f}\mathbf{f}\cdot\mathbf{v}_f\nabla_x\cdot\delta\widehat{\mathbf{d}}_f \; d\Omega_t + \tag{3.4.19d}$$

$$+\int_{\Gamma_N^f}\nabla_x(\sigma_f\mathbf{n}^f\cdot\mathbf{v}_f)\delta\widehat{\mathbf{d}}_f + (\sigma_f\mathbf{n}^f\cdot\mathbf{v}_f)\tilde{\eta}\; d\gamma + \tag{3.4.19e}$$

$$+\int_{\Omega_t^f} q\nabla_x\mathbf{u}:\eta \;\; d{\Omega_t}^f \tag{3.4.19f}$$

$$\forall\mathbf{v}_f\in V^f,\ \forall q\in U^p.$$

Remark 3.4.3. *Also the IP stabilization terms reported in Section §2.7.1 feature integrals over the elements faces, thus on a moving domain. This geometric nonlinearity would lead to other terms in the Jacobian matrix. To avoid this computation, and to save computational time, in all our discretizations we compute the stabilization terms on the domain at the previus time level. Bazilevs et al. [BCHZ08] use equal order finite elements, which do not fulfill the inf-sup condition, and a residual–based variational multiscale stabilization technique. The stabilization terms in that case are computed considering the geometry implicitly, the derivatives of these terms are detailed but some of them are neglected in the Jacobian matrix.*

The computation of the complete Jacobian of the FSI problem has been tackled in [BC10a], where the 2D case is considered. The approach followed there is similar to the one we described. It consists in restricting all the integrals to a single finite element, mapping the integrals to the reference configuration and passing the derivatives under the integral sign. The momentum equation is discretized there in its non-conservative form, and the Crank–Nicholson scheme is adopted to advance in time.

Also in [BCHZ08] it is reported a method to compute analytically the shape derivatives terms. The scheme is the same: the integrals on the current finite element are recast back to the original configuration, where the derivatives are taken. The form of the momentum conservation is also the non-conservative one, while the time advancing scheme chosen is the generalized-α method (cf. Section §2.3).

It is worth pointing out that the derivatives are computed on the time discrete system, thus they depend on the time discretization chosen. In particular they depend on the way the momentum conservation is written (either in conservative or non-conservative form).

3.4.3 Implementation

Particular attention should be devoted to the implementation of the terms described above in a Finite Elements code. In fact computing the shape derivatives in an inefficient way

can result in an unnecessary waste of time which may lead to a slow numerical scheme. We explain in this section briefly how this implementation is handled in the free finite element library LifeV[5] and how it may affect the efficiency of the algorithm. All the contributions in terms of coding of the present work were implemented inside this library.

We consider as an example the term

$$\int_{\Omega_t^f} q\nabla_x \mathbf{u} : \eta \;\; d{\Omega_t}^f, \tag{3.4.20}$$

which is particularly representative. The term η is a second order tensor. However to construct it we need the increment $\delta\widehat{\mathbf{d}}_f$, and η depends linearly on $\delta\widehat{\mathbf{d}}_f$. Thus calling $\Theta\delta\widehat{\mathbf{d}}_f = \eta$ and exploiting this linearity we can write the integrand in (3.4.20) in Einstein notation as

$$q\partial_i \mathbf{u}^j \Theta^i_{jk} \delta\widehat{\mathbf{d}}_f{}^k. \tag{3.4.21}$$

At elementary level, introducing the indices l, p and m ranging from 1 to the number N_e of degrees of freedom of the element, restricting the variables q, $\mathbf{u}$ and $\delta\widehat{\mathbf{d}}_f$ to the element e (3.4.21) becomes

$$(q_l)\partial_{pi}(\mathbf{u}_p)^j \Theta^i_{jk} (\delta\widehat{\mathbf{d}}_{f\,m})^k. \tag{3.4.22}$$

We can either choose to compute this quantity every time we perform a multiplication of the Jacobian matrix times a vector $\delta\widehat{\mathbf{d}}_f$, or alternatively we can write the term (3.4.22) as

$$(M_{lm})_k (\delta\widehat{\mathbf{d}}_{f\,m})^k.$$

In this case on every element we have to assemble explicitly the third order tensor $M \in \mathbb{R}^{N_e \times N_e \times 3}$ once per Newton iteration. For a term like e.g. (3.4.19c) the tensor would have been of fourth order $M \in \mathbb{R}^{3\times N_e \times N_e \times 3}$ since the test function in that case is a vector too.

The first approach corresponds to assembling directly the Jacobian-vector product, while the second corresponds to assembling explicitly the Jacobian matrix. In case of a matrix-free solver both approaches can be used, however the first one slows down considerably the computation, since at every solver iteration the shape derivatives vector needs to be re-assembled.

In some cases (see Section §3.2, Remarks 3.2.2 and 3.2.5) the shape derivative terms are anyway assembled in a vector and summed to the right hand side of a linear system. However also in this case it is much more convenient to express the vector as a matrix-vector product, so that the matrix part is assembled just once during the solution of the Jacobian system.

This consideration becomes particularly important when dealing with parallel solvers. In fact usually the number of linear iterations increases when increasing the number of processors, so that the overhead generated by the shape derivative vector assembling would be multiplied times the number of linear iterations. Thus choosing the second approach can benefit to the algorithm parallel performance as well.

Another observation on the implementation is that many tensor quantities (e.g. $\nabla_x \delta\widehat{\mathbf{d}}_f$ or $I\nabla_x \cdot \delta\widehat{\mathbf{d}}_f$) appearing in the shape derivatives assembling are similar and shared between the different terms. Thus to save CPU-time one single loop over the elements should be done to evaluate all the integrals.

These considerations represent an original contribution of this work concerning the Newton algorithm for FSI, and they are taken into account for the implementation in the FE library LifeV. Numerical results on this subject are reported in Section §5.4.

[5] www.lifev.org

Solution of the Algebraic Linear System

In this chapter we address some solution strategies for the algebraic Jacobian system (3.1.2) arising from the linearization of (2.5.1) after time and space discretizations. We show how the block structure of the problem can be exploited leading to modular schemes where the three fields are considered separately. We show in particular that a domain decomposition method (e.g. the Dirichlet–Neumann method) can be recovered at algebraic level with a proper choice of the preconditioner for the linear system.

This chapter contains an original contribution to the field of preconditioners for FSI. Most of the preconditioners discussed are implemented in the open source library LifeV.

For the time discretization considered in this work, the FSI Jacobian system (3.1.2) is non-symmetric indefinite. Efficient methods for solving these kind of problems are either iterative (GMRES, Bi-CGSTAB, see [VdV03]) or direct (unsymmetric multifrontal, parallel QR factorization methods, see [Dav06]). We chose to use the preconditioned GMRES iterative method because it is flexible and robust enough for our problems. The preconditioner computation plays a fundamental role in improving the efficiency of GMRES. Since we deal with a multiphysics problem it is natural to look for block diagonal or block triangular preconditioners which allow to exploit the different physics of the subproblems described.

In Section §4.1 some well known methodologies to build block triangular preconditioners are revised. In particular we focus on preconditioners obtained from block LU factorizations and on block Gauss–Seidel (GS) preconditioners. In Section §4.2 a block GS preconditioning is applied to the FSI system, several factorizations are proposed which allow a separate treatment of the different fields. In Section §4.3 we describe a methodology to build parallel preconditioners which are suited for FSI applications and we study the combination of classical domain decomposition preconditioners with a block triangular preconditioning strategy. The latter is the strategy that applied to all the time discretizations presented (GCE, FI, and other variants) results the most effective in the numerical simulations (cf. Part III).

4.1 Block Triangular Preconditioners

As previously mentioned, block triangular and block diagonal preconditioners are particularly well-suited for multiphysics problems, since they allow to treat separately each field, and exploit the characteristics of the different physical problems (see e.g. [Axe94, Kla98, Sim04, BGL05, TW05, Krz10, AK10] and references therein).

4.1.1 Schur Complement Preconditioners

As already mentioned, the inexact factorization methods described in Section §2.8 can be used to build preconditioners for the FSI system. These methods rely on an inexact block LU factorization obtained approximating the Schur complement. The preconditioners obtained in this way are indeed very common in other application fields. Examples can be found for instance in [RVS10, EHS$^+$06] and references therein, where preconditioners for Stokes and Navier–Stokes equations are described, or in [PS09] where a block preconditioner for the bidomain system modeling the electromechanics of the heart is devised. We refer to [Axe94, BGL05, ESW05, Krz10, AK10] for a general description of the theoretical properties of these preconditioners.

Let us consider a 2×2 generic block system

$$A\mathbf{X} = \mathbf{b}, \tag{4.1.1}$$

such that

$$A = \begin{pmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{pmatrix} \tag{4.1.2}$$

is non singular and the block A_{11} is invertible. In GMRES and many other iterative methods, the preconditioner may be applied on the right or on the left, i.e., respectively just before or just after the computation of the residual. When using a *right preconditioner* the linear system can be rewritten as

$$AP^{-1}\mathbf{Y} = \mathbf{b}; \quad \mathbf{X} = P^{-1}\mathbf{Y}.$$

When using a *left preconditioner* the equivalent problem is

$$P^{-1}A\,\mathbf{X} = P^{-1}\mathbf{b}.$$

Computing a block LU factorization of A yields

$$A = \begin{pmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ A_{12}{A_{11}}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & A_{21} \\ 0 & A_{22} - A_{12}{A_{11}}^{-1}A_{21} \end{pmatrix} = LU. \tag{4.1.3}$$

By taking the second factor of (4.1.3) as right preconditioner for A the right-preconditioned matrix reads

$$AU^{-1} = \begin{pmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{pmatrix} \begin{pmatrix} {A_{11}}^{-1} & -{A_{11}}^{-1}A_{21}S^{-1} \\ 0 & S^{-1} \end{pmatrix} = \begin{pmatrix} I & 0 \\ A_{12}{A_{11}}^{-1} & I \end{pmatrix}, \tag{4.1.4}$$

and GMRES converges in 3 iterations (see e.g. [Sim04]). Frequently, a preconditioning strategy is obtained by approximating the Schur complement $S = A_{22} - A_{12}{A_{11}}^{-1}A_{21}$.

Notice that the same conclusion is obtained for a left preconditioner: if we consider the factorization $A = \tilde{L}\tilde{U}$ such that

$$A = \begin{pmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{pmatrix} = \begin{pmatrix} A_{11} & 0 \\ A_{12}{A_{11}}^{-1} & A_{22} - A_{12}{A_{11}}^{-1}A_{21} \end{pmatrix} \begin{pmatrix} I & A_{21} \\ 0 & I \end{pmatrix} = \tilde{L}\tilde{U}, \tag{4.1.5}$$

and use the lower factor $\tilde{L}$ as left preconditioner we have

$$(\tilde{L})^{-1}A = (\tilde{L})^{-1}\tilde{L}\tilde{U} = U,$$

which has similar spectral properties as (4.1.4). It is easy to compute the block inverse of $\tilde{L}$, as done for U, in terms of S^{-1} and ${A_{11}}^{-1}$.

Alternatively the reduced system

$$S\mathbf{X}_2 = \mathbf{b}_2 - A_{12}{A_{11}}^{-1}\mathbf{b}_1, \tag{4.1.6}$$

where $\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}$ and $\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{bmatrix}$, can be considered instead of (4.1.1). This system is equivalent to the original one, and also in this case choosing a good preconditioner for (4.1.6) amounts to devising an approximation for the Schur complement matrix S.

An indicator for the quality of a preconditioner is the *conditioning* of the preconditioned system. If we solve the linear system (4.1.1) with Krylov iterations and left preconditioner P, then the conditioning can be represented by the *condition number* of $P^{-1}A$

$$K(P^{-1}A) = \frac{\max \sigma(P^{-1}A)}{\min \sigma(P^{-1}A)}, \tag{4.1.7}$$

where $\sigma(A)$ denotes the set of singular values of a matrix A (see e.g. [QSS00]).

The theoretical estimates for the condition number of block-preconditioned systems often rely on the hypothesis that the system is positive (semi)definite. Let us consider the sets of indexes V_1, V_2, and $W = V_1 \times V_2$. In case matrix (4.1.2) is partitioned accordingly with V_1 and V_2, if the system is symmetric positive semidefinite and the block A_{11} is positive definite, then the condition number estimates for the system preconditioned using a Schur complement preconditioner depends on the Cauchy–Bunyakowski–Schwarz (CBS) constant

$$\gamma = \sup_{\mathbf{w}_1 \in W_1, \mathbf{w}_2 \in W_2} \frac{\mathbf{w}_1^T A \mathbf{w}_2}{\{\mathbf{w}_1^T A \mathbf{w}_1 \, \mathbf{w}_2^T A \mathbf{w}_2\}^{\frac{1}{2}}}, \tag{4.1.8}$$

where

$$W_1 = \left\{ \mathbf{v} = \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{0} \end{bmatrix} \in W | \mathbf{v}_1 \in V_1 \right\} \quad W_2 = \left\{ \mathbf{v} = \begin{bmatrix} \mathbf{0} \\ \mathbf{v}_2 \end{bmatrix} \in W | \mathbf{v}_2 \in V_2 \right\}.$$

In [Axe94, Ch.9.3] and [AK10] the condition number of the system (4.1.6) preconditioned with $P = A_{22}$ is shown to satisfy the following (sharp) bound: $K(P^{-1}S) \leq \frac{1}{1-\gamma^2}$. For instance in the numerical FE approximation of a 2D diffusion problem, if the two blocks refer to two different subdomain problems, the CBS constant is $\gamma = 1 - O(h)$, where h is the characteristic dimension of the triangulation. Thus the condition number behaves like $O(h^{-1})$.

The same estimate holds if we build a preconditioner for the matrix in (4.1.3) by taking the U factor and replacing the Schur complement S with A_{22}, which leads to a block GS preconditioner.

We refer to [Axe94, Ch.9], [AK10, KM09] for these and other examples of Schur complements preconditioners and their condition number estimates depending on γ.

In the case of two subdomains, we consider the Steklov–Poincaré approach, described in Section §3.2 for the nonlinear case and which will also be discussed in Section §4.2. The matrix A of a linear diffusion problem can be divided in three subproblems, one per subdomain problem plus an interface problem. Reducing the problem to the interface by means of the Schur complements one can write the algebraic equivalent of (3.2.18) (notice that the algebraic counterparts of the Steklov–Poincaré operators are the Schur complements). Preconditioning this system with one of the local Schur complements, as done for the nonlinear continuous

case in (3.2.20) using a Steklov–Poincaré operator, leads to a system where the condition number is constant, and in particular is independent of h. We say in these cases that the preconditioner is *optimal*. A proof of this result can be found in [TW05].

4.1.2 Block Gauss–Seidel Preconditioners

Another classical strategy to build block triangular preconditioners leads to the block GS preconditioners (which can be seen as a particular case of the block preconditioners discussed above, where the Schur complement is approximated with a diagonal block). As for the Schur complements preconditioners, also the block GS preconditioners exploit the block structure nature of multiphysics problems.

Applied to matrix (4.1.2), a block GS preconditioner reads

$$P_{GS} = \begin{pmatrix} A_{11} & 0 \\ A_{12} & A_{22} \end{pmatrix} = \begin{pmatrix} I & 0 \\ A_{12}{A_{11}}^{-1} & I \end{pmatrix} \begin{pmatrix} A_{11} & 0 \\ 0 & A_{22} \end{pmatrix}, \tag{4.1.9}$$

thus

$$P_{GS}^{-1} = \begin{pmatrix} {A_{11}}^{-1} & 0 \\ -{A_{22}}^{-1}A_{12}{A_{11}}^{-1} & {A_{22}}^{-1} \end{pmatrix} = \begin{pmatrix} I & 0 \\ -{A_{22}}^{-1}A_{12} & I \end{pmatrix} \begin{pmatrix} {A_{11}}^{-1} & 0 \\ 0 & {A_{22}}^{-1} \end{pmatrix}.$$

The corresponding left-preconditioned system then reads

$$P_{GS}^{-1}A = \begin{pmatrix} {A_{11}}^{-1} & 0 \\ 0 & {A_{22}}^{-1} \end{pmatrix} \begin{pmatrix} A_{11} & A_{21} \\ 0 & S \end{pmatrix}. \tag{4.1.10}$$

The robustness of these preconditioners have been tested for instance in [Hei04] for the FSI system.

4.2 Applications to the FSI system

In this section we report some popular solution methods discussed e.g. in [BQQ08a, DDFQ06, BNV08, GGNV10] with a different formalism (similar to the one used in Gee at al. [GKW10]).

We describe some common choices for the preconditioners in the FSI context. In particular we focus on the block GS preconditioners, and we show how classical domain decomposition methods (like most of the so called *partitioned procedures* in FSI) can be seen, when applied to a linearized system, as block GS preconditioned Richardson methods.

We have already seen how to interpret classical domain decomposition methods as fixed point strategies (cf. Section §3.2). Without using a DD strategy at the continuous level, it is possible to work on the algebraic system and recover modularity using a block GS preconditioner.

To clarify the ideas we present first a block GS preconditioner in the case of the GCE discrete algebraic system, and we extend the strategy to the Jacobian system of the nonlinear case in a second step. We remark that the matrix in (2.7.12) has the same form as the fluid–structure blocks in (2.7.10), thus the preconditioning strategies devised for GCE can be easily employed also in the other cases, by applying them to the fluid–structure part of the matrix. The specific way this is done is detailed in Section §4.2.4. Recalling the GCE algebraic system (2.7.12) we can write a block GS iteration as follows

$$\left(\begin{array}{cc|cc|c} C_{ff} & C_{f\Gamma} & 0 & 0 & 0 \\ C_{\Gamma f} & C_{\Gamma\Gamma} & 0 & 0 & 0 \\ \hline 0 & 0 & N_s & N_{s\Gamma} & 0 \\ 0 & 0 & N_{\Gamma s} & N_{\Gamma\Gamma} & -I \\ \hline 0 & I & 0 & -I/\delta t & 0 \end{array}\right) \left(\begin{array}{c} {\mathbf{u}_f}^{k+1} \\ \hline {\mathbf{u}_{f\Gamma}}^{k+1} \\ \hline \widehat{\mathbf{d}}_s^{k+1} \\ \widehat{\mathbf{d}}_{s\Gamma}^{k+1} \\ \hline \lambda^{k+1} \end{array}\right) = \left(\begin{array}{c} {\mathbf{r}_f}^{n+1} \\ \hline {\mathbf{r}_{f\Gamma}}^{n+1} - \lambda^k \\ \hline {\mathbf{r}_s}^{n+1} \\ {\mathbf{r}_{s\Gamma}}^{n+1} \\ \hline -(I/\delta t)\mathbf{d}_{s\Gamma}^n \end{array}\right), \tag{4.2.1}$$

where k represents the block GS iteration index and $n+1$ is the current time level.

Writing equation (2.7.12) as $A\mathbf{X} = \mathbf{b}$ and calling P_{ND} the matrix in (4.2.1) we can rewrite the system (4.2.1) as

$$P_{ND}\mathbf{X}^{k+1} = (P_{ND} - A)\mathbf{X}^k + \mathbf{b},$$

which is the Richardson method preconditioned with P_{ND} (with acceleration parameter set to 1)

$$\mathbf{X}^{k+1} = \mathbf{X}^k + {P_{ND}}^{-1}(-A\mathbf{X}^k + \mathbf{b}). \tag{4.2.2}$$

The block GS preconditioner P_{ND} is also called *Neumann–Dirichlet* preconditioner, because the Richardson method (4.2.2) corresponds to the Neumann–Dirichlet algorithm when the problem is linear (see [QV99]).

To simplify the notations in the following part of this section we condense the fluid and the solid blocks

$$\mathcal{C} = \begin{pmatrix} C_{ff} & C_{f\Gamma} \\ C_{\Gamma f} & C_{\Gamma\Gamma} \end{pmatrix} \quad \mathcal{N} = \begin{pmatrix} N_s & N_{s\Gamma} \\ N_{\Gamma s} & N_{\Gamma\Gamma} \end{pmatrix}.$$

Furthermore we call $\mathcal{O}$ and $\mathcal{I}$ the null and identity matrices respectively, regardless of the numbers of block rows/columns, so that the GCE matrix in system (2.7.12) reads with the new notations

$$A = \left(\begin{array}{cc|cc|c} \multicolumn{2}{c|}{\mathcal{C}} & \multicolumn{2}{c|}{\mathcal{O}} & \begin{matrix}0\\I\end{matrix} \\ \hline \multicolumn{2}{c|}{\mathcal{O}} & \multicolumn{2}{c|}{\mathcal{N}} & \begin{matrix}0\\-I\end{matrix} \\ \hline 0 & I & 0 & -I/\delta t & 0 \end{array}\right). \tag{4.2.3}$$

It is possible to write the Dirichlet–Neumann method as a preconditioned Richardson method as well. In our implementation we test the following two forms of Dirichlet–Neumann preconditioners:

$$P_{DN}^{(1)} = \left(\begin{array}{cc|cc|c} \multicolumn{2}{c|}{\mathcal{C}} & \multicolumn{2}{c|}{\mathcal{O}} & \begin{matrix}0\\I\end{matrix} \\ \hline \multicolumn{2}{c|}{\mathcal{O}} & \multicolumn{2}{c|}{\mathcal{N}} & \begin{matrix}0\\0\end{matrix} \\ \hline 0 & I & 0 & -I/\delta t & 0 \end{array}\right) \tag{4.2.4}$$

and

$$P_{DN}^{(2)} = \left(\begin{array}{cc|cc|c} \multicolumn{2}{c|}{\mathcal{C}} & \multicolumn{2}{c|}{\mathcal{O}} & \begin{matrix}0\\I\end{matrix} \\ \hline \multicolumn{2}{c|}{\mathcal{O}} & \multicolumn{2}{c|}{\mathcal{N}} & \begin{matrix}0\\-I\end{matrix} \\ \hline 0 & I & 0 & 0 & 0 \end{array}\right). \tag{4.2.5}$$

In both cases the Neumann solid problem is decoupled from the other equations, while the last block row represents the Dirichlet coupling condition for the fluid problem.

An efficient and easy way to improve the Dirichlet–Neumann algorithm for the GCE system (equivalent to the relaxed Richardson method) is to use GMRES preconditioned with $P_{DN}^{(1)}$ or $P_{DN}^{(2)}$. We call these methods DN-Richardson and DN-GMRES. They were both investigated and compared in [BQQ08a] applied to the GCE system.

The reason why $P_{DN}^{(1)}$ and $P_{DN}^{(2)}$ are called Dirichlet–Neumann preconditioners is that for $i \in \{1, 2\}$ one application of $(P_{DN}^{(i)})^{-1}$ implies the solution of a Dirichlet problem in the fluid subdomain and of a Neumann problem in the structure subdomain, as it is shown below in (4.2.6) and (4.2.7), where the block factorizations are reported. As a consequence these preconditioners suffer of the same dilemma of the Dirichlet–Neumann scheme when imposing Dirichlet conditions everywhere on the fluid boundary, as described in [KW08a]. In fact this case the pressure is up to a constant, which means that there is a null eigenvalue. There are several ways to overcome this problem. The most naive is to use a Neumann–Dirichlet preconditioner instead of a Dirichlet–Neumann one. This can be achieved by using a different block GS preconditioner, as shown above. Another possibility is to substitute the last block row in (2.7.12) with a linear combination $R_5 \to \alpha_1 R_5 + \alpha_2(R_2 + R_4)$ (and to apply the same transformation to the right hand side of the system), with properly tuned parameters α_1 and α_2, which leads to the following matrix

$$\left(\begin{array}{cc|c} \mathcal{C} & \mathcal{O} & \begin{matrix}0\\ I\end{matrix} \\ \hline \mathcal{O} & \mathcal{N} & \begin{matrix}0\\ -I\end{matrix} \\ \hline \alpha_2 C_{\Gamma f} \quad \alpha_2 C_{\Gamma\Gamma} + \alpha_1 I & \alpha_2 N_{\Gamma s} \quad \alpha_2 N_{\Gamma\Gamma} - \alpha_1 I/\delta t & 0 \end{array}\right).$$

Then by neglecting the $-I$ block we obtain a Robin–Neumann preconditioner, where the fluid problem is endowed with an interface Robin condition.

We remark that the construction of both the preconditioners $P_{DN}^{(1)}$ and $P_{DN}^{(2)}$, as well as of P_{ND}, is *modular*, in the sense that these preconditioners can be represented as the product of two matrices containing the solid and the fluid blocks respectively. We report here explicitly the factorization of $P_{DN}^{(1)}$ and $P_{DN}^{(2)}$ which is considered in our numerical tests (cf. Section §6.2):

$$\begin{aligned} P_{DN}^{(1)} &= \left(\begin{array}{cc|c} \mathcal{C} & \mathcal{O} & \begin{matrix}0\\ I\end{matrix} \\ \hline \mathcal{O} & \mathcal{N} & \begin{matrix}0\\ 0\end{matrix} \\ \hline 0 \quad I & 0 \quad -I/\delta t & 0 \end{array}\right) \\ &= \underbrace{\left(\begin{array}{cc|c} \mathcal{I} & \mathcal{O} & \begin{matrix}0\\ 0\end{matrix} \\ \hline \mathcal{O} & \mathcal{N} & \begin{matrix}0\\ 0\end{matrix} \\ \hline 0 \quad 0 & 0 \quad 0 & I \end{array}\right)}_{P_{S,1}^{(1)}} \underbrace{\left(\begin{array}{cc|c} \mathcal{C} & \mathcal{O} & \begin{matrix}0\\ I\end{matrix} \\ \hline \mathcal{O} & \mathcal{I} & \begin{matrix}0\\ 0\end{matrix} \\ \hline 0 \quad I & 0 \quad -I/\delta t & 0 \end{array}\right)}_{P_{F,2}^{(1)}}, \end{aligned} \tag{4.2.6}$$

while for $P_{DN}^{(2)}$:

$$P_{DN}^{(2)} = \left(\begin{array}{cc|c} \mathcal{C} & \mathcal{O} & \begin{matrix}0\\I\end{matrix} \\ \hline \mathcal{O} & \mathcal{N} & \begin{matrix}0\\-I\end{matrix} \\ \hline \begin{matrix}0 & I\end{matrix} & \begin{matrix}0 & 0\end{matrix} & 0 \end{array}\right) = \underbrace{\left(\begin{array}{cc|c} \mathcal{C} & \mathcal{O} & \begin{matrix}0\\I\end{matrix} \\ \hline \mathcal{O} & \mathcal{I} & \begin{matrix}0\\0\end{matrix} \\ \hline \begin{matrix}0 & I\end{matrix} & \begin{matrix}0 & 0\end{matrix} & 0 \end{array}\right)}_{P_{F,1}^{(2)}} \underbrace{\left(\begin{array}{cc|c} \mathcal{I} & \mathcal{O} & \begin{matrix}0\\0\end{matrix} \\ \hline \mathcal{O} & \mathcal{N} & \begin{matrix}0\\-I\end{matrix} \\ \hline \begin{matrix}0 & 0\end{matrix} & \begin{matrix}0 & 0\end{matrix} & I \end{array}\right)}_{P_{S,2}^{(2)}}. \tag{4.2.7}$$

We have shown that the application of the inverse preconditioner P^{-1} amounts to inverting the fluid and structure factors separately. In practice for a vector $\mathbf{X}$ the computation of e.g. $P_{F,1}^{(2)}\mathbf{X}$ and $P_{S,2}^{(2)}\mathbf{X}$ is performed by means of an LU factorization of the matrices $P_{F,1}^{(2)}$ and $P_{S,2}^{(2)}$, or by GMRES (or CG) iterations. The former strategy is appealing because the factorizations computed can be reused throughout the solution of the global linear system, however memory issues may limit the size of the problem considered. The latter consists in using e.g. preconditioned GMRES iterations for the fluid problem and preconditioned CG for the structure one. However this strategy introduces inner iterations to solve the subproblems. Since the number of inner iterations usually increases when the number of processors grows, even though the outer iterations remain constant, the total number of iterations (which is the product of the two) grows too fast. A possible remedy could be the use of a flexible GMRES with a fixed number of inner iterations. This has been done in [EHS+08, EHST03] for the Navier–Stokes equations. Our solution is explained in the next section and consists in substituting the factors with preconditioners computed with classical strategies (e.g. domain decomposition) tailored on the problem.

We describe in the three following subsections some other block preconditioners for the GCE system (2.6.2). Most of these preconditioning techniques are implemented in the FE library LifeV. We anticipate that the strategies devised for the GCE approach can be employed as well in the other cases for the first two blocks (corresponding to the fluid and structure equations), as shown in Section §4.2.4.

4.2.1 Robin–Robin Preconditioners

A generalization of the Dirichlet–Neumann approach is the Robin-Robin one [BNV08], as discussed in Section §3.2. Substituting, in the GCE time-discretized equation (2.6.2) in strong form, the Neumann and Dirichlet coupling conditions with Robin transmission conditions

leads to the following

$$\begin{aligned}
\rho_f \delta_t \mathbf{u}^{n+1} + \rho_f(\mathbf{u}^n - \mathbf{w}^{n+1}) \cdot \nabla_x \mathbf{u}^{n+1} - \nabla_x \cdot {\sigma_f}^{n+1} &= {\mathbf{f}_f}^{n+1} && \text{in } \Omega^f_{t\,n+1} \\
\nabla_x \cdot \mathbf{u}^{n+1} &= 0 && \text{in } \Omega^f_{t\,n+1} \\
\alpha_f(\mathbf{u}^{n+1} \circ {\mathcal{A}_t}^{n+1} - \delta_t \widehat{\mathbf{d}}_s^{n+1}) + \widehat{\sigma}_f^{n+1} \cdot \widehat{\mathbf{n}}^f + \mathbf{\Pi}^{n+1} \cdot \widehat{\mathbf{n}}^s &= 0 && \text{on } \widehat{\Gamma} \\
\widehat{\rho}_s \delta_{tt} \widehat{\mathbf{d}}_s^{n+1} - \nabla_{\widehat{x}} \cdot \mathbf{\Pi}^{n+1} &= {\mathbf{f}_s}^{n+1} && \text{in } \widehat{\Omega}^s \\
\alpha_s(\mathbf{u}^{n+1} \circ {\mathcal{A}_t}^{n+1} - \delta_t \widehat{\mathbf{d}}_s^{n+1}) + \widehat{\sigma}_f^{n+1} \cdot \widehat{\mathbf{n}}^f + \mathbf{\Pi}^{n+1} \cdot \widehat{\mathbf{n}}^s &= 0 && \text{on } \widehat{\Gamma} \\
\text{+conditions on the external boundaries.}
\end{aligned} \tag{4.2.8}$$

The algebraic system deriving from (4.2.8) reads

$$\left(\begin{array}{cc|cc}
C_{ff} & C_{f\Gamma} & 0 & 0 \\
C_{\Gamma f} & C_{\Gamma\Gamma} + \alpha_f I & N_{\Gamma s} & N_{\Gamma\Gamma} - \alpha_f I/\delta t \\
\hline
0 & 0 & N_s & N_{s\Gamma} \\
C_{\Gamma f} & C_{\Gamma\Gamma} + \alpha_s I & N_{\Gamma s} & N_{\Gamma\Gamma} - \alpha_s I/\delta t
\end{array}\right)
\left(\begin{array}{c}
{\mathbf{u}_f}^{n+1} \\
{\mathbf{u}_\Gamma}^{n+1} \\
\widehat{\mathbf{d}}_s^{n+1} \\
\widehat{\mathbf{d}}_{s\Gamma}^{n+1}
\end{array}\right) =$$

$$\left(\begin{array}{c}
\mathbf{r}_f \\
{\mathbf{r}_{s\Gamma}}^{n+1} + {\mathbf{r}_{f\Gamma}}^{n+1} - \alpha_f \widehat{\mathbf{d}}_{s\Gamma}^n/\delta t \\
\mathbf{r}_s \\
{\mathbf{r}_{s\Gamma}}^{n+1} + {\mathbf{r}_{f\Gamma}}^{n+1} - \alpha_s \widehat{\mathbf{d}}_{s\Gamma}^n/\delta t
\end{array}\right). \tag{4.2.9}$$

Here we chose not to introduce the multiplier λ for the sake of clarity, however the system can be easily rewritten in the *augmented* form (2.7.12).

As done in the previous case, we can write a block GS iteration for this system discretized in space and time.

$$\left(\begin{array}{cc|cc}
C_{ff} & C_{f\Gamma} & 0 & 0 \\
C_{\Gamma f} & C_{\Gamma\Gamma} + \alpha_f I & 0 & 0 \\
\hline
0 & 0 & N_s & N_{s\Gamma} \\
C_{\Gamma f} & C_{\Gamma\Gamma} + \alpha_s I & N_{\Gamma s} & N_{\Gamma\Gamma} - \alpha_s I/\delta t
\end{array}\right)
\left(\begin{array}{c}
{\mathbf{u}_f}^{k+1} \\
{\mathbf{u}_\Gamma}^{k+1} \\
\widehat{\mathbf{d}}_s^{k+1} \\
\widehat{\mathbf{d}}_{s\Gamma}^{k+1}
\end{array}\right) =$$

$$\left(\begin{array}{c}
{\mathbf{r}_f}^{n+1} \\
N_{\Gamma s}\widehat{\mathbf{d}}_s^k + (\alpha_f I/\delta t - N_{\Gamma\Gamma})\widehat{\mathbf{d}}_{s\Gamma}^k + {\mathbf{r}_{s\Gamma}}^{n+1} + {\mathbf{r}_{f\Gamma}}^{n+1} - \alpha_f \widehat{\mathbf{d}}_{s\Gamma}^n/\delta t \\
\mathbf{r}_s \\
{\mathbf{r}_{s\Gamma}}^{n+1} + {\mathbf{r}_{f\Gamma}}^{n+1} - \alpha_s \widehat{\mathbf{d}}_{s\Gamma}^n/\delta t
\end{array}\right). \tag{4.2.10}$$

With a method analogous to that used in the Dirichlet–Neumann case it is possible to show that the Robin–Robin method can be rewritten as system (2.7.12) preconditioned with

$$P_{RR} = \begin{pmatrix}
I & 0 & 0 & 0 \\
0 & \frac{1}{\alpha_f - \alpha_s} I & 0 & \frac{1}{\alpha_s - \alpha_f} I \\
0 & 0 & I & 0 \\
0 & \frac{\alpha_s}{\alpha_s - \alpha_f} I & 0 & \frac{\alpha_f}{\alpha_f - \alpha_s} I
\end{pmatrix}
\left(\begin{array}{cc|cc}
C_{ff} & C_{f\Gamma} & 0 & 0 \\
C_{\Gamma f} & C_{\Gamma\Gamma} + \alpha_f I & 0 & 0 \\
\hline
0 & 0 & N_s & N_{s\Gamma} \\
C_{\Gamma f} & C_{\Gamma\Gamma} + \alpha_s I & N_{\Gamma s} & N_{\Gamma\Gamma} - \alpha_s I/\delta t
\end{array}\right). \tag{4.2.11}$$

In fact defining

$$Q = \begin{pmatrix}
I & 0 & 0 & 0 \\
0 & \alpha_f I & 0 & I \\
0 & 0 & I & 0 \\
0 & \alpha_s I & 0 & I
\end{pmatrix},$$

and

$$P = \begin{pmatrix} C_{ff} & C_{f\Gamma} & 0 & 0 \\ C_{\Gamma f} & C_{\Gamma\Gamma} + \alpha_f I & 0 & 0 \\ 0 & 0 & N_s & N_{s\Gamma} \\ C_{\Gamma f} & C_{\Gamma\Gamma} + \alpha_s I & N_{\Gamma s} & N_{\Gamma\Gamma} - \alpha_s I/\delta t \end{pmatrix},$$

if $A\mathbf{X} = \mathbf{b}$ represents equation (2.7.12), we can write the system (4.2.9) as

$$P\mathbf{X}^{k+1} = (P - QA)\mathbf{X}^k + Q\mathbf{b},$$

and thus

$$Q^{-1}P\mathbf{X}^{k+1} = Q^{-1}P\mathbf{X}^k - A\mathbf{X}^k + \mathbf{b},$$

which is the Richardson method preconditioned with $P_{RR} = Q^{-1}P$.

4.2.2 Dirichlet–Dirichlet Preconditioners

The preconditioning strategy that we are going to describe is reported here formally, but it is not tested in the numerical applications part since it is not implemented yet in LifeV.

Let us recall the definition of the Steklov–Poincaré operators for the fluid and structure subdomains (3.2.19). Each application of one of these operators implies the solution of a subproblem with Dirichlet coupling conditions on the interface. The equation

$$\Sigma_f(\widehat{\mathbf{d}}_{s\Gamma}) + \Sigma_s(\widehat{\mathbf{d}}_{s\Gamma}) = 0 \text{ on } \widehat{\Gamma} \tag{4.2.12}$$

represents the Steklov–Poincaré formulation of the FSI problem, and expresses the weak stress continuity across the interface. The solution of equation (4.2.12), as shown in Section §3.2, can be viewed as the result of a further partition of the original problem, where we solve separately an equation for the fluid and solid internal domains and an equation for the interface domain which is coupled with both. We consider again the GCE time discretization, and we recall that the discrete Steklov–Poincaré operators are the Schur complements on the interface degrees of freedom of the matrix in (2.7.12).

As done in Section §3.2 we introduce the multiplier $\eta^{n+1} = \widehat{\mathbf{d}}_{s\Gamma}^{n+1}$ representing the interface displacement. The system in strong form discretized in time reads

$$\begin{aligned}
\rho_f \delta_t \mathbf{u}^{n+1} + \rho_f(\mathbf{u}^n - \mathbf{w}^{n+1}) \cdot \nabla_x \mathbf{u}^{n+1} - \nabla_x \cdot {\sigma_f}^{n+1} &= {\mathbf{f}_f}^{n+1} && \text{in } \Omega_{t\,n+1}^f \\
\nabla_{\widehat{x}} \cdot \mathbf{u}^{n+1} &= 0 && \text{in } \Omega_{t\,n+1}^f \\
\mathbf{u}^{n+1} - \delta_t \eta^{n+1} &= 0 && \text{on } \widehat{\Gamma} \\
\widehat{\rho}_s \delta_{tt} \widehat{\mathbf{d}}_s^{n+1} - \nabla_{\widehat{x}} \cdot (\mathbf{\Pi})^{n+1} &= {\mathbf{f}_s}^{n+1} && \text{in } \widehat{\Omega}^s \\
\widehat{\mathbf{d}}_{s\Gamma}^{n+1} &= \eta^{n+1} && \text{on } \Gamma_0 \\
\mathbf{\Pi}^{n+1} \cdot \widehat{\mathbf{n}}^s + (\widehat{\sigma}_f)^{n+1} \cdot \widehat{\mathbf{n}}^f &= 0 && \text{on } \widehat{\Gamma}
\end{aligned} \tag{4.2.13}$$

+conditions on the external boundaries.

The resulting linear system is naturally written in augmented form as

$$\left(\begin{array}{cc|cc|c} C_{ff} & C_{f\Gamma} & 0 & 0 & 0 \\ 0 & I & 0 & 0 & -I/\delta t \\ \hline 0 & 0 & N_s & N_{s\Gamma} & 0 \\ 0 & 0 & 0 & I & -I \\ \hline C_{\Gamma f} & C_{\Gamma\Gamma} & N_{\Gamma s} & N_{\Gamma\Gamma} & 0 \end{array}\right) \left(\begin{array}{c} {\mathbf{u}_f}^{n+1} \\ {\mathbf{u}_\Gamma}^{n+1} \\ \widehat{\mathbf{d}}_s^{n+1} \\ \widehat{\mathbf{d}}_{s\Gamma}^{n+1} \\ \eta^{n+1} \end{array}\right) = \left(\begin{array}{c} {\mathbf{r}_f}^{n+1} \\ -\eta^n/\delta t \\ {\mathbf{r}_s}^{n+1} \\ 0 \\ {\mathbf{r}_{s\Gamma}}^{n+1} + {\mathbf{r}_{f\Gamma}}^{n+1} \end{array}\right). \tag{4.2.14}$$

Notice that all the entries in the fourth and the last columns can be interchanged, because of the condition $\eta^{n+1} = \widehat{\mathbf{d}}_{s\Gamma}^{n+1}$. The algebraic counterpart of the Stéklov–Poincaré formulation recalled here could be obtained by computing the Schur complements of the fluid and structure blocks, in order to eliminate the variables ${\mathbf{u}_f}^{n+1}, {\mathbf{u}_\Gamma}^{n+1}, \widehat{\mathbf{d}}_s^{n+1}, \widehat{\mathbf{d}}_{s\Gamma}^{n+1}$ in (4.2.14), obtaining a linear system in η^{n+1}. However in this section, as done for the Dirichlet–Neumann case, we describe a preconditioner for the whole FSI problem (without restricting to a subset of variables) whose computation involves the same type of problems encountered when solving a Richardson iteration for the Schur complements system (i.e., two Dirichlet problems for the fluid and structure).

System (4.2.14) can be solved with block GS iterations, or a block GS preconditioner can be used for GMRES iterations on the system (4.2.14).

We call *Dirichlet–Dirichlet* preconditioner the following:

$$P_{DD} = \left(\begin{array}{cc|cc|c} C_{ff} & C_{f\Gamma} & 0 & 0 & 0 \\ 0 & I & 0 & 0 & 0 \\ \hline 0 & 0 & N_s & N_{s\Gamma} & 0 \\ 0 & 0 & 0 & -I & 0 \\ \hline C_{\Gamma f} & C_{\Gamma\Gamma} & N_{\Gamma s} & 0 & N_{\Gamma\Gamma} \end{array}\right).$$

Instead of considering Dirichlet problems F_D and S_D in the fluid and solid blocks we can take Neumann ones (F_N and S_N, see remark 3.2.6). Then stressing the velocity continuity coupling condition is obtained by imposing the equality between the inverse of the Steklov–Poincaré operators (which are *Neumann to Dirichlet maps*). We obtain (with the same steps retrieved for the previous case) the following linear system

$$\left(\begin{array}{cc|cc|c} \multicolumn{2}{c|}{\mathcal{C}} & \multicolumn{2}{c|}{\mathcal{O}} & \begin{matrix}0\\I\end{matrix} \\ \hline \multicolumn{2}{c|}{\mathcal{O}} & \multicolumn{2}{c|}{\mathcal{N}} & \begin{matrix}0\\-I\end{matrix} \\ \hline 0 & I & 0 & -I/\delta t & 0 \end{array}\right) \left(\begin{array}{c} {\mathbf{u}_f}^{n+1} \\ {\mathbf{u}_\Gamma}^{n+1} \\ \widehat{\mathbf{d}}_s^{n+1} \\ \widehat{\mathbf{d}}_{s\Gamma}^{n+1} \\ \lambda^{n+1} \end{array}\right) = \left(\begin{array}{c} {\mathbf{r}_f}^{n+1} \\ {\mathbf{r}_{f\Gamma}}^{n+1} \\ {\mathbf{r}_s}^{n+1} \\ {\mathbf{r}_{s\Gamma}}^{n+1} \\ -\widehat{\mathbf{d}}_{s\Gamma}^{n}/\delta t \end{array}\right). \tag{4.2.15}$$

This time the multiplier is denoted with λ^{n+1} and it represents the residual of the two equations on the interface.

Sometimes in practical implementations the usage of preconditioners like P_{DN} in (4.2.4) or (4.2.5) causes problems because of the zero diagonal block in the lower-right corner. To avoid this problem we can perform a change of variables in (4.2.1) derived from the fourth row of the system (2.7.12):

$$\widehat{\mathbf{d}}_{s\Gamma}^{n+1} = N_{\Gamma\Gamma}^{-1}(-N_{\Gamma_s}\widehat{\mathbf{d}}_s^{n+1} + \lambda^{n+1} + {\mathbf{r}_{s\Gamma}}^{n+1}).$$

This change of variable in the last equation of (4.2.15) leads to the following system

$$\left(\begin{array}{cc|c|c} \multicolumn{2}{c|}{\mathcal{C}} & 0 & \begin{matrix}0\\I\end{matrix} \\ \hline \multicolumn{2}{c|}{\mathcal{O}} & \multicolumn{1}{c|}{\mathcal{N}} & \begin{matrix}0\\-I\end{matrix} \\ \hline 0 & N_{\Gamma\Gamma} & N_{\Gamma s}/\delta t \quad 0 & -I/\delta t \end{array}\right) \left(\begin{array}{c} {\mathbf{u}_f}^{n+1} \\ {\mathbf{u}_\Gamma}^{n+1} \\ \widehat{\mathbf{d}}_s^{n+1} \\ \widehat{\mathbf{d}}_{s\Gamma}^{n+1} \\ \lambda^{n+1} \end{array}\right) = \left(\begin{array}{c} {\mathbf{r}_f}^{n+1} \\ {\mathbf{r}_{f\Gamma}}^{n+1} \\ {\mathbf{r}_s}^{n+1} \\ {\mathbf{r}_{s\Gamma}}^{n+1} \\ -N_{\Gamma\Gamma}\widehat{\mathbf{d}}_{s\Gamma}^{n}/\delta t - {\mathbf{r}_{s\Gamma}}^{n+1}/\delta_t \end{array}\right). \tag{4.2.16}$$

A block GS iteration reads

$$\left(\begin{array}{cccc|c} \mathcal{C} & & \mathcal{O} & & 0 \\ & & & & 0 \\ \hline \mathcal{O} & & \mathcal{N} & & 0 \\ & & & & 0 \\ \hline 0 & N_{\Gamma\Gamma} & N_{\Gamma s}/\delta t & 0 & -I/\delta t \end{array}\right) \left(\begin{array}{c} \mathbf{u}_f{}^{k+1} \\ \mathbf{u}_\Gamma{}^{k+1} \\ \widehat{\mathbf{d}}_s^{k+1} \\ \widehat{\mathbf{d}}_{s\Gamma}^{k+1} \\ \lambda^{k+1} \end{array}\right) = \left(\begin{array}{c} \mathbf{r}_f{}^{n+1} \\ \mathbf{r}_{f\Gamma}{}^{n+1} - \lambda^k \\ \mathbf{r}_s{}^{n+1} \\ \mathbf{r}_{s\Gamma}{}^{n+1} + \lambda^k \\ -N_{\Gamma\Gamma}\widehat{\mathbf{d}}_{s\Gamma}^n/\delta t - \mathbf{r}_{s\Gamma}{}^{n+1}/\delta t \end{array}\right). \tag{4.2.17}$$

The corresponding block GS preconditioner involves two Neumann subproblems, thus we call it P_{NN}.

Remark 4.2.1. *In the classical domain decomposition nomenclature this preconditioner corresponds indeed to a* Neumann–Neumann *preconditioner, while P_{DD} corresponds to a FETI (Finite Elements Tearing and Interconnecting) preconditioner [TW05].*

4.2.3 Other Additive Preconditioners

Another possibility concerning the preconditioning strategies for the system (2.7.12) is to rewrite the matrix in (4.2.15) as sum of two matrices $A = A_{(1)} + A_{(2)}$ and then approximate its inverse with $A^{-1} \approx A_{(1)}^{-1} + A_{(2)}^{-1}$. We call these types of preconditioners *additive* (see [AK10]).

To further simplify the notations here we introduce the following convention $\mathbb{I} = \begin{pmatrix} 0 \\ I \end{pmatrix}$ and $\mathbb{I}^T = \begin{pmatrix} 0 & I \end{pmatrix}$.

We have

$$A = \begin{pmatrix} \mathcal{C} & 0 & \mathbb{I} \\ 0 & \mathcal{N} & -\mathbb{I} \\ \mathbb{I}^T & -\mathbb{I}^T/\delta t & 0 \end{pmatrix} =$$

$$= \begin{pmatrix} \frac{1}{2}\mathcal{C} & 0 & \mathbb{I} \\ 0 & \frac{1}{2}\mathcal{N} & -\frac{1}{2}\mathbb{I} \\ \frac{1}{2}\mathbb{I}^T & 0 & 0 \end{pmatrix} + \begin{pmatrix} \frac{1}{2}\mathcal{C} & 0 & 0 \\ 0 & \frac{1}{2}\mathcal{N} & -\frac{1}{2}\mathbb{I} \\ \frac{1}{2}\mathbb{I}^T & -\mathbb{I}^T/\delta t & 0 \end{pmatrix} = A_{(1)} + A_{(2)}.$$

A block factorization for these two matrices can easily be computed, in fact

$$A_{(1)} = \begin{pmatrix} \frac{1}{2}\mathcal{C} & 0 & \mathbb{I} \\ 0 & I & 0 \\ \frac{1}{2}\mathbb{I}^T & 0 & \boxed{\mathrm{I}} \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & \frac{1}{2}\mathcal{N} & -\frac{1}{2}\mathbb{I} \\ 0 & 0 & I \end{pmatrix},$$

while

$$A_{(2)} = \begin{pmatrix} \frac{1}{2}\mathcal{C} & 0 & 0 \\ 0 & I & 0 \\ \frac{1}{2}\mathbb{I}^T & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & \frac{1}{2}\mathcal{N} & -\frac{1}{2}\mathbb{I} \\ 0 & -\mathbb{I}^T/\delta t & \boxed{\text{-I}} \end{pmatrix}.$$

Notice that in the factorization of the matrix $A_{(1)}$ we added an identity block (the boxed blocks highlighted above) to avoid that some factors have null diagonal blocks. The matrix $A_{(1)}$ involves two decoupled problems. Without the identity block added on the lower-right corner these problems would have been of Dirichlet type on the fluid and of Neumann type on the solid. However the introduction of the identity block mixes the two coupling conditions in the first factor generating a Robin fluid problem. The same consideration holds for the block factorization of the matrix $A_{(2)}$: without the highlighted identity block its inversion would

imply the solution of a Neumann problem for the fluid and a Dirichlet one for the solid, while when the identity block is added the solid problem becomes of Robin type.

Another possible splitting of the matrix A would be

$$A = \tilde{A}_{(1)} + \tilde{A}_{(2)} = \begin{pmatrix} \frac{1}{2}\mathcal{C} & 0 & \mathbb{I} \\ 0 & \frac{1}{2}\mathcal{N} & -\mathbb{I} \\ 0 & 0 & I \end{pmatrix} + \begin{pmatrix} \frac{1}{2}\mathcal{C} & 0 & 0 \\ 0 & \frac{1}{2}\mathcal{N} & 0 \\ \mathbb{I} & -\mathbb{I}/\delta t & -I \end{pmatrix}.$$

with the following factorizations:

$$\tilde{A}_{(1)} = \begin{pmatrix} \frac{1}{2}\mathcal{C} & 0 & \mathbb{I} \\ 0 & I & 0 \\ 0 & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & \frac{1}{2}\mathcal{N} & -\mathbb{I} \\ 0 & 0 & I \end{pmatrix},$$

and

$$\tilde{A}_{(2)} = \begin{pmatrix} \frac{1}{2}\mathcal{C} & 0 & 0 \\ 0 & I & 0 \\ \mathbb{I} & 0 & I \end{pmatrix} \begin{pmatrix} I & 0 & 0 \\ 0 & \frac{1}{2}\mathcal{N} & 0 \\ 0 & -\mathbb{I}/\delta t & -I \end{pmatrix}.$$

The inversion of these two blocks implies again two Neumann and two Dirichlet solutions of the subdomain problems.

Remark 4.2.2. *We notice that for a splitting like $A = A_{(1)} + A_{(2)}$, the situation in which $A_{(1)} = A_{(2)} = \frac{1}{2}A$ would lead to the exact factorization of A, thus a situation in which $A_{(1)}$ is closer to $A_{(2)}$ is preferable.*

4.2.4 Extension to Other Time Discretizations

The considerations made for the GCE time discretization hold true also for the other variants, in particular we show here a block GS preconditioner for the Jacobian matrix of the fully implicit system J_{FI}.

The Jacobian matrix in (3.1.2) reads

$$J_{FI} = \left(\begin{array}{cc|cc|c|cc} D_{\mathbf{u}_f}C_f & D_{\mathbf{u}_\Gamma}C_f & 0 & 0 & 0 & D_{\widehat{\mathbf{d}}_f}C_f & D_{\widehat{\mathbf{d}}_{f\Gamma}}C_f \\ D_{\mathbf{u}_f}C_\Gamma & D_{\mathbf{u}_\Gamma}C_\Gamma & 0 & 0 & I & D_{\widehat{\mathbf{d}}_f}C_\Gamma & D_{\widehat{\mathbf{d}}_{f\Gamma}}C_\Gamma \\ \hline 0 & 0 & D_{\widehat{\mathbf{d}}_s}N_s & D_{\widehat{\mathbf{d}}_{s\Gamma}}N_s & 0 & 0 & 0 \\ 0 & 0 & D_{\widehat{\mathbf{d}}_s}N_\Gamma & D_{\widehat{\mathbf{d}}_{s\Gamma}}N_\Gamma & \boxed{\text{-I}} & 0 & 0 \\ \hline 0 & I & 0 & -I/\delta t & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & H_{ff} & H_{f\Gamma} \\ 0 & 0 & 0 & -I & 0 & 0 & I \end{array}\right), \tag{4.2.18}$$

where the block rows correspond, from top to bottom, to the linearized fluid problem, the solid problem, the condition on the velocity continuity at the interface, and the geometry problem. We used the abridged notations $D_{\mathbf{x}}C_f = D_{\mathbf{x}}C_{ff} + D_{\mathbf{x}}C_{f\Gamma}$, $D_{\mathbf{x}}C_\Gamma = D_{\mathbf{x}}C_{\Gamma f} + D_{\mathbf{x}}C_{\Gamma\Gamma}$, $D_{\mathbf{x}}N_s = D_{\mathbf{x}}N_{ss} + D_{\mathbf{x}}N_{s\widehat{\Gamma}}$, and $D_{\mathbf{x}}N_{\widehat{\Gamma}} = D_{\mathbf{x}}N_{\widehat{\Gamma}s} + D_{\mathbf{x}}N_{\widehat{\Gamma}\widehat{\Gamma}}$, for any variable $\mathbf{x}$. For $\mathbf{x} = \widehat{\mathbf{d}}_f$ and $\mathbf{x} = \widehat{\mathbf{d}}_{f\Gamma}$, $D_{\mathbf{x}}$ represents the shape derivatives introduced in Section §3.4.

As done for the GCE system in (4.2.3) we express also the Jacobian matrix J_{FI} in a more compact form, using the following notations

$$D_{\mathbf{u}_f}\mathcal{C} = \begin{pmatrix} D_{\mathbf{u}_f}C_f & D_{\mathbf{u}_\Gamma}C_f \\ D_{\mathbf{u}_f}C_\Gamma & D_{\mathbf{u}_\Gamma}C_\Gamma \end{pmatrix} \qquad D_{\widehat{\mathbf{d}}_f}\mathcal{C} = \begin{pmatrix} D_{\widehat{\mathbf{d}}_f}C_f & D_{\widehat{\mathbf{d}}_{f\Gamma}}C_f \\ D_{\widehat{\mathbf{d}}_f}C_\Gamma & D_{\widehat{\mathbf{d}}_{f\Gamma}}C_\Gamma \end{pmatrix}$$

$$D_{\widehat{\mathbf{d}}_s}\mathcal{N} = \begin{pmatrix} D_{\widehat{\mathbf{d}}_s}N_s & D_{\widehat{\mathbf{d}}_{s\Gamma}}N_s \\ D_{\widehat{\mathbf{d}}_s}N_\Gamma & D_{\widehat{\mathbf{d}}_{s\Gamma}}N_\Gamma \end{pmatrix} \qquad \mathcal{H} = \begin{pmatrix} H_{ff} & H_{f\Gamma} \\ 0 & I \end{pmatrix}.$$

By neglecting the shape derivatives, it is possible to devise a GS preconditioner of J_{FI} as follows

$$P_{QN} = \left(\begin{array}{c|cc|c} D_{\mathbf{u}_f}\mathcal{C} & \mathcal{O} & \begin{matrix}0\\I\end{matrix} & \mathcal{O} \\ \hline \mathcal{O} & D_{\widehat{\mathbf{d}}_s}\mathcal{N} & \begin{matrix}0\\-I\end{matrix} & \mathcal{O} \\ \hline \begin{matrix}0 & I\end{matrix} & \begin{matrix}0 & -I/\delta t\end{matrix} & 0 & \mathcal{O} \\ \hline \mathcal{O} & \begin{matrix}0 & 0\\0 & -I\end{matrix} & \begin{matrix}0\\0\end{matrix} & \mathcal{H} \end{array}\right).$$

The label QN stands for *Quasi Newton*. Indeed what we call Quasi Newton method consists of an inexact Newton method where the Jacobian J_{FI} is replaced with P_{QN}, see e.g. [FM05, Tez04, Dep04].

A factorization for P_{QN} reads

$$P_{QN} = \left(\begin{array}{c|cc|c} D_{\mathbf{u}_f}\mathcal{C} & \mathcal{O} & \begin{matrix}0\\I\end{matrix} & \mathcal{O} \\ \hline \mathcal{O} & D_{\widehat{\mathbf{d}}_s}\mathcal{N} & \begin{matrix}0\\-I\end{matrix} & \mathcal{O} \\ \hline \begin{matrix}0 & I\end{matrix} & \begin{matrix}0 & -I/\delta t\end{matrix} & 0 & \mathcal{O} \\ \hline \mathcal{O} & \begin{matrix}0 & 0\\0 & -I\end{matrix} & \begin{matrix}0\\0\end{matrix} & \mathcal{H} \end{array}\right) =$$

$$\underbrace{\left(\begin{array}{c|cc|c} D_{\mathbf{u}_f}\mathcal{C} & \mathcal{O} & \begin{matrix}0\\I\end{matrix} & \mathcal{O} \\ \hline \mathcal{O} & D_{\widehat{\mathbf{d}}_s}\mathcal{N} & \begin{matrix}0\\-I\end{matrix} & \mathcal{O} \\ \hline \begin{matrix}0 & I\end{matrix} & \begin{matrix}0 & -I/\delta t\end{matrix} & 0 & \mathcal{O} \\ \hline \mathcal{O} & \mathcal{O} & \begin{matrix}0\\0\end{matrix} & \mathcal{I} \end{array}\right)}_{P_{FS,1}} \underbrace{\left(\begin{array}{c|cc|c} \mathcal{I} & \mathcal{O} & \begin{matrix}0\\0\end{matrix} & \mathcal{O} \\ \hline \mathcal{O} & \mathcal{I} & \begin{matrix}0\\0\end{matrix} & \mathcal{O} \\ \hline \begin{matrix}0 & 0\end{matrix} & \begin{matrix}0 & 0\end{matrix} & I & \mathcal{O} \\ \hline \mathcal{O} & \begin{matrix}0 & 0\\0 & -I\end{matrix} & \begin{matrix}0\\0\end{matrix} & \mathcal{H} \end{array}\right)}_{P_{H,2}}. \tag{4.2.19}$$

$P_{FS,1}$ has a structure similar to the GCE matrix in (2.6.2). One can still split this factor using the same approximations as for the GCE matrix, however a better preconditioner can be devised by neglecting the highlighted term $-I$ in (4.2.18). In this way we find a triple

block factorization P_{DN} for the matrix J_{FI}.

$$P_{DN} = \left(\begin{array}{c|c|c|c} D_{\mathbf{u}_f}\mathcal{C} & \mathcal{O} & \begin{matrix}0\\ I\end{matrix} & D_{\widehat{\mathbf{d}}_f}\mathcal{C} \\ \hline \mathcal{O} & D_{\widehat{\mathbf{d}}_s}\mathcal{N} & \begin{matrix}0\\ 0\end{matrix} & \mathcal{O} \\ \hline \begin{matrix}0 & I\end{matrix} & \begin{matrix}0 & -I/\delta t\end{matrix} & 0 & \mathcal{O} \\ \hline \mathcal{O} & \begin{matrix}0 & 0\\ 0 & -I\end{matrix} & \begin{matrix}0\\ 0\end{matrix} & \mathcal{H} \end{array}\right) =$$

$$\underbrace{\left(\begin{array}{c|c|c|c} \mathcal{I} & \mathcal{O} & \begin{matrix}0\\ 0\end{matrix} & \mathcal{O} \\ \hline \mathcal{O} & D_{\widehat{\mathbf{d}}_s}\mathcal{N} & \begin{matrix}0\\ 0\end{matrix} & \mathcal{O} \\ \hline \begin{matrix}0 & 0\end{matrix} & \begin{matrix}0 & 0\end{matrix} & I & \mathcal{O} \\ \hline \mathcal{O} & \mathcal{O} & \begin{matrix}0\\ 0\end{matrix} & \mathcal{I} \end{array}\right)}_{P_{S,1}} \underbrace{\left(\begin{array}{c|c|c|c} \mathcal{I} & \mathcal{O} & \begin{matrix}0\\ 0\end{matrix} & \mathcal{O} \\ \hline \mathcal{O} & \mathcal{I} & \begin{matrix}0\\ 0\end{matrix} & \mathcal{O} \\ \hline \begin{matrix}0 & 0\end{matrix} & \begin{matrix}0 & 0\end{matrix} & I & \mathcal{O} \\ \hline \mathcal{O} & \begin{matrix}0 & 0\\ 0 & -I\end{matrix} & \begin{matrix}0\\ 0\end{matrix} & \mathcal{H} \end{array}\right)}_{P_{H,2}}$$

$$\underbrace{\left(\begin{array}{c|c|c|c} D_{\mathbf{u}_f}\mathcal{C} & \mathcal{O} & \begin{matrix}0\\ I\end{matrix} & D_{\widehat{\mathbf{d}}_f}\mathcal{C} \\ \hline \mathcal{O} & \mathcal{I} & \begin{matrix}0\\ 0\end{matrix} & \mathcal{O} \\ \hline \begin{matrix}0 & I\end{matrix} & \begin{matrix}0 & -I/\delta t\end{matrix} & 0 & \mathcal{O} \\ \hline \mathcal{O} & \mathcal{O} & \begin{matrix}0\\ 0\end{matrix} & \mathcal{I} \end{array}\right)}_{P_{F,3}}. \tag{4.2.20}$$

We can thus construct the preconditioner in a modular way, separating the FS block from the harmonic extension problem, or also separating all the three fields.

The last preconditioner (4.2.20) is used in many occasions in Part III, where the numerical results are retrieved. In particular its computational efficiency comes from the fact that on one hand the cost of its computation corresponds to the cost of computing a preconditioner for the fluid problem (if the solid problem is linear the factors corresponding to solid and to the geometry problems are constants for all the time levels). On the other hand the approximation of the full Jacobian matrix (4.2.18) is the same as the one used in (4.2.6) to obtain $P^{(1)}_{DN}$, thus the number of iterations performed is also close to the preconditioned GCE case.

4.3 Parallel Preconditioners for FSI

The preconditioners devised in the previous section can be used for FSI applications directly as we introduced them. However in practice their exact factorization is usually not computed explicitly. Indeed, when using a linear iterative solver, at each iteration we need to solve the system $P^{-1}\mathbf{X}$. This product can be computed in the cases discussed by means of one (or more) solve per subdomain problem, which are usually performed using another nested iterative solver. This is mainly due to the fact that the common direct solvers are either serial,

and thus require too much memory and time resources on large problems, or are not scalable in parallel, which produces a bottleneck in the simulation. To avoid these nested iterative solutions one possibility is to approximate the subdomain blocks with other blocks whose factorization requires neither subiterations nor parallel direct solutions. These approximations are scalable preconditioners for the subproblems, which need to be computed in parallel and which should not increase much the conditioning of the global preconditioned system. The factorizations described above are used in this section to build the parallel preconditioners implemented in our code (some of them are reported also in [CDFQ11]).

4.3.1 State of the Art

The research in the field of parallel preconditioning strategies is both challenging and necessary for most of the applications in numerical analysis. Theoretical important results have been achieved during the past decades, providing bounds for the condition number of the preconditioned systems. However the problem is still open, e.g. for indefinite problems the theory is not so developed as for symmetric positive definite (s.p.d.) problems. In this section all the estimates reported indeed refer to the case of a s.p.d. matrix coming from the discretization of a linear second order elliptic problem in 3D.

On the applications side there have been many recent advances as well. The availability of massively parallel architectures have pushed the software towards the development of highly scalable libraries for linear algebra computations. Domain decomposition and multigrid techniques have become standard approaches to handle parallelism, heuristic algorithms and coloring techniques have been devised for the domain partition in order to achieve an effective load balance between the processors and to minimize the communication [CP08, SKK02, CBD$^+$07]. Domain decomposition and multigrid preconditioners have been tested on a large variety of applications and they have shown to be effective for many FE computations (see e.g. [BCA$^+$10, LSS$^+$09, Sal06, KR10, TF01, PW11] and references therein). Many software packages allow to build these preconditioners from the system matrix, without using geometrical informations. The preconditioners built in this way are called *algebraic* (while the classical ones are *geometrical*), and are the most used in practical applications, especially when the geometry does not allow a structured partition in subdomains. Most of these preconditioners rely on a solution of one or more *coarse problems*, which can also be built using only informations at the algebraic level e.g. through *smoothed aggregation* techniques, see e.g. [TW05, Ch.3.10] [VMB96].

In this subsection we report theoretical estimates for the condition number of some of the most popular preconditioners used in parallel solvers for FSI. We focus in particular on domain decomposition preconditioners referring to [TW05, QV99] for a more detailed description.

Domain decomposition methods can be roughly divided into overlapping and iterative substructuring. The formers (e.g. Schwarz methods) imply a partition in overlapping subdomains which covers the original domain, while the latter (e.g. Dirichlet–Neumann, Neumann–Neumann, FETI, BDDC) involve the partition in non-overlapping subdomains, and the restriction of the subproblems defined on each subdomain to their boundary through Schur complements. A domain decomposition method is called *one level* when no coarse grid component is present, i.e., there is no *information exchange* between non neighboring subdomains. It is unavoidable that the condition number of a one level preconditioner P_{1l} is subject to a bound like $K({P_{1l}}^{-1}A) \leq O\left(\frac{1}{\delta^2 H^2}\right)$ (one–level additive Schwarz preconditioners applied to a Poisson problem), where A is the finite elements system matrix, δ is the thickness of the

overlap and H is the maximum diameter of the subdomains. The coefficient $\frac{1}{H^2}$ is due to the fact that there is no exchange of information between two subdomains which are not adjacent, and it is related to the fact that we consider a second order elliptic differential operator. This is similar to the condition number of the stiffness matrix in finite elements, where the dependence on $\frac{1}{h^2}$ can be seen as a consequence of a scaling argument [TW05].

To avoid this dependence on $\frac{1}{H^2}$ it is sufficient to add a *coarse problem* component to the preconditioner, subject to a set of assumptions. This and similar ideas were devised at the end of the eighties, among others by Bramble, Pasciak, Dryja, Wildlund. The bound for the condition number becomes

$$K({P_{2l}}^{-1}A) \leq O\left(1 + \left(\log \frac{H}{h}\right)^2\right),$$

where P_{2l} is an iterative substructuring preconditioner without overlap like FETI, Neumann–Neumann, BDDC. For the two-levels additive overlapping Schwarz preconditioners the estimate becomes

$$K({P_{2l}}^{-1}A) \leq O\left(1 + \frac{H}{\delta}\right).$$

Since the coarse problem is usually solved by a direct method, and since due to its limited size its solution is generally fast, in many parallel implementations every processor solves the coarse problem independently, which avoids the inter-processors communication. However sometimes in massively parallel computations also the coarse problem is too large to be solved serially on every processor. In these cases the partitioning can be performed recursively on the coarse problems, generating a multilevel preconditioner in which only the coarsest level is solved directly on each core.

A very popular choice consists in the algebraic domain decomposition preconditioners. In this case the coarse problem is computed through an algebraic interpolation-restriction operator which has to be defined for each level. A well known algorithm to generate the coarse space is the *smoothed aggregation* technique developed by Vaněk, Brenzina and coauthors at the end of the nineties [VMB96]. For these methods, applied to the multilevel overlapping Schwarz preconditioner, the same bound for the condition number holds as in the case of geometrical coarse problem [Sar02].

An "alternative" preconditioning strategy which is very common and effective consists in using multigrid preconditioners. As for the Schwarz strategy, the multigrid method can usually be interpreted as a preconditioned Richardson method, and the *multigrid preconditioner* can be used in other iterative schemes such as GMRES. It consists of a multilevel strategy, in which each level is solved inexactly, and a prescribed number of iterations of a chosen iterative method (*smoother*) is performed at each level. In the coarsest level the problem is usually solved exactly with a direct method. The multigrid strategy is usually tuned for the specific application, featuring V or W cycles, pre or post smoothing, different smoothers and aggregation techniques. We refrain from discussing all the possibilities, and we refer to [KM09] for a survey on this subject and for an overview of a similar preconditioning technique (Algebraic Multilevel Iteration, AMLI, first introduced by Axelsson and Vassilevski [AV89]), which can be seen as a generalization of the algebraic multigrid (AMG) preconditioner.

The algebraic multigrid method has been popularized by Ruge and Stüben in 1987 [RS87], while an algebraic multigrid preconditioner was used recently in e.g. [PS09, LSS$^+$09], see also [Not10] and references therein for a spectral analysis.

In [BC10b] a 2D FSI solution obtained using an additive Schwarz preconditioner with and without the coarse problem is compared. There the presence of a coarse problem seems to greatly improve the condition number of the FSI system. In that case the weak scalability is tested: the size of the problem is increased with the number of processors by refining the mesh.

A multigrid preconditioner for FSI problems is described in [GKW10]. There the convergence rate is tested with respect to the space discretization, showing independence of h for the preconditioner considered.

Remark 4.3.1. *Although domain decomposition and multigrid preconditioners have a different origin, the difference between them is small. Indeed if a software implementation is general enough it can embed both the strategies within the same code (see e.g. the ML package in Trilinos*[6]*).*

The preconditioners that we use in the next chapter are based on the algebraic additive Schwarz preconditioner (AAS) that we briefly overview here, referring to the aforementioned literature for the details. We consider the generic linear system $A\mathbf{X} = \mathbf{b}$ (e.g. the discretization of system (3.1.2)). We call $V \subseteq \mathbb{R}^n$ the discrete space in which the solution of the linear system is defined. Following the notations used in [CK02], we introduce the set of indexes $S = \{1 \dots n\}$ representing the degrees of freedom of the system. Given an overlapping partition of S in I subsets, $\{S_i\}_{1 \le i \le I}$, we define the subspaces $V_i \subseteq V$ as $V_i = \{\mathbf{X} = (v_1, \dots v_n)^T \in \mathbb{R}^n | v_k = 0 \text{ if } k \notin S_i\}$ of dimension n_i. Next we introduce the restriction matrix $R_i \in \mathbb{R}^{n_i \times n}$ such that $(R_i)_{lj}\delta_{jk} = \delta_{lk}$, $l \in S_i$, $k, j \in S$ and δ is the Kronecker symbol. The prolongation matrix $R_i^T \in \mathbb{R}^{n \times n_i}$ transforms a short vector of n_i components into one with n components by keeping the original components and setting to zero the new ones. In this framework we define the one level AAS preconditioner associated to A as

$$P_{AS}(A)^{-1} = \sum_{i=1}^{I} R_i^T (R_i A R_i^T)^{-1} R_i. \tag{4.3.1}$$

As mentioned above it is possible to add a coarse component (corresponding to $i = 0$) to improve the condition number when the number of subdomains increases. The parallel structure of our code[7] includes a mesh partitioner based on the library ParMETIS, AAS and AMG preconditioners handled by the IFPACK and ML packages embedded in the Trilinos library.

4.3.2 Composed Preconditioners for Geometry–Convective Explicit FSI

In this section we develop the block–representation of the preconditioners for the matrix A in (2.7.12). This section links the study of the block preconditioner carried out in Section §4.2 to the classical domain decomposition preconditioners described above. The composition of the two strategies leads to the parallel preconditioners that we developed and that are tested and applied in the third part of this work.

The approach that we advocate consists in substituting the fluid and solid factors with suitable preconditioners, whose factorization can be computed in parallel. For example, with the notations (4.2.6), instead of $P_{DN}^{(1)} = P_{S,1}^{(1)} P_{F,2}^{(1)}$ or $P_{DN}^{(2)} = P_{F,1}^{(2)} P_{S,2}^{(2)}$, we use, respectively,

[6] `http://trilinos.sandia.gov`

[7] LifeV, `http://www.lifev.org`

$P_{AS-DN1} = P_{AS}(P^{(1)}_{S,1})P_{AS}(P^{(1)}_{F,2})$ or $P_{AS-DN2} = P_{AS}(P^{(2)}_{F,1})P_{AS}(P^{(2)}_{S,2})$. According to the domain decomposition terminology this choice corresponds to employ inexact solvers for the solid and fluid subdomains. A spectral analysis for a similar kind of block triangular preconditioners for stabilized saddle point problems, with symmetric positive definite diagonal blocks, is carried out e.g. in [Kla98, Sim04], while in [GKW10] a similar strategy is used in FSI context, where the blocks corresponding to all three fields are substituted with AMG preconditioners. With this approach we avoid the inner iterations, since we can solve the local inexact problems by LU factorization.

A numerical comparison of the following preconditioning techniques is presented in Section §6.2 for the GCE time discretized system:

1. one–level AAS preconditioner built using the matrix A in (2.7.12): $P_{AS}(A)$;
2. one–level AAS preconditioner built using an approximation of the type (4.2.4), obtained by neglecting the block $-I$ in matrix (2.7.12): P_{AS-DN1};
3. one–level AAS preconditioner built using a different block GS approximation (4.2.5), obtained neglecting the term $-I/\delta t$ in (2.7.12): P_{AS-DN2}.

The preconditioners P_{AS-DN1} and P_{AS-DN2}, besides preserving the modularity in their construction, have similar or better behavior than $P_{AS}(A)$ when increasing the number of processors. Furthermore, their factorization is cheaper in terms of computational time and memory usage than building the factorization of the whole matrix A. In our framework, one can choose different preconditioning techniques for the different sub-blocks, which is appealing for multiphysics systems, since physics-specific preconditioners can be used.

4.3.3 Composed Preconditioners for Geometry Implicit FSI

For what concerns the fully implicit system, we consider in the following four different preconditioners:

- $P_{AS}(J_{FI})$;
- $P_{AS-QN} = P_{AS}(P_{QN})$;
- $P_{AS-GS} = P_{AS}(P_{FS,1})P_{AS}(P_{H,2})$ where the factorization is the one reported in (4.2.19);
- $P_{AS-DN} = P_{AS}(P_{S,1})P_{AS}(P_{H,2})P_{AS}(P_{F,3})$, with the factorization reported in (4.2.20).

By using a linear form for the convective term, i.e., choosing the convective explicit time discretization, we can isolate the geometrical nonlinearity and investigate its effect on the precision and stability of the system. Furthermore in our applications we have observed that the behavior of the same preconditioner in the CE and FI cases is not very different. As discussed extensively in the previous chapter the geometrical nonlinearity can be solved e.g. with fixed–point iterations, inexact or exact Newton. The latter approach involves the computation of the shape derivatives which are reported in Section §3.4.1. We remark that in case of linear elasticity for the solid problem building P_{AS-DN} is as cheap as one fluid preconditioner computation, since the factors related to the harmonic extension problem and to the solid problem are constant throughout the whole simulation and thus can be reused at every time step.

4.3.4 Spectral Analysis

Let us address the general framework of a 4-blocks matrix

$$A = \begin{pmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{pmatrix}, \tag{4.3.2}$$

where the block lines correspond to different coupled problems, the coupling being expressed by the matrices A_{12} and A_{21}. The only a-priori assumption on the matrices A, A_{11} and A_{22} is that they are invertible, therefore the following analysis holds true for any linear system whose matrix has a 4-blocks structure like (4.3.2) with nonsingular diagonal blocks.

The idea is to replace the diagonal blocks A_{11} and A_{22} of the preconditioner P_{GS} (4.1.9) by suitable preconditioners P_1 and P_2. In the following we estimate the influence of this approximation on the conditioning of the system. To this end we suppose that we have an estimate for the condition number of the two preconditioned blocks. Calling $\sigma_A^{\max} = \max_\sigma(\sigma(A))$ and $\sigma_A^{\min} = \min_\sigma(\sigma(A))$ respectively the maximum and the minimum singular values for the matrix A, then for $i \in \{1, 2\}$

$$\delta_i = K(P_i^{-1}A_i) = \sigma^{\max}_{P_i^{-1}A_i} / \sigma^{\min}_{P_i^{-1}A_i}. \tag{4.3.3}$$

We introduce a preconditioner of the form

$$P = \begin{pmatrix} P_1 & 0 \\ A_{12} & \alpha P_2 \end{pmatrix}, \tag{4.3.4}$$

where α is an arbitrary positive scalar. In order to bound the condition number of the matrix $P^{-1}A$ we compute explicitly P^{-1} as follows

$$P^{-1} = \begin{pmatrix} P_1^{-1} & 0 \\ -\frac{1}{\alpha}P_2^{-1}A_{12}P_1^{-1} & \frac{1}{\alpha}P_2^{-1} \end{pmatrix}.$$

Then the preconditioned system can be factored as

$$P^{-1}A = \begin{pmatrix} I & 0 \\ \Sigma & I \end{pmatrix} \begin{pmatrix} P_1^{-1} & 0 \\ 0 & \frac{1}{\alpha}P_2^{-1} \end{pmatrix} \begin{pmatrix} A_{11} & A_{21} \\ 0 & S \end{pmatrix}, \tag{4.3.5}$$

where $\Sigma = \frac{1}{\alpha}P_2^{-1}A_{12}(A_1^{-1}P_1 - I)$.

An upper bound of the condition number for the preconditioned system can be obtained using the inequality $K(AB) \leq K(A)K(B)$. We first find the singular values of the block lower triangular factor (that we note L).

Proposition 4.3.1. *The maximum and minimum singular values of L satisfy*

$$\sigma_L^{\max} = \sqrt{1 + \frac{(\sigma_\Sigma^{\max})^2 + \sqrt{(\sigma_\Sigma^{\max})^4 + 4(\sigma_\Sigma^{\max})^2}}{2}}, \tag{4.3.6}$$

and

$$\sigma_L^{\min} = \sqrt{1 + \frac{(\sigma_\Sigma^{\max})^2 - \sqrt{(\sigma_\Sigma^{\max})^4 + 4(\sigma_\Sigma^{\max})^2}}{2}}. \tag{4.3.7}$$

Proof. By the definition of the singular values we have

$$(\sigma_L^{\max})^2 = \max(\text{eigs}(L^T L)),$$

where

$$L^T L = I + \begin{pmatrix} \Sigma^T\Sigma & \Sigma^T \\ \Sigma & 0 \end{pmatrix}.$$

Let us consider the matrix on the right: its null eigenvalues correspond to unitary eigenvalues of L. Its nonzero eigenvalues can be written in function of the eigenvalues of $\Sigma^T\Sigma$. As a matter of fact, from the definition of eigenvalue,

$$\begin{pmatrix} \Sigma^T\Sigma & \Sigma^T \\ \Sigma & 0 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \lambda \begin{pmatrix} v_1 \\ v_2 \end{pmatrix},$$

by formally substituting v_2, we obtain the following relation

$$\left(1 + \frac{1}{\lambda}\right) \Sigma^T\Sigma v_1 = \lambda v_1,$$

that gives

$$\lambda = \frac{\lambda_{\Sigma^T\Sigma} \pm \sqrt{\lambda^2_{(\Sigma^T\Sigma)} + 4\lambda_{(\Sigma^T\Sigma)}}}{2} = \frac{(\sigma_\Sigma)^2 \pm \sqrt{(\sigma_\Sigma)^4 + 4(\sigma_\Sigma)^2}}{2}.$$

Eventually we get

$$\sigma_L = \sqrt{1 + \frac{(\sigma_\Sigma)^2 \pm \sqrt{(\sigma_\Sigma)^4 + 4(\sigma_\Sigma)^2}}{2}}.$$

The functions $f_\pm(x) = 2 + x \pm \sqrt{x^2 + 4x}$ are positive for $x \geq 0$, in fact we can estimate

$$1 + \frac{(\sigma_\Sigma)^2 - \sqrt{(\sigma_\Sigma)^4 + 4(\sigma_\Sigma)^2}}{2} \geq 1 + \frac{(\sigma_\Sigma)^2 - \sqrt{(\sigma_\Sigma)^4 + 4(\sigma_\Sigma)^2 + 4}}{2} = 1 + \frac{(\sigma_\Sigma)^2 - (\sigma_\Sigma)^2 - 2}{2} = 0.$$

f_+ is increasing and f_- is decreasing; moreover $f_+(x) > f_-(y)$ for all x, y in $\mathbb{R}^+$. This implies the two identities (4.3.6) and (4.3.7).

□

The value σ_L^{min} behaves asymptotically as $\frac{\sqrt{2}}{(\sigma_\Sigma)}$ when $(\sigma_\Sigma \to \infty)$, while clearly $(\sigma_L^{\min} \to 1)$ when $(\sigma_\Sigma \to 0)$. To show that the limit for $(\sigma_\Sigma \to \infty)$ behaves as $\sqrt{2}(\sigma_\Sigma)^{-1}$ it is sufficient to develop the term under squared root in Mc. Laurin expansion for $(\frac{1}{\sigma_\Sigma} \to 0)$:

$$1 + \frac{(\sigma_\Sigma)^2(1 - \sqrt{1 + \frac{4}{(\sigma_\Sigma)^2}})}{2} \approx 1 + \frac{(\sigma_\Sigma)^2(1 - (1 + \frac{1}{2}\frac{4}{(\sigma_\Sigma)^2} - \frac{4}{(\sigma_\Sigma)^4}))}{2} = \frac{2}{(\sigma_\Sigma)^2},$$

we can thus assert that the minimum value of σ_L is obtained for $\sigma_\Sigma = \sigma_\Sigma^{\max}$.

Using some standard algebra we can write

$$\left(\frac{\sigma_L^{\max}}{\sigma_L^{\min}}\right)^2 = \left(\frac{\sqrt{1 + \frac{(\sigma_\Sigma^{\max})^2 + \sqrt{(\sigma_\Sigma^{\max})^4 + 4(\sigma_\Sigma^{\max})^2}}{2}}}{\sqrt{1 + \frac{(\sigma_\Sigma^{\max})^2 - \sqrt{(\sigma_\Sigma^{\max})^4 + 4(\sigma_\Sigma^{\max})^2}}{2}}}\right)^2 =$$

$$\frac{2+(\sigma_\Sigma^{\max})^2+\sigma_\Sigma^{\max}\sqrt{(\sigma_\Sigma^{\max})^2+4}}{2+(\sigma_\Sigma^{\max})^2-\sigma_\Sigma^{\max}\sqrt{(\sigma_\Sigma^{\max})^2+4}} =$$
$$\frac{(2+(\sigma_\Sigma^{\max})^2+\sigma_\Sigma^{\max}\sqrt{(\sigma_\Sigma^{\max})^2+4})^2}{(2+(\sigma_\Sigma^{\max})^2)^2-(\sigma_\Sigma^{\max})^2((\sigma_\Sigma^{\max})^2+4)} =$$
$$\left(\frac{2+(\sigma_\Sigma^{\max})^2+\sigma_\Sigma^{\max}\sqrt{(\sigma_\Sigma^{\max})^2+4}}{2}\right)^2.$$

Thus the condition number of the factor L can be expressed as:

$$K(L)=\frac{\sigma_L^{\max}}{\sigma_L^{\min}}=\frac{2+(\sigma_\Sigma^{\max})^2+\sqrt{(\sigma_\Sigma^{\max})^4+4(\sigma_\Sigma^{\max})^2}}{2}\equiv\kappa(\sigma_\Sigma^{\max}),$$

which means that the condition number of L only depends on the maximum singular value of $\Sigma=\frac{1}{\alpha}P_2^{-1}A_{12}(A_1^{-1}P_1-I)$. Note that good conditioning of L strongly depends on how well $A_1^{-1}P_1$ approaches the identity.

We now rewrite the block diagonal and upper triangular factors in (4.3.5) as

$$\begin{pmatrix} P_1^{-1}A_{11} & 0 \\ 0 & \frac{1}{\alpha}P_2^{-1}A_{22}\end{pmatrix}\begin{pmatrix} {A_{11}}^{-1} & 0 \\ 0 & {A_{22}}^{-1}\end{pmatrix}\begin{pmatrix} A_{11} & A_{21} \\ 0 & S\end{pmatrix}=\begin{pmatrix} P_1^{-1}A_{11} & 0 \\ 0 & \frac{1}{\alpha}P_2^{-1}A_{22}\end{pmatrix}P_{GS}^{-1}A,$$

whose condition number is bounded by

$$\frac{\max\{\sigma^{\max}_{P_1^{-1}A_{11}},\frac{1}{\alpha}\sigma^{\max}_{P_2^{-1}A_{22}}\}}{\min\{\sigma^{\min}_{P_1^{-1}A_{11}},\frac{1}{\alpha}\sigma^{\min}_{P_2^{-1}A_{22}}\}}\,\delta_{GS}, \tag{4.3.8}$$

where δ_{GS} is the condition number of the preconditioned matrix (4.1.10)

$$\delta_{GS}=K(P_{GS}^{-1}A). \tag{4.3.9}$$

As a result we obtain a bound for the condition number of the preconditioned linear system

$$K(P^{-1}A)\le\kappa(\sigma_\Sigma^{\max})\frac{\max\{\sigma^{\max}_{P_1^{-1}A_{11}},\frac{1}{\alpha}\sigma^{\max}_{P_2^{-1}A_{22}}\}}{\min\{\sigma^{\min}_{P_1^{-1}A_{11}},\frac{1}{\alpha}\sigma^{\min}_{P_2^{-1}A_{22}}\}}\,\delta_{GS}. \tag{4.3.10a}$$

If the maximum singular values of the preconditioned sub-blocks ($\sigma^{\max}_{P_1^{-1}A_{11}}$ and $\sigma^{\max}_{P_2^{-1}A_{22}}$) are available, we can improve this estimate by appropriately defining the scalar α. In particular, if $\alpha=\sigma^{\max}_{P_2^{-1}A_{22}}/\sigma^{\max}_{P_1^{-1}A_{11}}$ then (4.3.8) simplifies to $\max\{\delta_1,\delta_2\}\delta_{GS}$, where δ_1 and δ_2 are defined by (4.3.3). Note that the maximum singular value $\kappa(\sigma_\Sigma^{\max})$ of $\Sigma=\frac{1}{\alpha}P_2^{-1}A_{12}(A_1^{-1}P_1-I)$ depends on α; therefore the rescaling is useful if either $\alpha>1$ (i.e., $\sigma^{\max}_{P_2^{-1}A_{22}}>\sigma^{\max}_{P_1^{-1}A_{11}}$), or if the gain in bounding the factor (4.3.8) justifies the loss in $\kappa(\sigma_\Sigma^{\max})$.

The rescaling by α obviously does not affect the quantities δ_1 and δ_2, thus the final condition number estimate reads

$$K(P^{-1}A)\le\kappa(\sigma_\Sigma^{\max})\max\{\delta_1,\delta_2\}\delta_{GS}. \tag{4.3.10b}$$

Estimate (4.3.10b) shows that we have to choose the preconditioners for the diagonal blocks according to the following criteria:

- P_1 and P_2 are well suited preconditioners for the matrices A_{11}, A_{22};
- ${A_{11}}^{-1}P_1$ is near the identity, such that $\kappa(\sigma_\Sigma^{\max})$ is small;

and that we should define α as an approximate ratio of the maximum singular values of the preconditioned sub-systems. The same estimate also suggests that in some circumstances the roles of A_{11} and A_{22} should be interchanged.

Apparently, from the form of σ we would expect that a large α would improve the conditioning of the system. However, we can note that for α larger than $\sigma^{\max}_{P_2^{-1}A_{22}}/\sigma^{\max}_{P_1^{-1}A_{11}}$ and than $\sigma^{\min}_{P_2^{-1}A_{22}}/\sigma^{\min}_{P_1^{-1}A_{11}}$, (4.3.10b) becomes

$$K(P^{-1}A) \leq \kappa(\sigma_\Sigma^{\max})\frac{\sigma^{\min}_{P_2^{-1}A_{22}}}{\sigma^{\min}_{P_1^{-1}A_{11}}}\alpha\delta_{GS}. \tag{4.3.10ac}$$

Therefore for large α the condition number get worse.

Thanks to this analysis, we are able to build parallel preconditioners of the coupled problem based only on the subproblems. Of utmost importance, the scalability of our problem depends on the coupling only through σ_Σ^{max}. Thus if we find a block GS preconditioner for which σ_Σ^{max} is bounded the scalability will not depend on the coupling, but only on the scalability properties of the sub-problems.

To apply Proposition 4.3.1 to the FSI block matrix of the previous section we need the following proposition.

Remark 4.3.2. *We assume here that the restriction–prolongation matrices R_i and R_i^T are built in such way that in the nodes lying in the overlapping region the sum is weighted by the number of overlapping subdomains in that point. In this way the matrices having only one nonzero element per row are invariant under P_{AS}.*

Proposition 4.3.2. *Consider the matrix $A = \begin{pmatrix} A_{11} & A_{21} \\ A_{12} & A_{22} \end{pmatrix}$, and the preconditioner $P = \begin{pmatrix} A_{11} & 0 \\ A_{12} & A_{22} \end{pmatrix}$, where A_{12} is a rectangular matrix with at most one nonzero element per row. If we call P_{AS} the AAS operator (4.3.1), then*

$$P_{AS}\left(\begin{pmatrix} A_{11} & 0 \\ A_{12} & I \end{pmatrix}\right) P_{AS}\left(\begin{pmatrix} I & 0 \\ 0 & A_{22} \end{pmatrix}\right) = \begin{pmatrix} P_{AS}(A_{11}) & 0 \\ A_{12} & P_{AS}(A_{22}) \end{pmatrix} \tag{4.3.11}$$

Proof. Let us first recall the way in which the overlap in the AAS preconditioner is computed. The neighbors of a d.o.f. i are the d.o.f. corresponding to the columns j for which the matrix elements (i, j) are different from zero.

We consider the first factor in (4.3.11) (being the second factor block-diagonal), that we denote $P_{AS}(P_1)$. Using the notation introduced for the description of AAS preconditioners we get

$$\left(P_{AS}\begin{pmatrix} A_{11} & 0 \\ A_{12} & I \end{pmatrix}\right)^{-1} = \sum_{i=1}^{n} R_i^T (R_i P_1 R_i^T)^{-1} R_i. \tag{4.3.12}$$

The local matrix $R_i P_1 R_i^T$ has the form

$$P_i = \begin{pmatrix} A_{11i} & 0 \\ A_{12i} & I_i \end{pmatrix}.$$

We can formally write its inverse

$$P_i^{-1} = \begin{pmatrix} {A_{11}}_i^{-1} & 0 \\ -A_{12i}{A_{11}}_i^{-1} & I_i \end{pmatrix} = \begin{pmatrix} {A_{11}}_i^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ -A_{12i}{A_{11}}_i^{-1} & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & I_i \end{pmatrix}.$$

Thus we can rewrite (4.3.12) as

$$\left(P_{AS}\begin{pmatrix} A_{11} & 0 \\ A_{12} & I \end{pmatrix}\right)^{-1} =$$

$$\sum_{i=1}^{n} R_i^T \begin{pmatrix} {A_{11}}_i^{-1} & 0 \\ 0 & 0 \end{pmatrix} R_i + \sum_{i=1}^{n} R_i^T \begin{pmatrix} 0 & 0 \\ -A_{12i}{A_{11}}_i^{-1} & 0 \end{pmatrix} R_i + \sum_{i=1}^{n} R_i^T \begin{pmatrix} 0 & 0 \\ 0 & I_i \end{pmatrix} R_i.$$

Notice that due to the fact that A_{12} has at most one nonzero per row

$$\sum_{i=1}^{n} R_i^T \begin{pmatrix} 0 & 0 \\ -A_{12i}{A_{11}}_i^{-1} & 0 \end{pmatrix} R_i = \sum_{i=1}^{n} R_i^T \begin{pmatrix} 0 & 0 \\ -A_{12i} & 0 \end{pmatrix} R_i R_i^T \begin{pmatrix} 0 & 0 \\ 0 & {A_{11}}_i^{-1} \end{pmatrix} R_i =$$

$$= \sum_{i=1}^{n} \begin{pmatrix} 0 & 0 \\ -A_{12} & 0 \end{pmatrix} R_i^T \begin{pmatrix} 0 & 0 \\ 0 & {A_{11}}_i^{-1} \end{pmatrix} R_i.$$

This leads eventually to

$$\left(P_{AS}\begin{pmatrix} A_{11} & 0 \\ A_{12} & I \end{pmatrix}\right)^{-1} = \begin{pmatrix} P_{AS}^{-1}(A_{11}) & 0 \\ -A_{12}P_{AS}^{-1}(A_{11}) & I \end{pmatrix} = \begin{pmatrix} P_{AS}(A_{11}) & 0 \\ A_{12} & I \end{pmatrix}^{-1}.$$

From the last identity we have that

$$P_{AS}\left(\begin{pmatrix} A_{11} & 0 \\ A_{12} & I \end{pmatrix}\right) = \begin{pmatrix} P_{AS}(A_{11}) & 0 \\ A_{12} & I \end{pmatrix}.$$

Substituting in (4.3.11) we obtain the thesis. □

Part III

Applications and Simulations

Applications to Hemodynamics

In this chapter we focus on the main application driving our development of the FSI solvers described in the previous chapters and developed in the C++ finite element library LifeV[8]. Our aim is to simulate blood flow in large compliant arteries under both physiological and pathological conditions.

In Section §5.1 we give a brief description of the circulation and of the most common cardiovascular diseases with their possible treatments. In Section §5.2 we give some motivations and references for the study of FSI in arterial networks and we describe the pipeline used to obtain FSI simulations from medical raw data. In Section §5.3 we illustrate different approaches to obtain a high quality mesh, we emphasize the importance of building a boundary layer mesh to obtain accurate values for the wall shear stress even with coarse meshes, and we report a simulation of blood flow in a femoropopliteal bypass. In Section §5.4 we describe a physiological FSI simulation of blood flow in the aortic arch. The choice of appropriate boundary conditions and the effect of modelling the compliance of the arterial wall are discussed. Different methods to solve the nonlinearity and different kinds of time discretizations described in Part II are compared in Section §5.5.

5.1 Applications and Motivation

The goal of this PhD work is the simulation of blood flow in large arteries. For reader's convenience, in this section we briefly summarize the basic elements of blood circulation and the principal pathologies related to it. We also give a short account on the mathematical modeling and the associated results that could be obtained by means of our simulations.

In this description of the human circulatory system, cardiovascular diseases and their treatment, we mainly refer to Thiriet [FQV09, Ch.1] and Olufsen et al. [OOL04, Ch.2].

5.1.1 Circulation

The cardiovascular system consists of two parallel circulations: the *systemic* and *pulmonary* ones. The systemic arteries transport the oxygenated blood and nutrients toward all the body, the blood exchanges the oxygen through the capillary networks and returns to the right atrium through the venous system. The pulmonary arteries transport the deoxygenated blood from the right ventricle to the lungs, there the blood exchanges carbon dioxide with oxygen and returns back to the left atrium through the pulmonary veins. A schematic picture representing systemic and pulmonary circulations is reported in Figure 5.1[9]. The arterial walls

[8] `http://www.lifev.org`

[9] Image from `http://www.tutorvista.com/biology/systemic-circulation-diagram`.

of the pulmonary arteries and veins are much thinner than in the systemic circulation, due to the lower pressure in the pulmonary circulation. In the following description we focus on the systemic circulation.

The arteries are in general classified into three categories, although this distinction is sometimes not evident. The first category is the one of large arteries, from approximatively 2.5 cm of diameter down to 0.1 cm. These arteries are characterized by an elastic behaviour of the wall, whose thickness to diameter ratio is approximately constant and whose Young modulus slightly increases when moving distally. The large arteries wall is composed by three layers: intima, a layer containing the endothelial cells which sense and react to the normal and shear stress coming from the fluid; media, a thick elastic layer composed mainly of elastin and collagen fibers and of a variable amount of smooth muscle cells; adventitia, a loose external layer of variable thickness which does not contribute much to the compliance of the wall. Often the boundary between the adventitia and the tissue surrounding the artery is not well defined.

The second category is the one of arterioles, which play a fundamental role in the cardiovascular regulation. In this case although the thickness of the wall decreases with the artery diameter, the thickness to diameter ratio is not constant any more, but it increases when vessels become smaller. The walls of the arterioles is almost rigid and it contains smooth muscle cells. The presence of these muscle cells allows the arterioles to regulate the intraluminal pressure and radius, in order to meet the needs of the tissue.

The third category of arteries consists of the capillaries. These arteries are very small (few μm), their wall is composed only by an endothelial cells layer, so that nutrient, oxygen and carbon dioxide can be exchanged with the interstitial fluid of the tissue cells. The blood flow at the capillary level is not pulsatile any more: the compliance of the arterial wall has the effect of damping the pulsation of the heart. While the large arteries and arterioles always bifurcate in two, the capillaries can have complicated patterns and bifurcations, also featuring closed rings.

The flow in the veins is not pulsatile, the pressure is lower than in both systemic and pulmonary circulations. The venous wall is thinner than the arterial one, and valves may be present, in order to prevent the flow to go in the wrong direction when the vessel is compressed by a muscle contraction.

Blood is composed mainly by red blood cells (erythrocytes, whose concentration, or hematocrit, ranges from 40 to 45%) and plasma (composed for the 98% of water). The red blood cells have an elastic behaviour, thanks to a cytoskeleton, and can deform when passing through the capillaries[10]. The behaviour of blood is found to be non Newtonian, since the viscosity depends on the shear. The reason is this that when the shear is low the red blood cells tend to interact, increasing the macroscopic viscosity of blood (the *shear thinning* effect). For high shear (higher than about $1\,\mathrm{s}^{-1}$) there is no interaction and the viscosity of blood in large arteries is approximately constant (around $0.03\,\mathrm{g/(cm{\cdot}s)}$ and $0.04\,\mathrm{g/(cm{\cdot}s)}$). However, also for high shear, the viscosity is found experimentally to drop when the radius of the vessel is smaller than 0.1 cm.

For these and other reasons the macroscopic modeling of blood flow as a Newtonian fluid and of the arterial wall as an elastic structure is justified only for large arteries, while the circulation on arterioles and capillaries should be simulated using other models, accounting

[10] the study of the deformations of the single red blood cells has many medical applications and is subject of research in *microfluidics*.

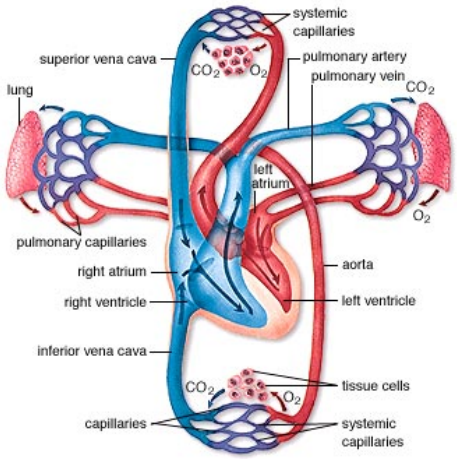


Figure 5.1: Systemic and pulmonary circulation.

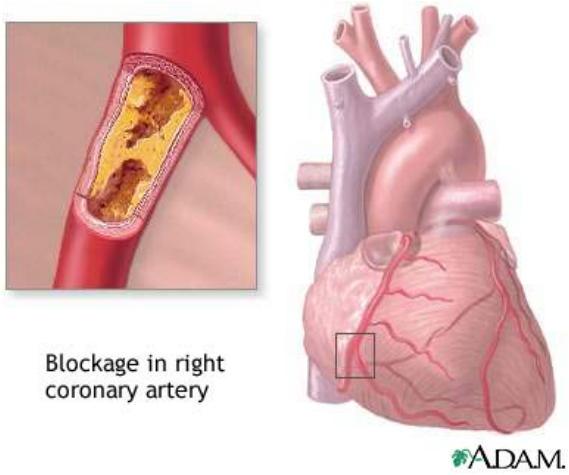


Figure 5.2: Atherosclerosis.

for the aforementioned effects. The circulation in the arterioles and capillaries is often called *peripheral circulation*, and a proper model describing it in detail should take into account the meso and micro scales. If the main focus is on the large arteries simulation, then the peripheral circulation can be accounted for using reduced order models, cf. Section §2.9.

5.1.2 Cardiovascular Diseases

We resume in the following the main cardiovascular diseases with their possible treatment, in order to show how numerical simulations of such pathological situations can potentially help medical doctors in the decision making process. The main cardiovascular diseases affecting large arteries are the following:

- Congenital defects, such as bi-leaflet aortic valve, heart septal defects, pulmonary or tricuspid atresia. These defects often require surgical intervention (e.g. the *Fontan procedure* for pulmonary or tricuspid atresia [FB71]), in this case numerical simulations

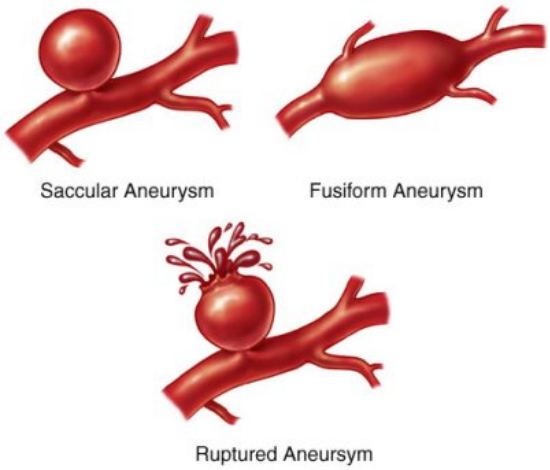


Figure 5.3: Saccular and fusiform aneurysms.

can give a prediction of the hemodynamic features (pressure, flow rate, etc.) pre and post intervention.

- Atherosclerosis, i.e., the accumulation of fatty materials, fibrous elements and calcium on the inner surface of the arterial wall, see Figure 5.2[11]. The formation of atherosclerotic plaques generates a stenosis, and at a late stage can provoke a total stenosis of the artery. Furthermore thrombi can detach from the plaque and lead to infarctions and strokes. Stenosed arteries are in general treated by grafting or stenting, or by removing the plaque. The first method consists in substituting (or bypassing) the stenosed artery using another vessel and creating artificial anastomoses. The second method consists in dilating the arterial wall using a balloon and in inserting a device (stent) which supports the vessel walls, preserving the original lumen. The main problem of this approach is the potential restenosis, which can be limited adopting drug eluting stents inhibiting the cells proliferation in the stented region. Experimental evidence suggests that the wall shear stress is involved in the formation of atherosclerotic plaques. Thus a numerical simulation and a wall shear stress evaluation can help in identifying which locations are more at risk or to choose an optimal strategy for the surgical interventions.

- Aneurysms, i.e., a dilation of an arterial segment. The vessel wall of an aneurysm is much thinner and stiffer than in the physiological case, the collagene fibers being main responsible of the mechanic response. The aneurism rupture causes possibly lethal haemorrages. The aneurysms can be either fusiform or saccular (see Figure 5.3[12]). The main strategies for the treatment of saccular aneurysms are cutting them with a surgical intervention (clipping), or filling them with thin metallic coils, so that most of the aneurysmal cavity is excluded from the circulation. Fusiform aneurysms are usually treated by grafting, or by using external or internal stents. All these treatments could largely benefit from numerical simulations, whose prediction can avoid issues like recanalisation.

[11] Image from A.D.A.M., Inc.
[12] Image from `http://www.daviddarling.info/encyclopedia/A/aneurysm.html`.

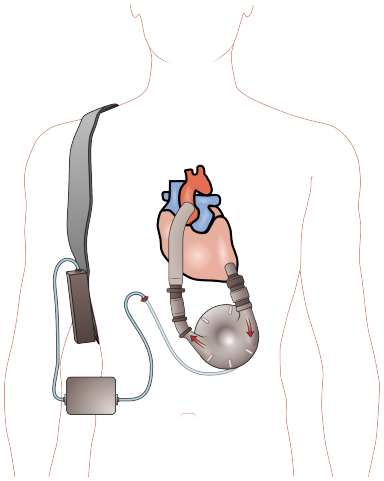


Figure 5.4: Ventricular assisted device.

- Heart failure. Implantation of *ventricular assisted devices* is a reliable treatment for patients with terminal heart failure. These devices are used as a bridge to transplantation, bridge to recovery or destination therapy (for patients who are not eligible for heart transplantation). It consists of an extracorporeal circulation device used to bypass the left ventricle and to pump the oxygenated blood directly into the aorta see Figure 5.4[13]. Frequently the hemodynamic modifications due to this treatment lead to complications that can be investigated using numerical simulations.

5.2 Problem Description

Blood flow dynamics and arterial wall mechanics are thought to be an important factor in the pathogenesis and treatment of cardiovascular diseases. Indeed more and more data become available in clinical routine and, at the same time, numerical methods are also more efficient and capable of reproducing *in silico* complex phenomena. A number of specific hemodynamic and vascular mechanic factors – notably wall shear stress (WSS), pressure and mural stress, flow rate, and residence time – are implicated in aneurysm growth and rupture [BT96, SD00] or in the pathogenesis of atherosclerosis [KWO$^+$97]. Judicious control of these hemodynamic factors may also govern the outcomes of vascular therapies [GCF94, GSP92]. Computational fluid dynamics and computational fluid structure interaction provide a viable option for understanding the complex nature of blood flow and arterial wall mechanics and for obtaining those relevant quantities. A better understanding of the hemodynamics could thus improve the prediction and diagnosis in both healthy and pathological situations [FQV09, Chap. 1]. A simulation with rigid walls fails to predict some essential characteristics of the blood flow (such as pressure wave propagation). Thus it cannot be considered reliable in many situations (e.g. when the vessels undergo relatively large displacements). However considering the full 3D FSI problem increases the computational cost, thus it is of utmost importance to develop and implement parallel and scalable computational algorithms.

[13]Image from Wikipedia `http://en.wikipedia.org/wiki/Ventricular_assist_device`.

Modeling the fluid–structure interaction between the blood and the arterial wall is a challenging task. The research in this field is developing fast concerning both the modeling aspects and computational efficiency. Taking into account the compliance of the vessels can be achieved by introducing a 3D or 2D elastic structure, using a Lagrangian (see e.g. [IOP03]), Eulerian (e.g. [CMM08]), or Arbitrary Lagrangian Eulerian formulation (e.g. [SH07], cf. Chapter 1, in [PR95] this approach was pioneered for hemodynamic applications). The structure model should take into account the nonlinearity due to the collagen activation and the presence of surrounding tissue inducing a static pressure and a dynamic response to the wall displacement. While in many contexts the rheological properties of blood flow have a great influence on its dynamics, the flow in large healthy arteries is known to have an almost Newtonian behavior [FQV09, OOL04].

Alternative options that avoid the introduction of a structure model for the external wall are transpiration techniques (e.g. [FFT00, DFF03]), the coupled momentum method, used e.g. by Kim et al. [KVCF$^+$09], or other similar approaches (e.g Nobile et al. [NV08]). These methods consist of dropping the non-slip Dirichlet condition at the fluid–structure interface, and substituting it with a proper condition that emulates the presence of a surrounding structure. In particular the coupled momentum method shows good results in many physiological situations, and it has the advantage of being computationally cheap because the mesh is fixed. However, although it is well suited for small displacements, this method can be inappropriate when the displacements become large [FVCJ$^+$06]. Furthermore, as the fixed control volume where the fluid equations are solved allows the fluid to pass through the interface, the quantities computed at the boundary, such as the wall shear stress, are subject to a further approximation.

Thus numerical simulations can help to predict the WSS distribution in a specific geometry of the vessel, improving diagnosis and prevention. A reliable numerical tool that carries out all the process from the patient-specific segmentation to the simulation could help a medical doctor, e.g. to identify the regions at risk for pathologies such as atherosclerosis, infarction or aneurisms development [FQV09, Ch. 1]. Furthermore, if the model is validated, a simulation of the WSS distribution can help to solve the inverse problem, i.e., to identify the role that WSS plays in the development of pathologies such as atherosclerosis.

The numerical computations require meshes describing the patient-specific three-dimensional cardiovascular geometry. This geometry needs to be reconstructed from medical images, as explained in the next section.

5.2.1 From DICOM Images to Numerical Simulations

We describe here the set of tools used to build a complete pipeline from raw medical images to FSI simulations. A standard file format for medical images data from e.g. CT scans or MRI is the *DICOM* format. These images can be handled by many open source softwares for visualization (see e.g. OsiriX[14]) or segmentation (see 3D Slicer[15] or vmtk[16]). Although some of these softwares provide advanced tools to detect, extract and smoothen the surfaces of the different organs visible in the raw image, the segmentation step cannot be completely automatic due to the noise present in the images which often produces artifacts that have to be manually removed.

[14] `http://www.osirix-viewer.com/AboutOsiriX.html`
[15] `http://www.slicer.org`
[16] `http://www.vmtk.org`

While the arterial lumen, which is the part of the vessel occupied by the fluid, is usually easy to detect from the medical images, it is much more difficult to accurately extract the vessel wall. For this reason often the geometry of the vessel wall is obtained using empirical algebraic relations available from literature (see e.g. Langewouters et al. [Lan82], cf. Section §5.4) relating the lumen diameter and the vessel wall thickness. However this approach requires the definition and computation of the vessel radius, and thus of a vessel centerline, which is not trivial for an arbitrary, possibly branching, artery. Some software, like vmtk, can handle this (see e.g. [Ant02] for an explanation of the methodology).

Once the vessel wall geometry is available, the finite elements discretization requires the generation of a mesh for both fluid and solid domains. The surface mesh extracted from the raw medical images however is frequently not suitable for FE applications. In these cases a *surface remeshing* step is required, cf. Section §5.3. Both these steps can be performed using open-source softwares, such as Gmsh[17].

Once we have built a tetrahedral mesh conforming at the fluid–structure interface we can use the LifeV library to run the FSI simulation on the patient specific geometry. LifeV is a free C++ parallel finite elements library distributed under LGPL licence, developed at EPFL in Lausanne, at Politecnico di Milano, at Emory University in Atlanta and at INRIA in Paris. Most of the preconditioning strategies and the different algorithms for the FSI problem described in this work have been implemented in the LifeV library as part of this research project.

We make below some general comments, which are valid for all the simulations of this section. LifeV is based on Trilinos[18], a parallel library written in C++ which among other features implements some standard preconditioning strategies, iterative and direct solvers and it acts as interface for the linear algebra packages (BLAS, Lapack, UMFPACK). The parallelization is achieved by partitioning the mesh using the ParMETIS library[19], so that each processor holds a mesh partition and all the vectors and matrices are distributed accordingly. The parallelization is based on the MPI API. For FSI the fluid and solid geometries are partitioned independently, and each processor holds one fluid and one solid partition.

In most of the simulations performed the fluid problem is discretized in space using P1-P1 finite elements stabilized with the interior penalty technique described in [PB05]. The stabilization parameters chosen, cf. Section §2.7.1, are $\gamma_\beta = 1$, $\gamma_{div} = 0.2$ and $\gamma_p = 0.05$. The solid and the geometry problems are discretized in space with P1 finite elements. The discretization in time is based on the implicit Euler method for the fluid (in its non conservative formulation, cf. Chapter 2) and a mid-point second order scheme for the structure (see Chapter 2 or [Nob01, Chap. 4]).

In all of the simulations presented in this chapter, besides the standard Dirichlet or Neumann boundary conditions, less standard *defective flux* conditions are employed. The imposition of flux defective boundary conditions is addressed e.g. in [FGNQ02a, FVV08, VV05], see also Section §2.9, in the rigid walls case. It consists of imposing the mean velocity on a given surface through Lagrange multipliers. The extension to the compliant case is straightforward, the only difference being the fact that in this case the section where the conditions are imposed can change in time. A review and an extension of these methods may be found in Vergara [Ver10].

[17] `http://geuz.org/gmsh`
[18] `http://trilinos.sandia.gov`
[19] `http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview`

5.2.2 Unsteady Blood Flow in a Compliant Iliac Artery

Here we give a concrete example of the pipeline used in our applications for running patient–specific simulations starting from raw medical data. This process involves several softwares that are distributed under a public license, and are constantly updated and improved. The many different steps can be resumed in the following list:

1. The manual segmentation, i.e., the extraction of the surface of the vessel lumen from the medical data (often in DICOM format). This step is perhaps the most critical, since it cannot be completely automated. Many tools can help to carry out the segmentation step, such as vmtk and 3D Slicer. In the current example this step is avoided, since the geometry of the lumen of an iliac artery was already available from the Simtk website[20].

2. The 3D surface extracted is usually composed of triangles. At this stage the methods described in Section §5.3 (see [MCG+11]) can be used to automatically optimize the surface mesh. In the current example the harmonic map algorithm illustrated in Section §5.3 is employed.

3. The centerlines of the lumen are computed using vmtk. This allows to compute the thickness proportional to the intraluminal radius. The extrusion of the vessel wall mesh can present some critical aspects. In fact an extrusion in the normal outward direction may generate inconsistent triangles in proximity of a bifurcation. This issue becomes even more critical when the surface mesh is refined. This situation at present is not detected or solved automatically by most of the softwares, developers of the libraries cited above (vmtk, Gmsh) are currently working to overcome this issue.

4. A grid for the fluid and solid volumes can be generated using Gmsh, optionally adding a boundary layer in the fluid domain (in this particular case the boundary layer is not present).

5. Eventually the FSI simulation can be run using LifeV, with all the possibilities concerning boundary conditions and problem formulations discussed so far. The boundary conditions imposed in this case are defective fluxes on all the inlet and outlets, while a Robin condition is applied on the external wall to simulate the effect of a surrounding tissue (see Equation (5.4.1)).

A picture showing the solution at $t = 0.2\,\mathrm{s}$ is reported in Figure 5.5. The inlet flux imposed was obtained from a previous simulation of an aortic arch, while the outlet fluxes correspond to the inlet value weighted accordingly with the area of each outlet surface (without accounting for the delay due to the travelling pressure wave). This set of boundary conditions is not meant to be physiological, and this subsection should be intended as a proof of concept, while we leave for the rest of the chapter the discussion about cases of physiological interest, comparisons and validation of the code.

[20] https://simtk.org

Figure 5.5: Snapshot of the FSI simulation on the iliac artery (at the time instant $t = 0.2s$). The represented streamlines are colored by the fluid velocity.

5.3 Quality Mesh Generation for Cardiovascular Flow Simulations

In this section we give a brief overview of some algorithms investigated in literature to build high quality meshes for FSI simulations. We focus mainly on two aspects of the mesh generation techniques: the necessity to keep bounded the aspect ratio of the mesh tetrahedra and the importance of simulating the fluid boundary layer. The former improves the algorithm in terms of both the discretization error [BA76] and the condition number [Fri72]. The latter allows the accurate computation of physiological quantities of clinical interest on the FS interface, such as the Wall Shear Stress (WSS). The influence of the mesh quality on the efficiency and accuracy of the numerical methods has been investigated in e.g. [MCG+11, SMS07, BFOG97].

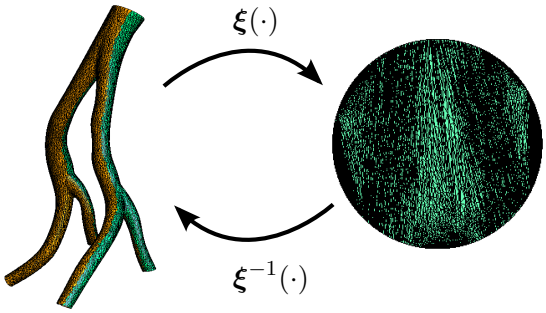


Figure 5.6: Map $\boldsymbol{\xi}$ in case of an iliac artery. In this case a portion of the 3D surface is mapped into a 2D disk to be remeshed.

5.3.1 Surface Remeshing Techniques

From 3D medical images it is usually possible to extract a triangulation of the surface of the vessel. However this surface mesh is often of low quality, oversampled, and featuring very distorted triangles. A remeshing technique is required for simulation purposes. Furthermore many algorithms to build meshes in 3D (e.g. Delaunay, Frontal) keep the original surface mesh on the interface, thus the poor quality of such mesh affects also the quality of the mesh inside the 3D domain. The main tools to handle the surface remeshing are

- mesh adaptation strategies [IN02, BCT02, WHMW07],
- meshing techniques that rely on a suitable surface parametrization [BLG00, LB03].

We concentrate here on the second category, referring to [MCG+11] and references therein for further details. The idea of the parametrization technique is to optimize a 3D surface mesh using algorithms for the 2D meshes. This is achieved by defining a local frame of reference on the surface (e.g. as done in Section §1.1), and thus a parametrization $\boldsymbol{\xi}(\widehat{x})$:

$$\begin{array}{rccc} \boldsymbol{\xi}: & \mathsf{S} \subset \mathbb{R}^3 & \rightarrow & \mathsf{S}' \subset \mathbb{R}^2 \\ & \widehat{x} & \mapsto & \boldsymbol{\xi}(\widehat{x}) \end{array} \tag{5.3.1}$$

that transforms continuously a 3D surface S into a surface S' embedded in $\mathbb{R}^2$.

The map $\boldsymbol{\xi}$ is used to pass from the 3D surface to the 2D plane, the remeshing step is carried out there using a 2D triangulation algorithm and then the mesh is mapped back to the surface embedded in $\mathbb{R}^3$ by means of the inverse of the map $\boldsymbol{\xi}$. The crucial point of this strategy is of course the definition of such a map.

Since our goal is to produce triangular elements as close as possible to equilateral triangles it is desirable for the map $\boldsymbol{\xi}$ to be conformal, so that the triangles with a good aspect ratio in the 2D mesh are mapped to similar triangles in the surface mesh.

We overview here two methods to build this map which are implemented on the free software Gmsh. One method computes $\boldsymbol{\xi}$ by solving two Laplace equations on the original surface triangulation, the other one is based on a least square conformal parametrization of the given triangulated surface S.

As shown by Remacle et al. in [RGCM10] and by Marchandise et al. in [MCdWV+10], both these parameterizations exist if the two surfaces S and S' have he same topology, e.g. if they

have zero genus and have at least one boundary. Moreover in order to prevent the numerical issue of indistinguishable coordinates the geometrical aspect ratio should be moderate, which is often not the case for vascular geometries. In case the topology is wrong and/or if the aspect ratio is too high it can be necessary to call a partitioning algorithm to create different patches of correct topology and moderate aspect ratio [MCW+10]. Those patches are then subsequently remeshed with a finite elements harmonic or least square conformal map.

Harmonic Mapping

We briefly summarize here a method proposed by Remacle et al. [RGCM10] to remesh a surface using harmonic maps. The standard finite elements basis on the surface triangulation is introduced using barycentric coordinates. Two Poisson equations (with proper boundary conditions) are discretized and solved on the surface mesh for the two coordinate functions u and v. This allows to build the map $\boldsymbol{\xi}$ in (5.3.1). The map created in this way is the one minimizing with respect to u and v the quantities

$$E_{\mathrm{H}}(u) = \frac{1}{2}\int_{\mathsf{S}} \|\nabla_{\widehat{x}}(u)\|^2 \, d\gamma, \qquad E_{\mathrm{H}}(v) = \frac{1}{2}\int_{\mathsf{S}} \|\nabla_{\widehat{x}}(v)\|^2 \, d\gamma. \tag{5.3.2}$$

The remeshing then takes place in 2D using a standard algorithm. The information about the metric change is taken into account in the remeshing step by the metric tensor $\mathbf{G}$ (see Section §1.1) associated to the local frame of reference: given that $\boldsymbol{\xi} = \begin{pmatrix} u \\ v \end{pmatrix}$, $\mathbf{G} = (\nabla_{\xi}\boldsymbol{\xi})^T \nabla_{\xi}\boldsymbol{\xi}$ is also the first fundamental form associated to the surface, see [GAS06].

Conformal Mapping

The least square conformal map as introduced by Levy at al. [LPRM02] requires the parametrization $\boldsymbol{\xi} = \begin{pmatrix} u \\ v \end{pmatrix}$ to be such that the gradient of u and the gradient of v are as orthogonal as possible and have the same norm. For a piecewise linear mapping, the least square conformal map can be obtained by minimizing the conformal energy:

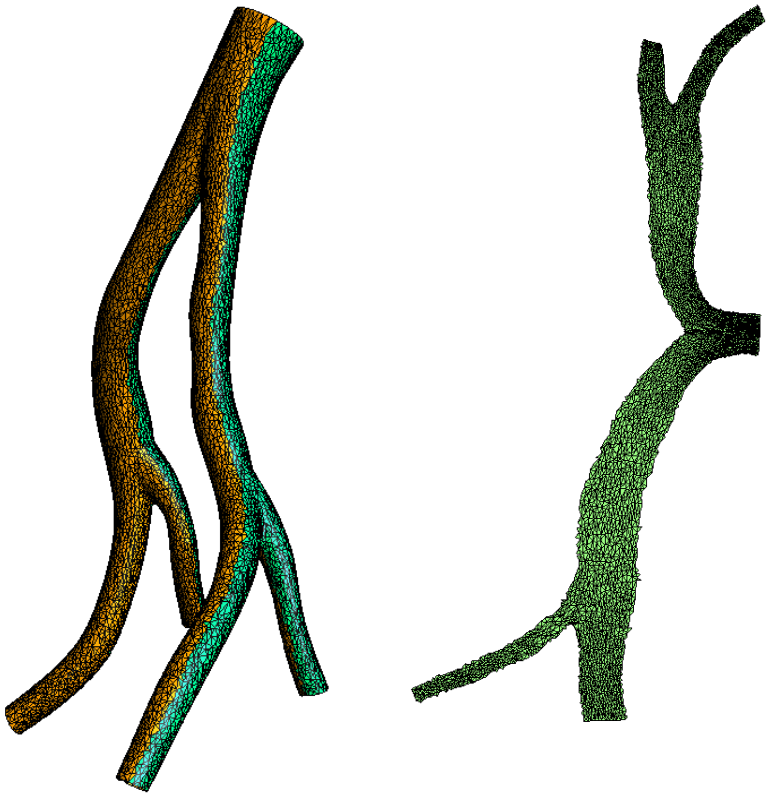
$$E_{\mathrm{LSCM}}(\boldsymbol{\xi}) = \int_{\mathsf{S}} \frac{1}{2}\|\nabla_{\widehat{x}} u^{\perp} - \nabla_{\widehat{x}} v\|^2 ds, \tag{5.3.3}$$

where $^{\perp}$ denotes a counterclockwise 90° rotation in S. For a 3D surface with normal vector $\mathbf{n}$, the counterclockwise rotation of the gradient can be written as: $\nabla_{\widehat{x}} u^{\perp} = \mathbf{n} \times \nabla u$.

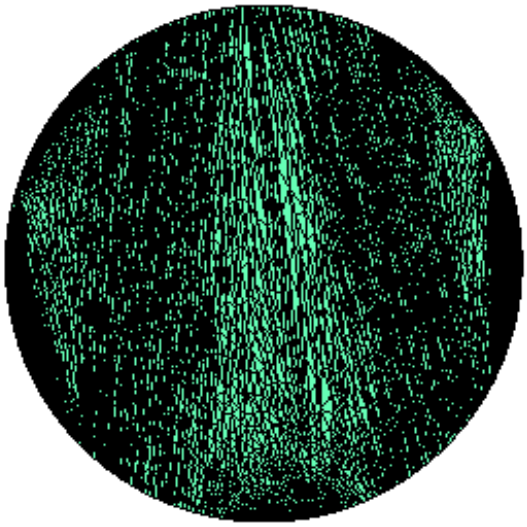
The form of (5.3.3) is further manipulated and the quadratic minimization problem is solved numerically with finite elements, see [MCG+11] for the details.

5.3.2 High Quality Meshes

Figure 5.7 shows two different steps in the parametrization-based remeshing algorithm of the initial triangulation of an iliac artery bifurcation. First the initial mesh is cut into different patches using the multiscale Laplacian partitioning method described in [MCdWV+10] (Figure 5.7a). Next, each mesh partition (orange and green) is parameterized onto a surface in $\mathbb{R}^2$ with a specific mapping algorithm (Figure 5.7bc). We show two different mappings: the Laplacian harmonic map onto a unit disk (Figure 5.7b) and the conformal map with open boundaries (Figure 5.7c).



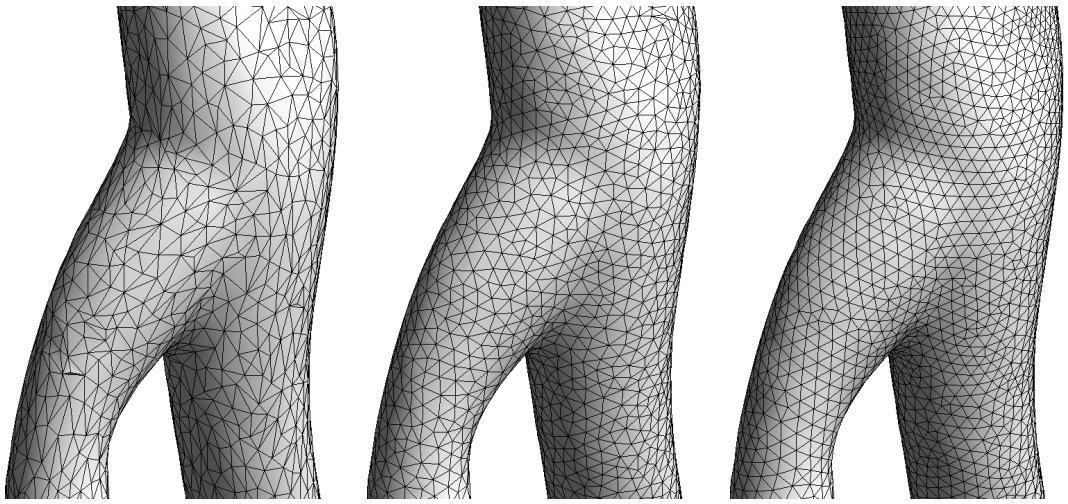
(a) (b)



(c)

Figure 5.7: Remeshing of an iliac bifurcation. The initial mesh is first split into two parts using the multiscale Laplacian partitioning method (a). Each part is then mapped in the parametric space by computing a Laplacian harmonic map onto a unit disk (c) and the conformal map with open boundaries (b).

After the mapping has been computed, the parameterized surface is remeshed using a 2D mesh generation algorithm and the new triangulation is then mapped back to the original surface. Figure 5.8 shows part of the remeshed iliac bifurcation for both the harmonic mapping and the conformal map.



(a) (b) (c)

Figure 5.8: Remeshing of an iliac bifurcation: a) part of the initial STL triangulation, b) remeshed geometry with the harmonic mapping using the MeshAdapt meshing algorithm, c) remeshed geometry with the conformal mapping using a frontal 2D meshing algorithm.

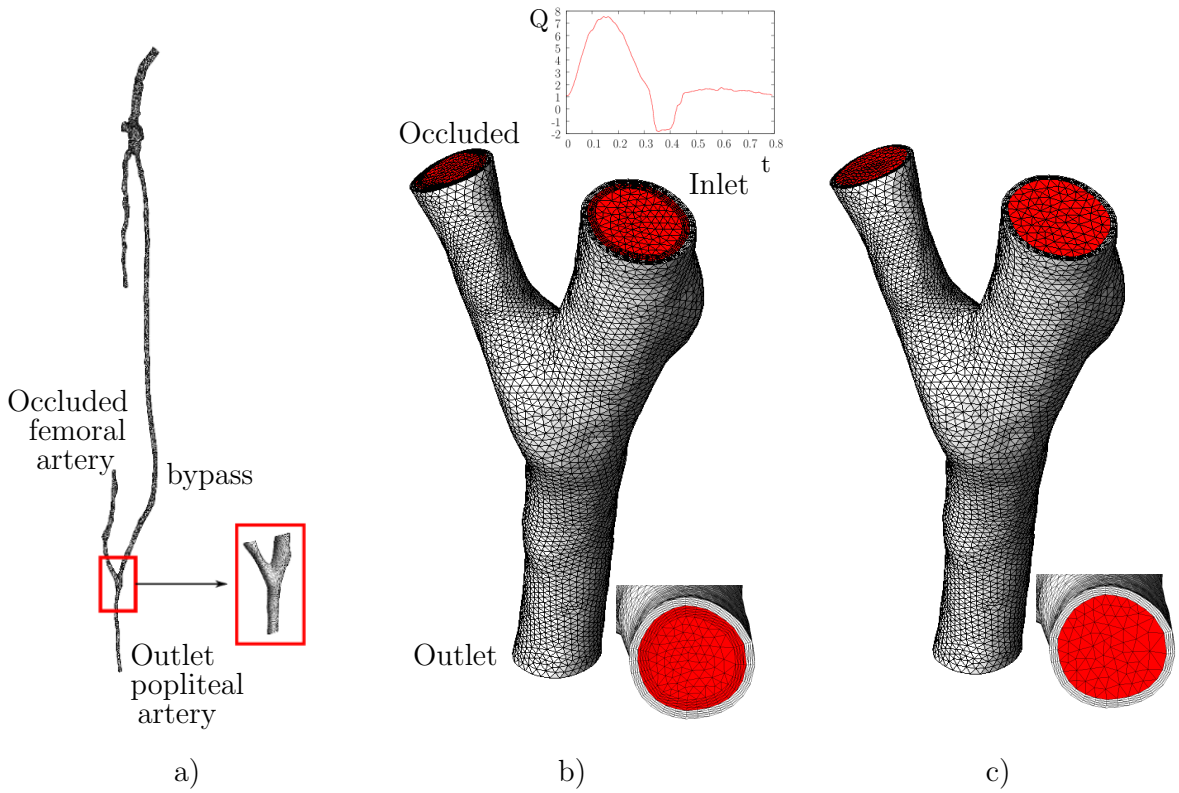


Figure 5.9: Segmentation of the arteries of the left lower limb of a patient with a venous graft that bypasses the occluded femoral artery. We have focused on the geometry of the distal anastomosis. Two different type of meshes are considered for the lumen volume (in red): b) meshes with a viscous boundary layer (BL meshes) and c) fully unstructured tetrahedral meshes (U meshes). The mesh of the vascular wall is colored in white. The presented meshes are of intermediate mesh size (see Table 5.2).

5.3.3 Unsteady Blood Flow in a Compliant Femoropopliteal Bypass

We compare in this subsection the FSI simulation on different meshes of blood flow in a pathological situation. The geometrical model of the distal anastomosis of a femoropopliteal bypass is obtained through a 3T MRI scanner of the left lower limb of a patient. The lumen geometry is subsequently reconstructed in 3D from the raw medical images using the open source software 3D Slicer[21] (see Figure 5.9). The vascular wall is obtained by extruding the lumen surface in the outward normal direction with a wall thickness corresponding to a tenth of the vessel radius. This extruded volume is then composed of 4 layers (the white volume in Figures 5.9b and 5.9c).

For the meshing of the lumen volume (see red volume in Figure 5.9b), we have considered different tetrahedral meshes: three fully unstructured meshes and three meshes with a viscous boundary layer. The size of these meshes is resumed in Table 5.2.

[21]The bypass geometry was provided by the group of Prof. E. Marchandise, iMCC-MEMA (Université catholique de Louvain (UCL)).

U Mesh	Fine	Intermediate	Coarse
# Nodes	154'732	36'490	8'508
BL Mesh			
# Nodes	212'633	57'318	23'697

Table 5.2: Different fluid meshes considered for the numerical simulation: the boundary layer meshes (BL mesh) such as in Figure 5.9b and the fully unstructured meshes (U mesh) such as in Figure 5.9c with three different mesh sizes h_{center}: fine, intermediate and coarse. For the BL meshes, the ratio h_{center}/h_{wall} is taken to be 10.

The boundary conditions imposed at the inlet-outlet of the vessel are patient-specific measured fluxes (Figure 5.9b), while an homogeneous Dirichlet condition is imposed on the occluded branch. The fluxes are imposed through a Lagrange multiplier as defective boundary conditions. The Young modulus and Poisson coefficient characterizing the elastic material modeling the arterial wall are respectively $E = 4 \cdot 10^6 \, \mathrm{dyn/cm^2}$ and $\nu = 0.45$. The fluid dynamic viscosity is $\mu = 0.035 \, \mathrm{g/(cm\,s^2)}$. The densities for blood and arterial wall are respectively $\rho_f = 1 \, \mathrm{g/cm^3}$ and $\rho_s = 1.2 \, \mathrm{g/cm^3}$. Timings and validation for the FSI solver used are discussed in Section §5.4, and are reported also in [CRD+11], while in Chapter 6 and in [CDFQ11] the scalability issue is addressed. The simulations reported in this section were run on the Cray XT6 supercomputer in the HECToR.

The FSI simulations presented in this section are run in parallel. As an example, the intermediate meshes are run on 48 cores, using 24 MPI processes per node and the simulation takes about 8 hours to perform one heartbeat. Due to the computational cost and since we are interested in the comparison of the WSS for the different meshes we did not run the simulation for several heartbeats, which would be necessary to reach periodicity and to obtain physiological results. We just ran for one heartbeat starting from a zero initial condition and we compared systolic hemodynamic values.

Figure 5.10 shows the results obtained with the boundary layer mesh of intermediate size. The streamlines clearly show the secondary flows which are in agreement with the WSS values. The flow impinging on the bed of the junction creates a region of high wall shear stress. Moreover, the blood flow is accelerated in the outlet popliteal artery since the graft is sewed on an artery of smaller diameter. This mismatch in diameter creates also a region of high wall shear stress near the outlet. The observed flow behavior in such an end-to-side bypass do not occur naturally in arteries and is widely implicated in the initiation of the disease formation processes [NSP99, GZG93].

Figure 5.11 shows the WSS distribution at peak systole obtained for the six different meshes. WSS shows to be higher in the simulations with boundary layer. Although we do not ensure that the WSS convergence is reached with the finest mesh, we already see a significant difference in the order of magnitude of the WSS. Considering as reference solution the one obtained using the finest mesh with boundary layer we conclude that the WSS is quantitatively better evaluated on the meshes with boundary layer, while even the finest mesh without boundary layer shows a substantial underestimation of the WSS with respect to all the meshes with boundary layer.

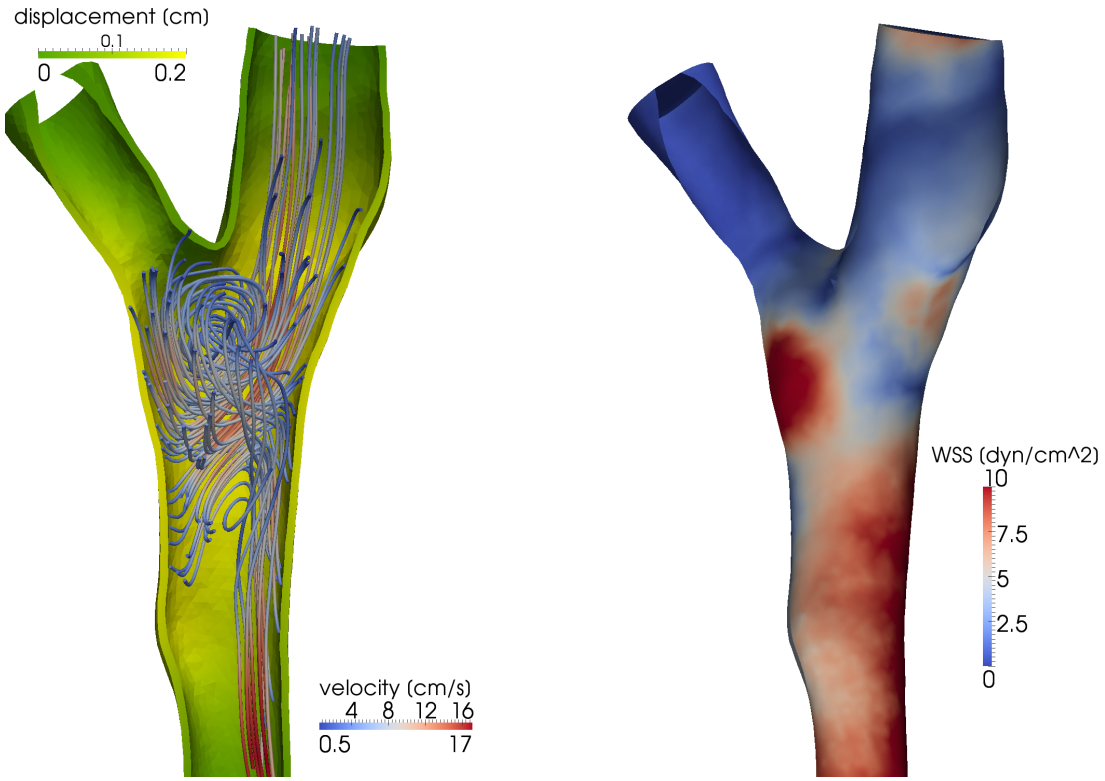


Figure 5.10: Bypass simulation. Streamlines, wall displacement magnitude and distribution of WSS at end systole ($t = 0.3\,\mathrm{s}$) obtained with the boundary layer mesh of intermediate size.

5.4 Blood Flow in the Aorta

The interaction between the blood flow and the arterial wall deformation has to be taken into account to correctly predict the behavior of the arterial flow, especially where the pressure impulse induces large deformations of the domain, as in the aortic arch. In this section we show the results of a simulation of the hemodynamics in the aorta, taking into account the interaction between the blood flow and the arterial wall, modeling blood as a Newtonian fluid and the aortic wall as a linear elastic structure.

The FSI simulation of blood flow in the aortic arch was performed in [GWM06] on a simplified geometry without branching and using a three-layer nonlinear model for the structure. In [GWM06] the FSI system is formulated in an ALE frame and solved with a standard Dirichlet–Neumann method imposing an inlet flux and homogeneous Neumann condition on the outlet. An FSI simulation in a patient-specific aorta using the ALE formulation can be found in Bazilevs et al. [BGH$^+$09], where a pathological case was considered. The physiological boundary conditions used were taken from a previous reduced model simulation, and particular attention is devoted to the fluid flow pattern, wall shear stress and oscillatory shear index. In our work, besides reporting the quantities that influence the genesis of several pathologies (e.g. the wall shear stress) we highlight some characteristics that are peculiar to the compliant wall simulations and validate the FSI model, such as the pulse wave velocity

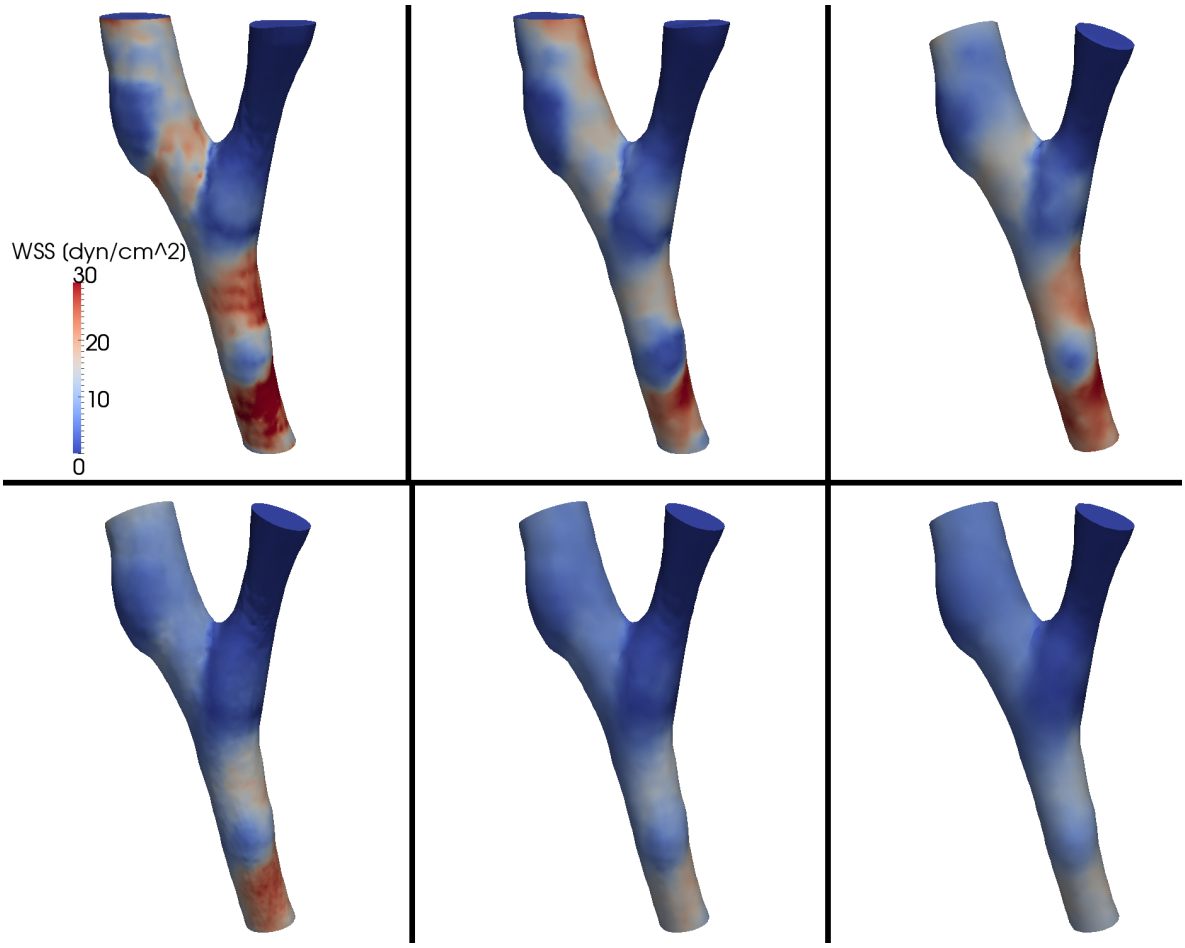


Figure 5.11: Comparison of the WSS pattern for six meshes at peak systole ($t = 0.2s$). On the first row are the meshes with boundary layer, referring to Table 5.2, from left to right: fine, intermediate, coarse. On the second row are the meshes without boundary layer. From left to right: fine, intermediate, coarse.

(PWV) and the radius change. In the next section we report the settings of our case study, selection and motivation of boundary conditions and a clear motivation for taking into account the compliance of the walls in aorta simulations. As a matter of fact, we show that some physiological patterns are captured only by the FSI simulations while they are not reproduced by a rigid walls simulation. We also summarize a quantitative analysis of the results from a bio-engineering point of view, referring to [RCD$^+$11] for the details.

Patient-specific proximal aorta and main arterial branches geometry is acquired using angiography MRI. The lumen geometry is obtained after segmentation and the arterial wall geometry is created with a variable thickness. The boundary conditions for the fluid domain at the ascending aorta root are either pressure or fluxes. Fluxes are imposed for each outlet. The waveforms are obtained using phase contrast MRI for blood velocity or, when not available at a specific location, completed by results from a 1D model. This 1D model was validated quantitatively on the same patient. To mimic the mechanical effects of surrounding tissues, a linear stress-displacement constitutive relation is applied on the outer surface of the arterial

wall. This is imposed as a Robin condition, as explained in the next subsection.

5.4.1 Boundary Conditions

When dealing with sectors of vascular districts, the boundary conditions for the numerical simulation play an important role. For instance they allow to take into account the global circulation and the presence of surrounding tissue. The former mainly affects the fluid flow, the latter the structure displacement.

A crucial issue concerning the imposition of boundary conditions on the arterial wall is the choice of the conditions for the terminations of the arterial branches. On the fluid outlets fluxes obtained by a 1D tree simulation are imposed by Lagrange multipliers as described in [Ver10, FGNQ02a]. At the inlet, i.e., at the aortic valve location, we impose either the flux measured on the patient or a pressure obtained from the 1D simulation. Here we do not account for the coupling between the 3D simulation and the 1D one.

Another possibility currently under development is to model the interaction between the segment of aorta considered and the rest of the circulation through the geometrical multiscale framework introduced in Section §2.9. In this way the morphology of the arterial tree and the peripheral circulation can be taken into account. Another simpler option which models the continuation of the artery segments at their boundary, and account for the resistance to the flow due to an hydrostatic pressure, consists in absorbing boundary conditions (a particular type of 0D models), also recalled in Section §2.9. However such conditions on the fluid outlets inhibit the imposition of other physiological quantities such as velocity or stress and emulates the continuation of the vessel with an infinite long cylinder.

In literature many strategies to impose absorbing boundary conditions on the fluid outlets have been devised. These allow in FSI to avoid the non physiological pressure wave reflections [Nob01, FQV09, KVCF$^+$09], cf. Section §2.9.

The 1D tree model and the simulation used to impose these boundary conditions are detailed in [RBP$^+$10]. Each piece of the tree is a one-dimensional reduction of the FSI problem (cf. Section §2.9). The main systemic arteries as well as a detailed description of the cerebral circulation are represented. The constitutive law for the arterial wall is non-linear and viscoelastic.

This has been validated qualitatively and quantitatively on the same person [RBP$^+$10]. Pressure was measured with applanation tonometry[22] and cerebral blood flow velocities with transcranial ultrasound and phase contrast MRI. Pressure and fluxes were computed at each segment of the arterial tree. The mean fluxes at each inlet and outlets are summarized in Table 5.3, and the history of the fluxes is shown in Figure 5.12.

Concerning the structure, at the best of our knowledge absorbing boundary conditions for FSI have not been implemented yet, thus we chose in our simulations to clamp the vessel wall at the aortic valve and to impose a Neumann homogeneous condition on the structure outflow terminations. With this choice we did not observe spurious reflection waves. A possible (more physiological) option would be to adopt the same strategy as for the fluid, that is to impose the displacement obtained from the previous 1D simulation (cf. Remark 2.9.1).

The simulation of a vessel subject to a high load, as it is the case in the thoracic aorta, undergoes large deformations concerning both the luminal radius and the vessel displacement.

[22] A non-invasive technique for assessing cardiovascular function. Applanation tonometry is frequently used to measure pulse wave velocity and to perform pulse wave analysis *pulse wave analysis*[TY01]

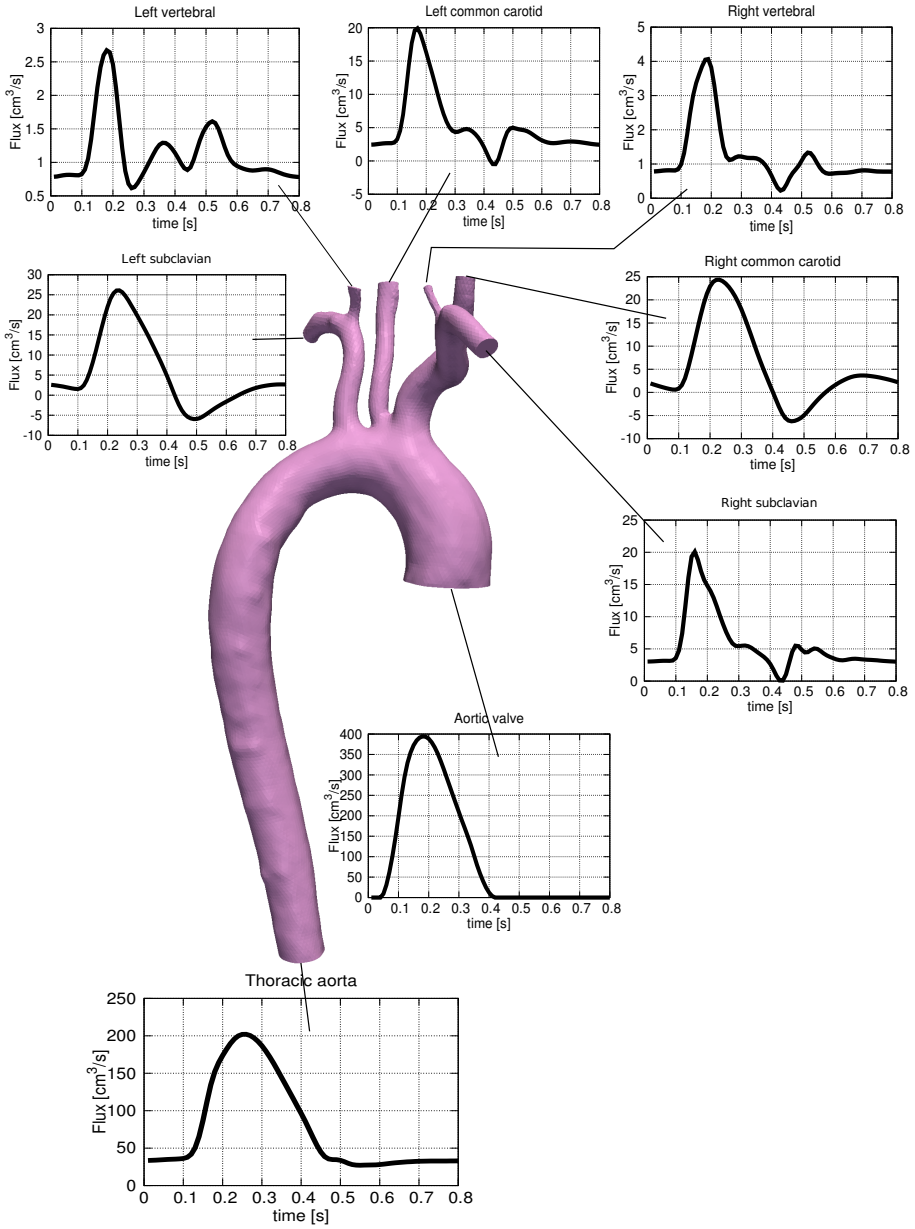


Figure 5.12: 3D representation of the lumen of the aortic arch and its principal branches. It consists of 1 inlet (ascending aorta root) and 7 outlets (vertebral, subclavian and carotid arteries, and thoracic aorta). The fluxes imposed as boundary conditions are represented for each inlet-outlet and are the result of a 1D tree simulation.

In particular, in a curved vessel, when imposing a free stress condition on the external wall, the movement of the domain turns out to be overestimated, as the surrounding tissue around the solid wall is not taken into account in the model.

The importance of considering the surrounding tissue effects is often neglected, although

	Mean fluxes (ml/s)
Ascending aorta	100.78
Thoracic aorta	76.60
RCCA	5.58
RVA	1.23
R subclavian	5.47
LCCA	5.18
LVA	1.14
L subclavian	5.57

Table 5.3: Mean flow rate over a heart cycle at inlet and outlets of the 3D domain.

in some cases it is shown to significantly change the hemodynamics (see e.g. [LDG$^+$08]). Furthermore the limit of the external layer (adventitia) of the arterial wall is sometimes arbitrarily chosen, since the transition from adventitia to the surrounding tissue is often gradual [OOL04]. In absence of a constitutive law for the heterogeneous tissues surrounding the aorta, and to obtain a simple model for the external response, we assume a linear algebraic stress displacement constitutive relation on the external wall. This choice is arbitrary, but leads to good agreement with literature [NOHM98, Fun97], when we properly tune the coefficients in the constitutive law.

The influence of the surrounding tissue on the arterial hemodynamics was recently investigated in-vivo and in-vitro on swines [LDG$^+$08]. In this work experiments with and without taking into account the surrounding tissue were performed. When the latter is taken into account, under the physiological range for pressure and displacement, the intraluminal pressure was shown to be proportional to the radius of the lumen. This suggests indeed a pressure-displacement linear constitutive relation. More precisely, this is modeled in our case as a Robin condition on the external wall:

$$p_o \widehat{\mathbf{n}}^s + \mathbf{\Pi} \cdot \widehat{\mathbf{n}}^s + \alpha \widehat{\mathbf{d}}_s = 0 \text{ on } \Gamma_s^{ext}, \tag{5.4.1}$$

where $\mathbf{\Pi}$ is the structure Piola stress tensor, p_o is a static pressure and $\widehat{\mathbf{n}}^s$ is the outward normal to Γ_s^{ext}. This model was adopted in [RCD$^+$11, CRD$^+$11] and a similar approach is proposed also in [MXA$^+$11].

Tuning the parameter α in (5.4.1) is rather difficult. We empirically found that a value of $\alpha \approx 10^4 \frac{dyn}{cm^3}$ leads to physiological displacements.

In the present simulations we chose a parameter α varying in space, in particular increasing when the arterial wall becomes thin. This roughly reproduces the stiffening effects predicted by the Moens-Korteweg relation (cf. Section §5.4.2, Equation (5.4.3)), which are not otherwise considered in the current simulations. In fact, as the wall thickness varies according to the intraluminal radius (as described in Section §5.4.2), using a constant coefficient α we observed a non physiological dilatation in the small branches when subject to a high intraluminal pressure. A more accurate possibility to overcome this problem consists in adopting a nonlinear structure model where the effects of collagen are taken into account, and/or in considering a variable Young modulus, related to the wall thickness through the Moens-Korteweg relation reported in Section §5.4.2.

5.4.2 Geometrical Model

The segmentation and the reconstruction of the arterial wall geometry for the present simulation was carried out by Dr. P. Reymond and collaborators (LHTC, EPFL) and it is described with more details in [RCD+11]. To obtain a 3D representation of the arterial lumen (interface between arterial wall and blood), MRI Time of Flight acquisition on a 3T MRI scanner (Siemens Trio-Tim 3T System) were performed. Details on the sequences utilized are mentioned in [RBP+10].

The arterial tree geometry was reconstructed in 3D from the raw DICOM medical images (ITK Snap). After pre-processing of the DICOM images, using gradient diffusion, a growing region based on contrast threshold was obtained. This served as a starting surface for an improved segmentation using an edge detection method based on intensity gradients.

First a geometrical surface of the lumen was created from the extracted surface (that was previously segmented) using ICEM CFD 11[23]. The lumen extracted surface of the aorta was obtained by MRI, however the thickness of the arterial wall was not retrieved with this technique. To build the geometry of the solid region in a physiological manner it has been based on human ex-vivo measurements carried out by Langewouters et al. [Lan82]. Based on ex-vivo samples, they performed thickness (h) and outer diameter (D_e) measurements on human thoracic aortas and reported thickness to arterial diameter ratio as a function of age and location:

$$\frac{h}{D_e} = a + b \cdot \text{age} = \text{Const.}$$

The thickness varies spatially and is proportional to the local lumen diameter. The reported value of this ratio is $c = 0.054$ for the thoracic aorta and $c = 0.07$ for the abdominal aorta for a 30 years old male. We use the mean value of these two aortic locations for the whole computational domain (i.e., we assume that ratio constant). Therefore the thickness to lumen diameter that we assumed for the whole aorta in the model is

$$\frac{h}{D_{lumen}} = \frac{c}{1-2c} \approx 0.071 \tag{5.4.2}$$

The lumen diameter (D_{lumen}) was deduced from the local distance between vessel centerline and arterial lumen mesh elements. The thickness was set locally in the normal direction, it varies spatially and decreases when moving downstream.

The centerline of the arterial tree was computed using the *Aneufuse* software developed by the EU Project @neurIST. An unstructured tetrahedral mesh, conforming at the FS interface, was then generated for the fluid region using Gambit[24].

Although the procedure for extracting the vessel wall thickness is not arbitrary, a certain variability is expected, due e.g. to the difference between the *ex-vivo* and the *in-vivo* measures. Furthermore the following components that describe the geometry and the stiffness of the arterial wall are function of the location:

- the elastic modulus of the arterial wall increases when moving distally (constituents of the arterial wall, elastin/collagen repartition is different)
- the arterial diameter to thickness ratio increases as well when moving distally.

[23] ANSYS Inc., Canonsburg, PA
[24] ANSYS Inc., Canonsburg, PA

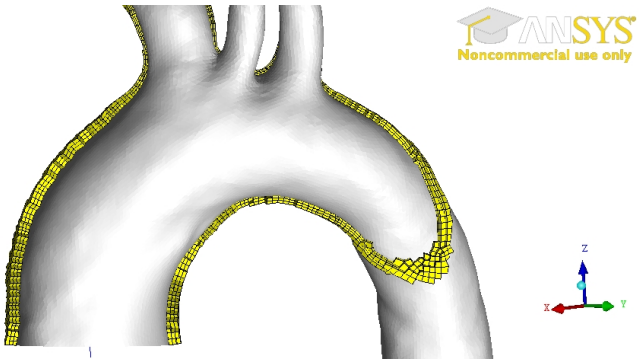


Figure 5.13: Arterial wall cut representation. The mesh has a variable thickness, proportional to the local lumen diameter of the vessel. In this example, the mesh contains 3 layers of prismatic elements, which are later replaced by tetrahedral elements in order to be compatible with the LifeV solver.

The contribution of these two components increases the pulse wave velocity, a clinical parameter of importance, which has been shown, in-vivo, to increase as well when moving distally. In fact according to the adapted Moens-Korteweg equation, the relationship between the elastic modulus ϵ and the pulse wave velocity PWV [NO05] is given by:

$$\mathrm{PWV} = \sqrt{\frac{\epsilon h}{\nu D(1-\nu^2)}}, \tag{5.4.3}$$

where ν is the Poisson coefficient of the arterial wall ($\nu = 0.45$). We obtained a PWV that is approximatively $5\,\mathrm{m/s}$. It corresponds to physiological data for a young person aorta.

The arterial wall constitutive relation in our case is linear elastic and isotropic. This is quite a restrictive assumption and the generalization to nonlinear elasticity is subject of ongoing work. However for a simulation of non-pathological range of intra arterial pressures it leads to results in the physiological range of values; as we will show (cf. Section §5.5), these results by far improve those obtained using the rigid walls assumption. In this study we assume also that the thickness to arterial lumen diameter ratio h/D and the elastic modulus ($\epsilon = 0.4\,\mathrm{MPa}$) for the whole geometry of the arterial wall are constant. In a similar study, by Kim et al. [KVCF$^+$09], a constant value of $\epsilon = 0.6\,\mathrm{MPa}$ was used. Due to the observations above the assumption of constant Young modulus becomes less physiologically accurate when considering an extended region of the cardiovascular system.

The increase in the coefficient α in (5.4.1) contributes to stiffen the arterial wall when moving distally, which mimics the effects described above.

5.4.3 Timings and Validation for FSI (GCE)

The FSI simulations are run for three heartbeats, Figure 5.14 reports the pressure and displacement for fixed time steps. There are no significant differences in the time history of pressures and fluxes between the second and the third heartbeat (Figures 5.16 and 5.17), which suggests that the solution reached a substantial periodicity.

These simulations are performed using the GCE time discretization, cf. Section §2.6. Other time discretizations are explored in Section §5.5, which are more reliable but more computa-

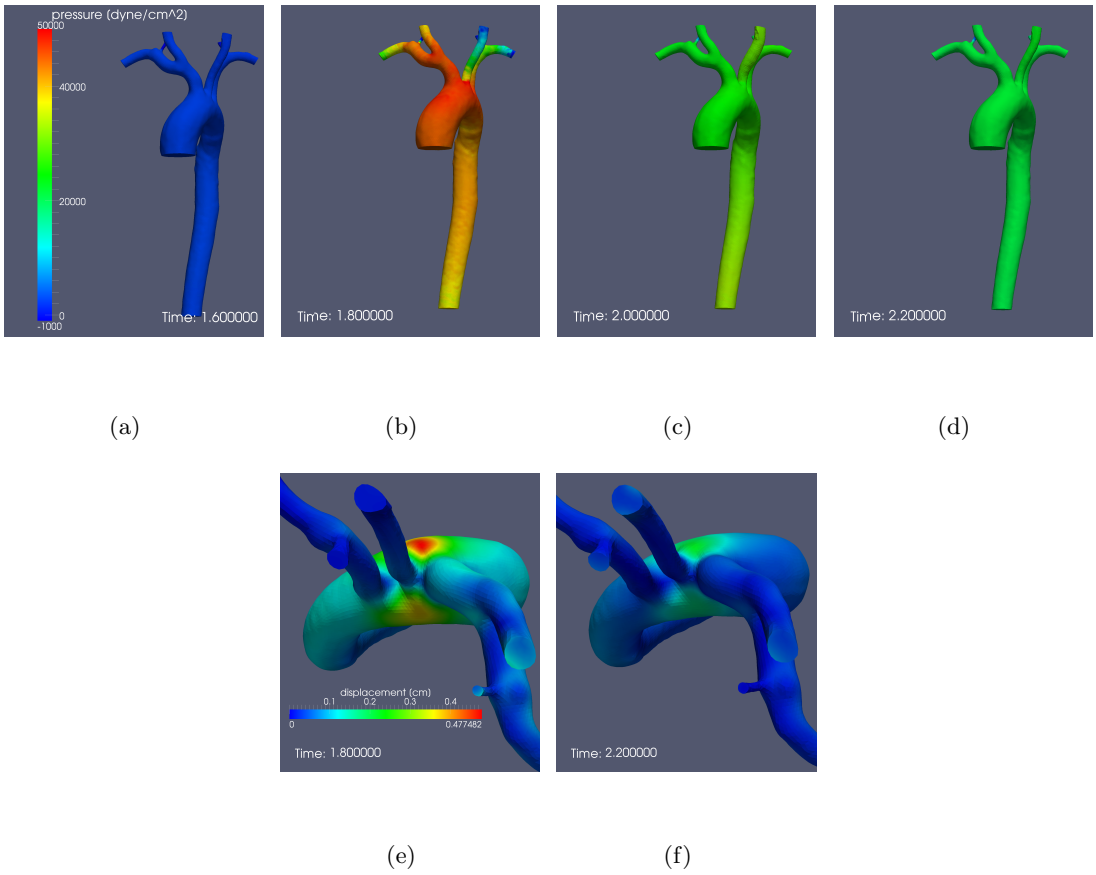


Figure 5.14: Pressure and displacement distribution at different times. Figure (a) represents the pressure at the end of the second heartbeat (at $t = 1.6\,\mathrm{s}$). The pictures (b), (c), (d) are taken at intervals of 0.2 s. We remark that the pressure reaches the maximum value in proximity of the systole, at $t \approx 1.8\,\mathrm{s}$. The (e) and (f) figures represent the displacement magnitude at $t = 1.8\,\mathrm{s}$ and $t = 2.2\,\mathrm{s}$. The location of the maximum displacement during systole in the aortic arch is probably due to the curved and branching shape of the geometry inducing a variation of the eccentricity of the lumen.

tionally expensive. In that section also other comparisons (with a rigid walls simulation and with a 1D model simulation) are carried out, while in the rest of this section we focus on the description of the results of the simulations obtained, in terms of computational efficiency and of clinical interest of the output.

The results of this subsection are obtained using the Cray XT4 supercomputer in HECToR. These simulations are run on 16 nodes for a total of 64 MPI processes, and the mesh considered is the one represented in Figure 5.18. For the domain partitioning we used the ParMETIS[25] library (Figure 5.18 shows the partition of the artery in 32 subdomains). The partition of the fluid and solid meshes is performed independently, and each processor holds both a fluid and a solid partition.

At every time-step most of the time is spent in the preconditioner computation, in the

[25] http://glaros.dtc.umn.edu/gkhome/metis/parmetis/overview

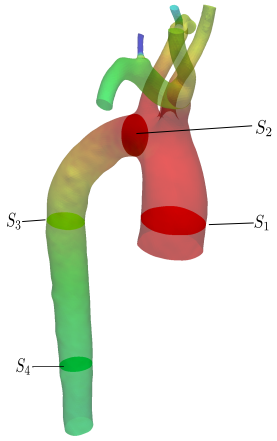


Figure 5.15: Sections in which the mean values of pressure and velocity are computed.

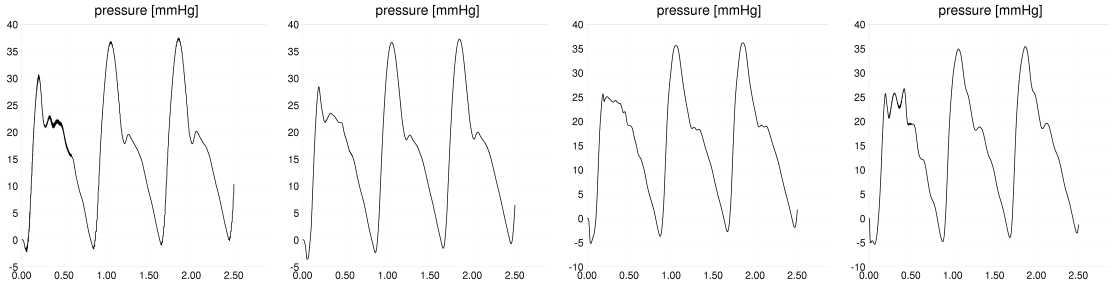


Figure 5.16: FSI simulation with inlet flux imposed. History of the mean pressures (in mmHg) at S_1, S_2, S_3 and S_4 in Figure 5.15 (starting from the left with the S_1 section, until the S_4 section on the right).

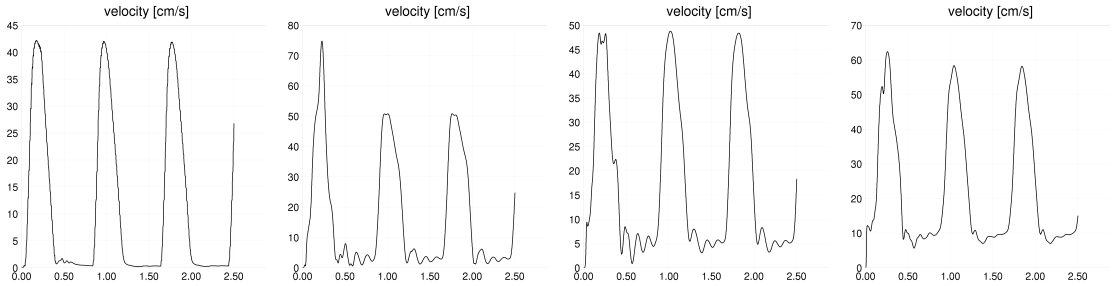


Figure 5.17: FSI simulation with inlet flux imposed. Mean velocities computed in the sections S_1, S_2, S_3 and S_4 represented in Figure 5.15.

solution of the linear system, and in the assembly of the matrix block corresponding to the fluid equations. The average time for these tasks is 16.6 s, 11.5 s and 8.1 s respectively (the assembly of the stabilization part at every time iteration takes 7.1 s, while the assembly of the rest of the block takes about 1 s).

The preconditioner considered is P_{AS-DN1} (cf. Section §4.3.2), where a one-level algebraic additive Schwarz (AAS) preconditioner with two layers overlap (see [QV99]) is employed for each factor. The LU factorization of the sub-blocks in the AAS preconditioner is achieved through the unsymmetric multifrontal method implemented in the package UMFPACK[26]. The linear system is solved with preconditioned GMRES iterations and the average number of iterations is about 25 to reach a tolerance of 10^{-7}.

The time-step chosen is $\delta t = 10^{-3}$ s, so that one heartbeat (0.8 s) consisted of 800 time levels. Summing the timings for the solution of the linear system, the preconditioner computation and the assembly of the fluid block we obtain a global timing per heartbeat of about 8 hours. We remark that the assembly of the fluid block scales when increasing the number of processors since it requires few inter-processor communications (the *ghost nodes* of each partition are repeated on all the processors), while the preconditioner computation, and in particular the GMRES solution, are less scalable operations. We refer to [CDFQ11] and to Chapter 6 for considerations on the scalability of the FSI system and for a discussion on suitable preconditioners for coupled problems.

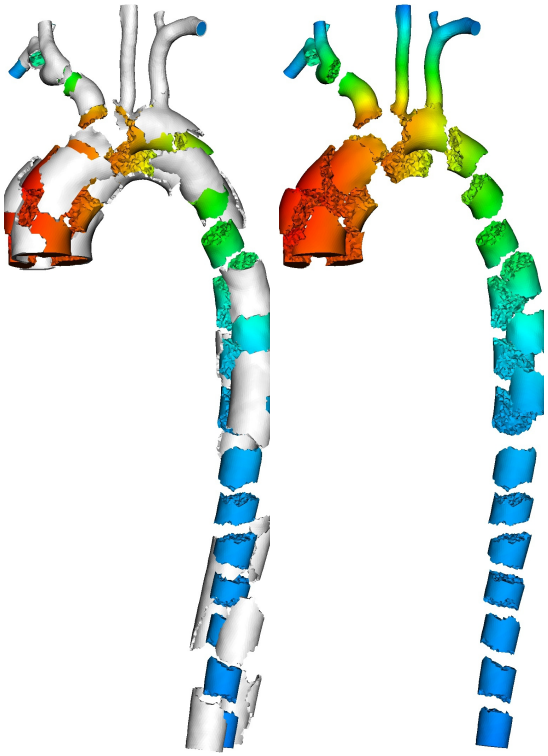


Figure 5.18: Aorta mesh partitioned with 32 processors using ParMETIS. The mesh is composed of 380′690 tetrahedra, i.e., 486′749 dofs.. On the right: the fluid mesh partitioned; on the left: both fluid and solid mesh partitioned.

In order to compare the time spent for one iteration of rigid walls with that spent for an iteration with FSI GCE we run another cycle of 200 time steps for both rigid walls and FSI

[26] http://www.cise.ufl.edu/research/sparse/umfpack

problems with pressure imposed at inlet. In these simulations we use the block preconditioner P_{AS-DN1} for FSI discussed in Section §4.3, see also [CDFQ11], which allows a separate treatment for the fluid and the solid blocks, while in the rigid walls case we use the same preconditioning strategy used for the fluid block in FSI.

These simulations are carried out on the Cray XT6 supercomputer in HECToR. The tolerance for solving the linear system is set to 10^{-7} in both cases. We use 4 MPI processes per node; the rigid walls problem is parallelized on 64 processors, while in the FSI case we use 128 processors. In both cases the overall time spent per time-step is of about 30 seconds.

The velocity of the pressure wave is measured by evaluating the foot of pressure in two sections of the descending aorta, at 5 cm distance from each other. The phase shift observed is 0.009 s, which corresponds to a velocity of 5.5 m/s. This value corresponds to the physiological PWV (around 5 m/s, see [NOHM98, Fun97]). However this is closely related to the Young modulus chosen in the solid model, which depends on the specific case and is influenced by many factors. For patient specific simulations these parameters should be accurately tuned.

Furthermore, frequently the Young modulus for in-vivo arteries is devised from the PWV using empirical formulas, or from measurements of pressure versus luminal area, thus already taking into account the effect of the surrounding tissue [LDG+08].

To test the influence of the flux boundary condition imposed at the inlet and also to have another comparison with the rigid walls simulation, we run for several heartbeats an FSI simulation imposing the pressure at the inlet obtained from the 1D model simulation. With this choice we do not guarantee that the inlet flux is zero when the aortic valve is closed. We observe indeed a reflux in the diastolic phase, which can be interpreted as the back flow inducing the closure of the aortic valve. This phenomenon is physiologically observed and cannot be simulated without taking into account the compliance of the wall. However imposing a stress condition at the inlet introduces another parameter to be tuned. In fact the outer static pressure p_o, that we set to zero in (5.4.1), in this case is no longer arbitrary. The value chosen is $p_o = 115'000\,\mathrm{dyn} \approx 86\,\mathrm{mmHg}$, which is slightly larger than the diastolic inlet pressure.

The time histories of the mean pressure and normal velocity computed on the sections represented in Figure 5.15 are reported in Figures 5.19 and 5.20, where only the second heartbeat is represented. Note that in the figures the zero value corresponds to $p_o \approx 86\,\mathrm{mmHg}$.

When imposing inlet pressure in the compliant case less physiological pressure waveforms are obtained at distal (for instance section S4) thoracic regions. This is due to a not appropriate pressure wave reflection at the outlet boundary. The backflow amplitude is sensitive to the pressure waveform and to the modeling of the aortic valve in the 1D. Considering that, it seems more preferable to impose the flux at the inlet. In addition, blood flow rate is measurable in-vivo non invasively.

Figures 5.21 and 5.22 show the time histories of the radius at the different sections of the aorta $S_2 - S_4$ (see Figure 5.15). The radius change (around 1 or 2 mm) matches the experimental results reported in [vPVS+09]. Figure 5.23 represents a section of the lumen boundary in the middle of the aortic arch at different times, showing in particular the change in eccentricity of the lumen. Patient-specific informations about the radius and eccentricity change in the aorta can be very useful for endovascular aortic repairs. Endografts should be designed in order to dynamically adapt to the changes in aortic shape. Furthermore *non uniform deformation and strain may very well contribute to the localization and development*

of vascular pathology [vPVS+09].

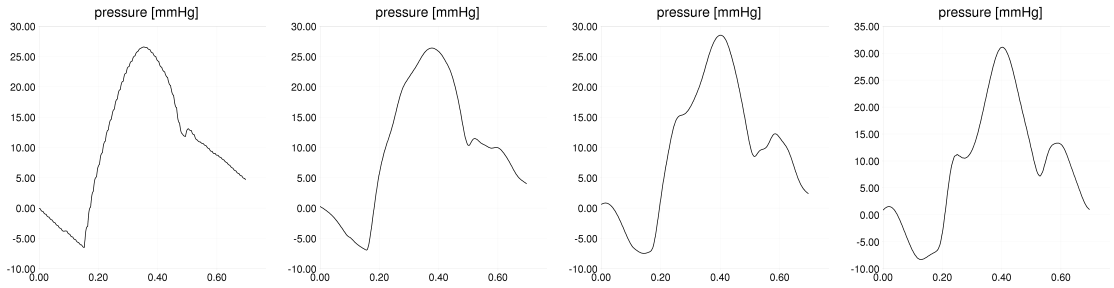


Figure 5.19: FSI simulation with inlet pressure imposed. History of the mean pressures (in mmHg) over the sections S_1, S_2, S_3 and S_4 represented in Figure 5.15. The zero value of the pressure correspond to $p_o \approx 86\,\text{mmHg}$.

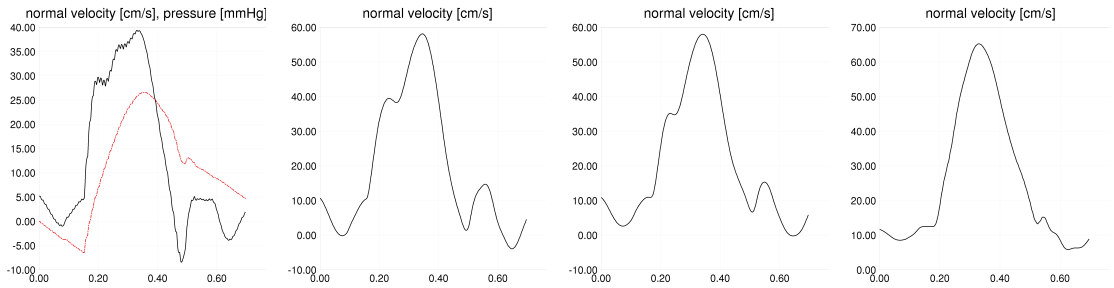


Figure 5.20: FSI simulation with inlet pressure imposed. History of the mean normal velocities (in $^{\text{cm}}/_{\text{s}}$) over the sections S_1, S_2, S_3 and S_4 represented in Figure 5.15. In the plot for the S_1 section (left) we show the pressure curve over the same section with a dashed line, to highlight the dependence of the reverse flow on the steep pressure decrease after the systole. Notice that at regime in the abdominal aorta no more backward flow is observed.

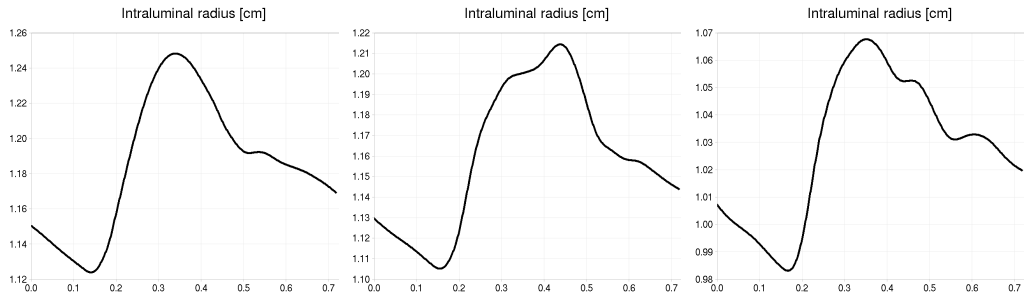


Figure 5.21: FSI simulation with inlet flux imposed. History of the mean radius over the sections S_2, S_3 and S_4 represented in Figure 5.15 (third heartbeat).

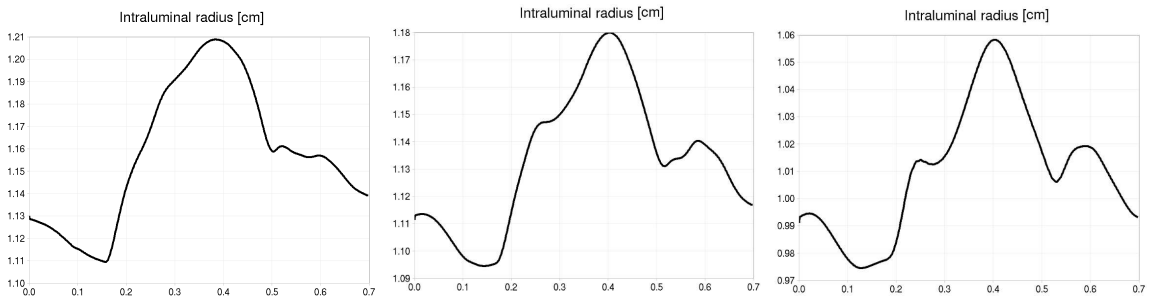


Figure 5.22: FSI simulation with inlet pressure imposed. History of the mean radius over the sections S_2, S_3 and S_4 represented in Figure 5.15 (third heartbeat).

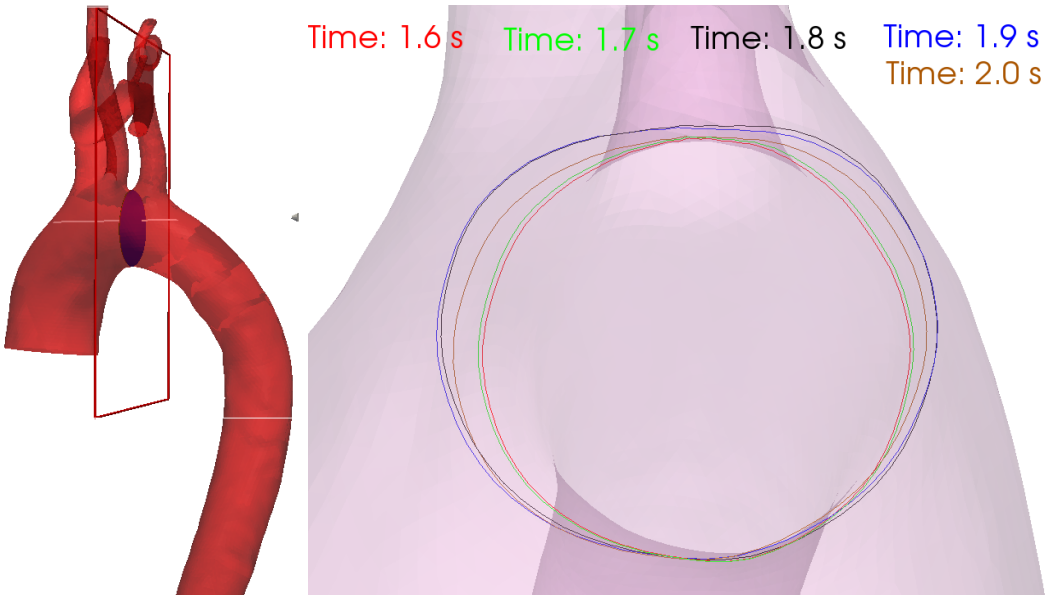


Figure 5.23: Several snapshots of a section of the lumen located in the middle of the aortic arch (shown on the left) during the third heartbeat of the inlet flux FSI simulation.

5.4.4 Numerical Assessment

In order to check the validity of our results we performed the same FSI simulation with inlet pressure imposed on a finer mesh for about 200 time levels. In particular the new mesh considered has $1'118'517$ elements for the fluid, $483'469$ for the structure, the overall number of degrees of freedom for the FSI problem is $2'044'263$ (see Figure 5.24). The simulations carried out to compare the results on the two meshes were run on two different clusters. for the finer problem we used the Cray XT5 supercomputer Rosa. The job was parallelized on 512 cores (using 6 cores per computing node). The overall time spent per time-step was about 150 seconds. For details on preconditioner and scalability we refer to Chapters 4 and 6 or to [CDFQ11]. The coarser problem was re-run on the Cray XT6 supercomputer in HECToR. The simulation was run in parallel on 128 cores, while the CPU time per time-step was about 30 seconds.

In Figure 5.25 we compare the WSS distribution for both meshes. The results obtained with both the simulations do not show substantial differences. The regions of low and high WSS can thus be qualitatively identified solving the coarser problem as well as the WSS magnitude. Since the computation for the large mesh is demanding larger computational resources we could not run it over several heartbeats.

5.4.5 Wall Shear Stress

The arterial wall tissue reacts to both the normal and shear stresses [FQV09, Ch. 1]. In particular, the wall shear stress is involved with the formation of atherosclerosis, which is a pathology characterized by a narrowing of the arterial lumen due to the accumulation of fatty material. However wall shear stress is difficult to measure in vivo with a sufficient spatial resolution.

The shear stress is defined as the tangential component of the traction vector $\sigma_f \widehat{\mathbf{n}}^f$. Thus

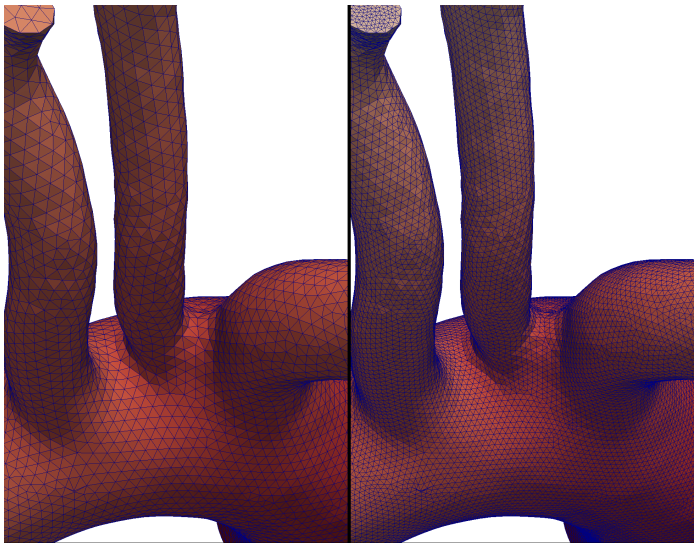


Figure 5.24: The coarse (left) and the fine (right) meshes.

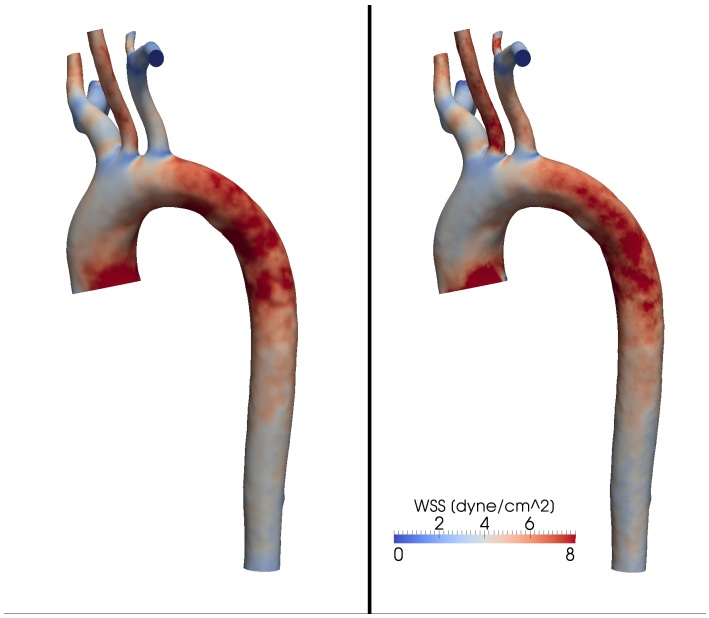


Figure 5.25: The WSS distribution for the coarse mesh (left) and the fine one (right) at time $t = 0.12\ s$.

in Newtonian fluids it is

$$\boldsymbol{\tau} = \sigma_f \mathbf{n}^f - (\sigma_f \mathbf{n}^f \cdot \mathbf{n}^f)\mathbf{n}^f = \mu_f(\nabla_x \mathbf{u} + \nabla_x \mathbf{u}^T)\mathbf{n}^f - \mu_f\{[(\nabla_x \mathbf{u} + \nabla_x \mathbf{u}^T) \cdot \mathbf{n}^f] \cdot \mathbf{n}^f\}\mathbf{n}^f \tag{5.4.4}$$

where μ_f is the dynamic viscosity. This equation corresponds to

$$\boldsymbol{\tau} = \mu_f[(\nabla_x \mathbf{u} + \nabla_x \mathbf{u}^T) - (\nabla_x \mathbf{u} + \nabla_x \mathbf{u}^T) : (\mathbf{n}^f \otimes \mathbf{n}^f)]\mathbf{n}^f.$$

If $\mathbf{u} = (u_1, u_2, u_3)^T$, writing the last equation in Einstein notations we obtain

$$\tau_j = \mu_f[(\partial_i u_j + \partial_j u_i) - (\partial_k u_l + \partial_l u_k)(n_k n_l)\delta_{ij}]n_i.$$

Fixing a point on the fluid–structure interface we can write all the quantities with respect to the associated local frame of reference $(\mathbf{t}_1, \mathbf{t}_2, \mathbf{n})$, where $\mathbf{t}_1$ and $\mathbf{t}_2$ are orthonormal vectors on the tangent plane and $\mathbf{n}$ is the normal vector. We denote the local coordinate system associated with this frame (h_1, h_2, h_3). The previous expression reads

$$\boldsymbol{\tau} = \mu_f \begin{pmatrix} (\partial_{h_2} u_{h_3} + \partial_{h_3} u_{h_2}) \\ (\partial_{h_1} u_{h_3} + \partial_{h_3} u_{h_1}) \\ 0 \end{pmatrix},$$

where now $\mathbf{u}_h = (u_{h_1}, u_{h_2}, u_{h_3})^T$ denotes the velocity field written with respect to the local coodinate system. We notice here that if the fluid wall is fixed every tangential derivative is zero, $\partial_{h_1} u_{h_3} = 0$, $\partial_{h_2} u_{h_3} = 0$. This leads to a straightforward relation between the magnitude of the wall shear stress and of the vorticity vector $\boldsymbol{\omega} = \nabla \times \mathbf{u}$. In fact we have that

$$\boldsymbol{\omega} = \begin{pmatrix} \partial_{h_2} u_{h_3} - \partial_{h_3} u_{h_2} \\ \partial_{h_3} u_{h_1} - \partial_{h_1} u_{h_3} \\ \partial_{h_1} u_{h_2} - \partial_{h_2} u_{h_1} \end{pmatrix} = \begin{pmatrix} -\partial_{h_3} u_{h_2} \\ \partial_{h_3} u_{h_1} \\ 0 \end{pmatrix}$$

and thus $\mu_f \|\boldsymbol{\omega}\|_2 = \|\boldsymbol{\tau}\|_2$.

When the wall is moving this relation is no longer valid. We compute thus in the compliant case the WSS from relation (5.4.4). However most of the time we do not observe a significant difference when computing the WSS magnitude through the vorticity vector even in the compliant walls case. This is perhaps due to the fact that the velocity gradient due to the boundary layer (the normal derivatives of the tangential velocity) dominates the other components of the velocity gradient. We refer to [WMZ06] for details and discussions about shear stress and vorticity relations.

As an example Figure 5.26 shows the wall shear stress (WSS) magnitude in the compliant and rigid walls simulations right after the systole of the second heartbeat. The period of one heartbeat is $T = 0.8\,\mathrm{s}$, so that $t = 1\,\mathrm{s}$ corresponds approximately to the systolic peak. The WSS distribution is similar at systole with compliant and rigid walls, although it is slightly larger in the latter case. Then at $t = 1.1\,\mathrm{s}$ and $t = 1.2\,\mathrm{s}$ the WSS magnitude is larger in the rigid walls case. In particular the WSS in the rigid walls simulation seems to be overestimated. The systolic WSS are in the physiological range, especially in the compliant case, if compared with measurements obtained in [BLS10] for the control patients. We do not observe remarkable differences between the two FSI simulations corresponding to different inlet boundary conditions.

The comparison between the rigid-wall simulation and FSI is only qualitatively addressed in Figure 5.26, where we show that accounting for the compliance of the vessel wall leads to clear differences in the WSS distribution. For a quantification of these differences we refer to Table 5.4 and to [RCD+11].

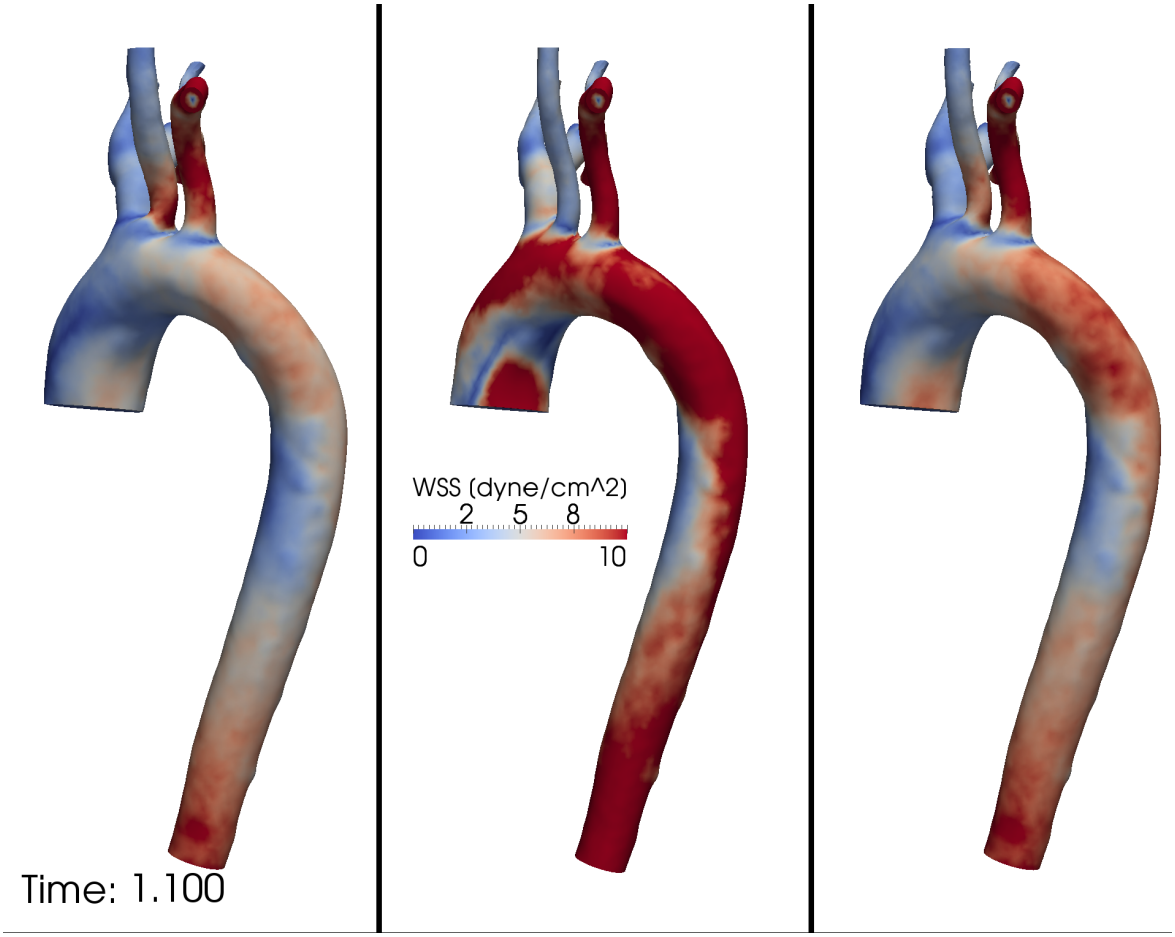


Figure 5.26: The wall shear stress distribution at $t = 1.1$ s for the compliant walls simulation with the inlet flux imposed (left), inlet pressure imposed (right) and rigid walls (center).

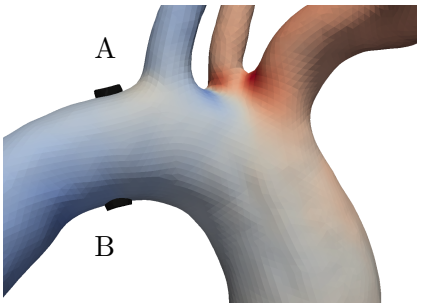


Figure 5.27: In black are highlighted the regions A and B where the averages of the WSS are computed.

5.5 Comparisons

5.5.1 GCE versus FI

We analyze the influence of some choices for the time discretization method, in particular how the implicit coupling of the geometry problem and the time discretization of the convective term influence the evaluation of quantities of physiological interest like WSS, pressure, displacement and velocity fields. We would like to quantify the error committed with the geometry–convective explicit (GCE) time discretization and other *geometry–implicit* variants with respect to the Fully Implicit (FI) version of the algorithm. We look for an optimal compromise between accuracy and computational cost of the simulation.

To this aim we prepare an initial condition by running a GCE simulation of an heartbeat on the same aorta geometry with a large time step ($\delta t = 8 \cdot 10^{-3}$), and we start the second heartbeat varying the method and the time step. We report the simulation until the end of the systole ($t \approx 0.3\,\text{s}$, while the systolic peak is observed at $t \approx 0.2\,\text{s}$). In the simulations performed we consider four schemes: besides the fully implicit and geometry–convective explicit schemes we call Convective Explicit (CE) the case in which both the fluid velocity and domain velocity are extrapolated using (2.4.4) and (2.4.5), and Geometry Implicit (GI) the case in which only the fluid velocity in the convective term is discretized using (2.4.4), while the fluid domain and its velocity are considered implicitly.

We report in Figure 5.28 the WSS obtained with the FI scheme, in both the regions A and B represented in Figure 5.27. The time steps considered are $\delta t = 0.0005\,\text{s}$, $\delta t = 0.001\,\text{s}$, $\delta t = 0.002\,\text{s}$, $\delta t = 0.004\,\text{s}$. This picture shows that for the FI scheme the position of the peaks remains almost unchanged when increasing the time step, but the pattern followed by the WSS histories changes substantially for the largest time step considered.

The same plot for the GCE time discretization is reported in Figure 5.29. In the latter case we see that the choice of the time step leads to a shift of the position of the peaks in the WSS. The FI solution will be our reference for the comparisons reported below.

Figure 5.30 represents the history of the WSS averaged in the region A, which is represented in Figure 5.27. The simulation for the smallest time step, i.e., $\delta t = 5 \cdot 10^{-4}\,\text{s}$ is performed only for GCE and FI time discretizations, since these are the extreme cases. We did not experience stability issues when increasing the time step, however from these graphs

it is clear that the largest time step fails in predicting the correct value for the averaged WSS, even in the FI case.

Furthermore these graphics show that, for a time step lower then $\delta t = 10^{-3}$, in the simulation of the aortic arch a GCE time discretization provides an approximation of the WSS which can be satisfactory, since it qualitatively follows the one obtained with the geometry–implicit variants.

Another interesting result is the difference between the fully implicit simulation and the GI and CE ones. This suggests that considering implicitly the part of the convective term nonlinear in $\mathbf{u}$ (2.4.2) brings non negligible changes. On the other hand from the similarity between CE and GI we conclude that treating implicitly the nonlinearity in $\mathbf{w}$ (2.4.3) in the convective term does not influence much the WSS in the location considered.

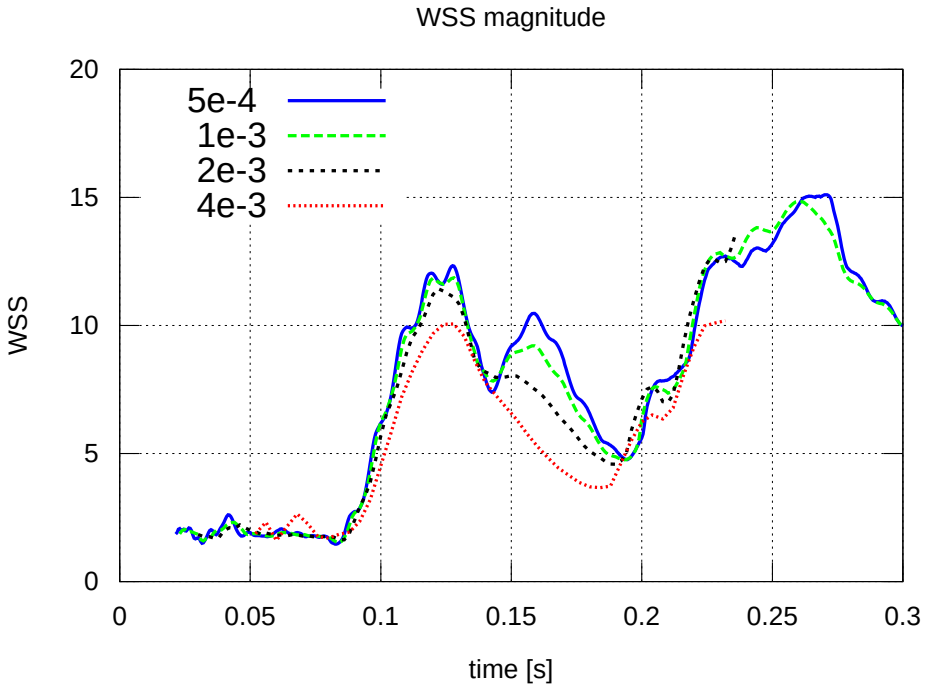
Since the number of Newton iterations in all the geometry implicit variants does not vary substantially we can conclude that among those the most performant and reliable method is the FI one. Concerning the GCE time discretization, it is considerably faster than all the other methods for a fixed time step and due to the dissipative behavior it shows to be stable also for large time steps, thus it can provide a fast, not always reliable, prediction of the hemodynamic, or a periodic initial condition for the other methods (as done for the current simulations).

In Figure 5.31 the same comparisons are performed on a different location (location B in Figure 5.27). The WSS is well approximated by the GCE especially before the systolic peak and for small time steps (below 1 ms), while it smooths most of the oscillations for larger time steps. As in the previous case the difference when considering the nonlinear term in $\mathbf{u}$ is important, while considering explicitly the domain velocity $\mathbf{w}$ in the convective term does not lead to substantial changes.

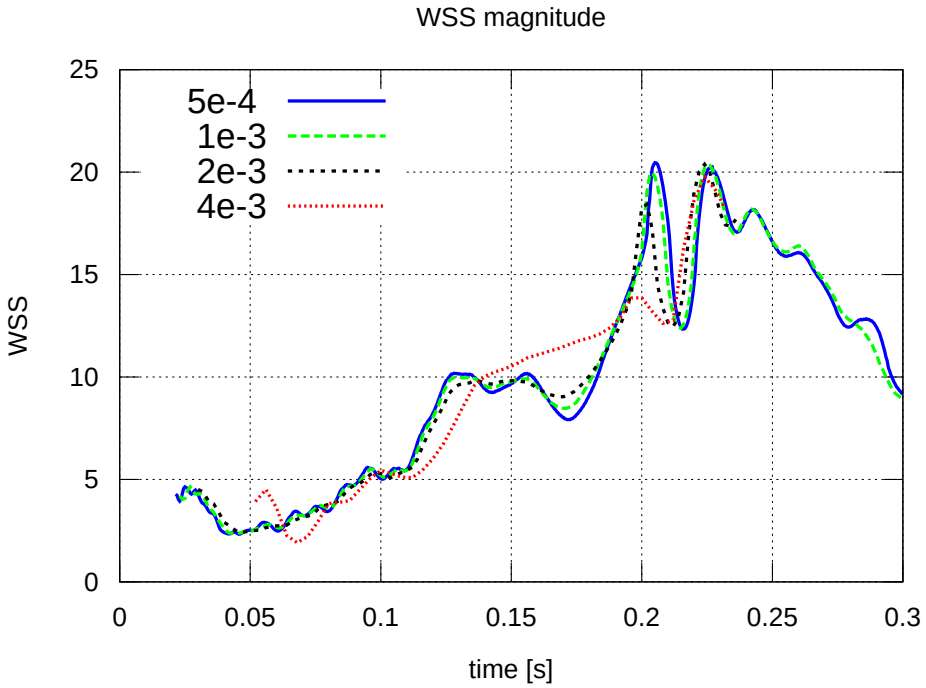
We report in Figure 5.34 and Figure 5.32 also the pressure and displacement in the region A obtained with the different time steps and discretization schemes. The same comparison for region B is shown in Figures 5.35 and 5.33. We notice from these pictures that the GCE time discretization, though being stable for all the time steps tested, fails in reproducing the time history of pressure and displacement of the FI for large time steps. The time step should thus be tuned according to the output of interest and to the precision requested.

Eventually we report in Figure 5.36 a comparison of the mean velocities and pressures computed with the various schemes for a fixed timestep of $\delta t = 10^{-3}$. The quantities represented are the averages on a section of the lumen in the descending aortic arch. We see from this plot that the difference between pressure and flux on a section of the lumen is small compared to the difference observed on the regions A and B of the arterial wall (Figure 5.27).

A picture of the reference solution (the FI simulation with time step $\delta t = 0.0005\,\mathrm{s}$) is represented in Figure 5.37 together with the WSS distribution at the end-systole. The decreasing of the inlet flux originates vortices in the ascending aorta, while the streamlines show the presence of secondary flows in the descending part of the aortic arch. This behavior is typical for blood flow in a curved vessel (see e.g. Doorly and Sherwin [FQV09, Ch. 5.3]). The simulations were run on the Cray XT4 cluster in HECToR, using 64 MPI processes (4 per node).

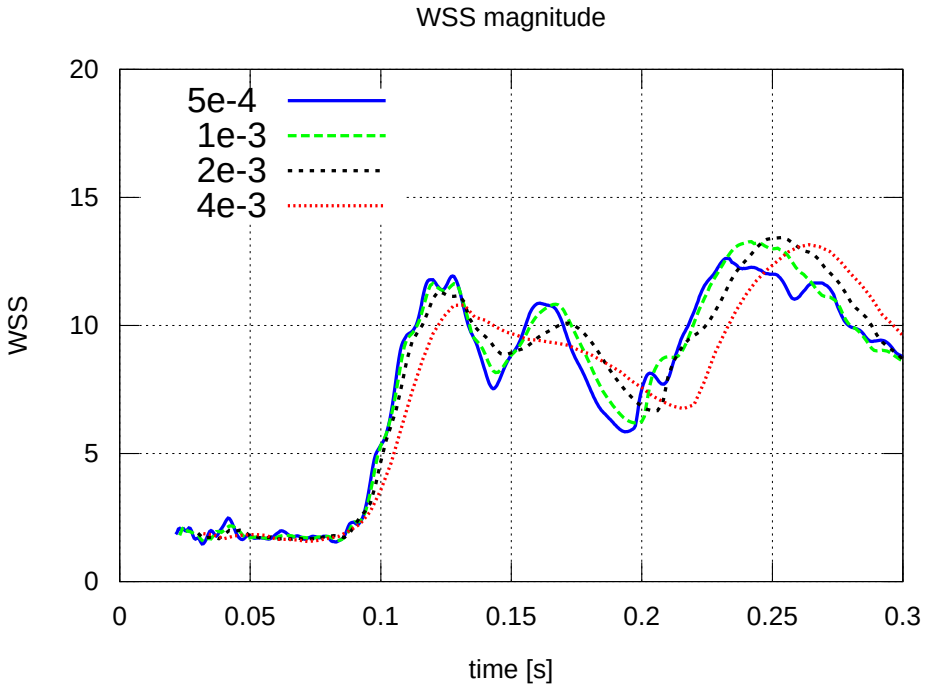


(a) location A

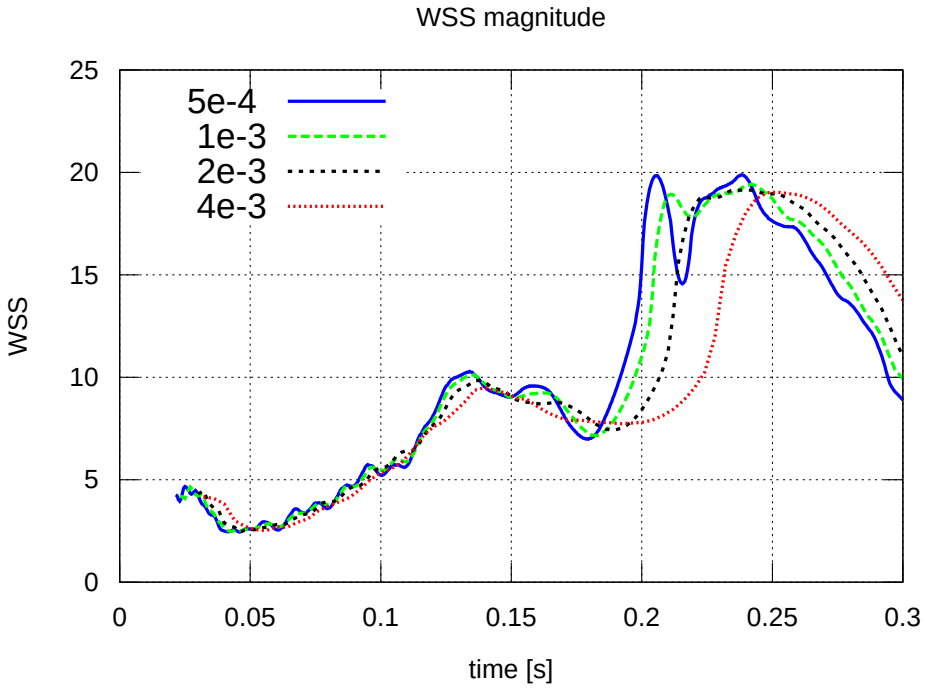


(b) location B

Figure 5.28: WSS ($^{dyn}/_{cm^2}$) for different time steps, FI discretization on the locations A (a) and B (b).

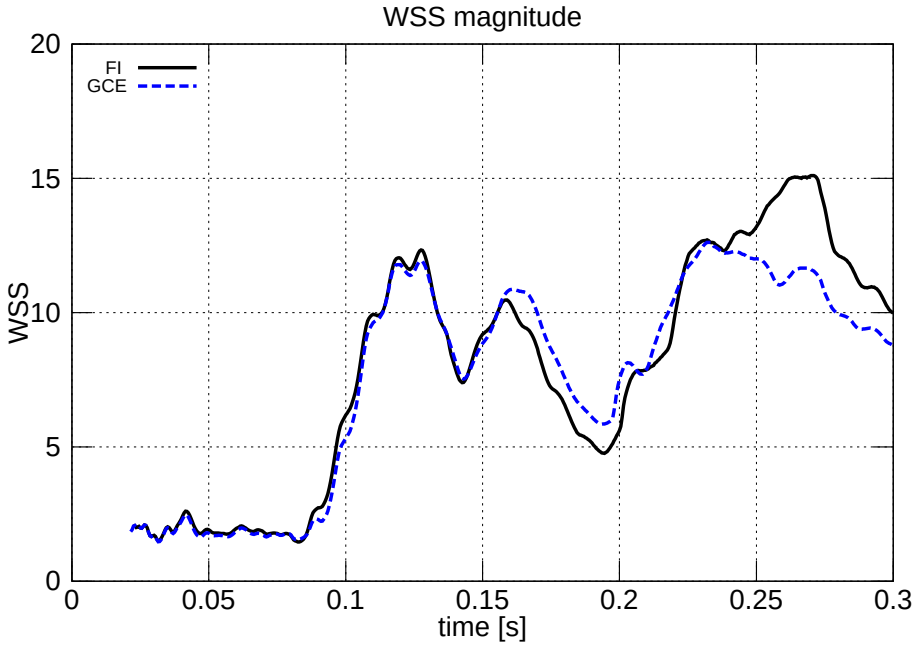


(a) location A

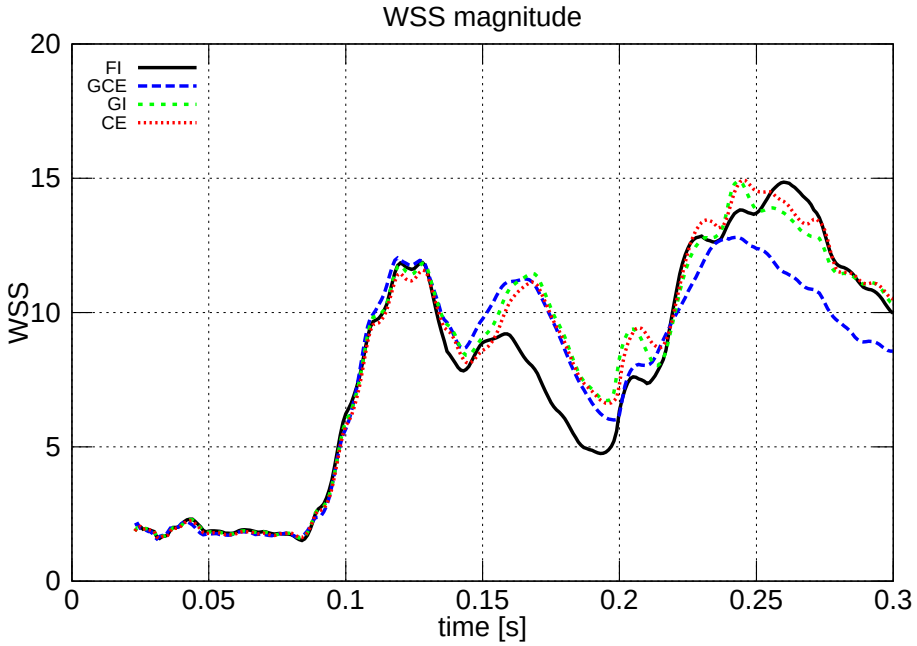


(b) location B

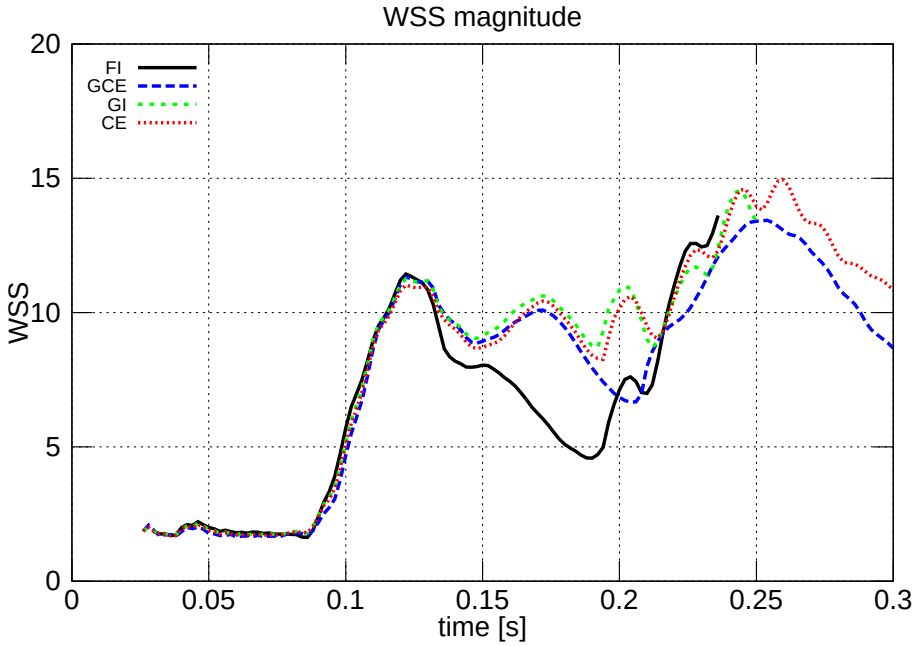
Figure 5.29: WSS ($^{dyn}/_{cm^2}$) for different time steps, GCE discretization on the locations A (a) and B (b).



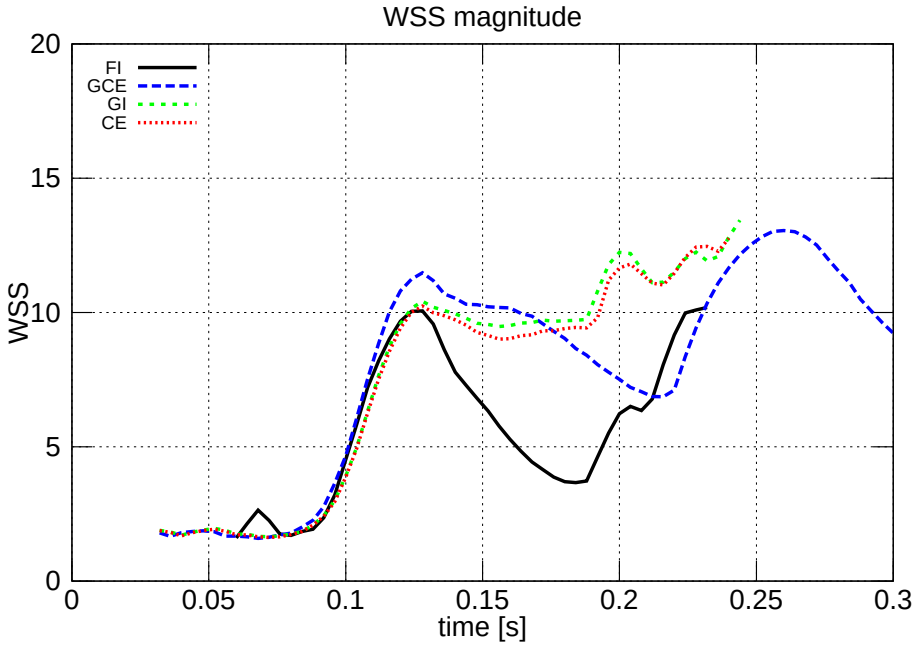
(a) $\delta t = 5 \cdot 10^{-4}$ s



(b) $\delta t = 10^{-3}$ s

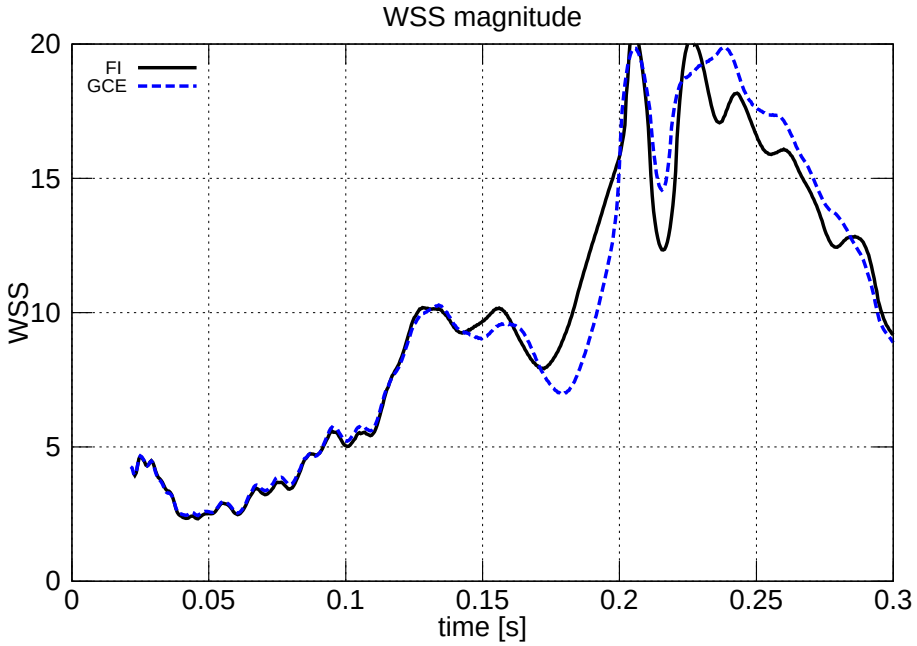


(c) $\delta t = 2 \cdot 10^{-3}$ s

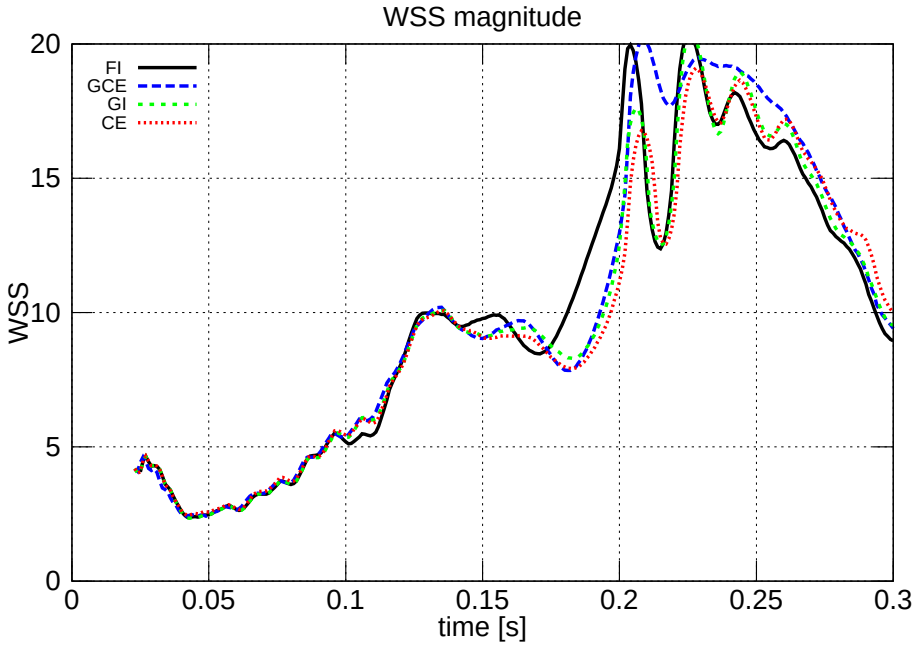


(d) $\delta t = 4 \cdot 10^{-3}$ s

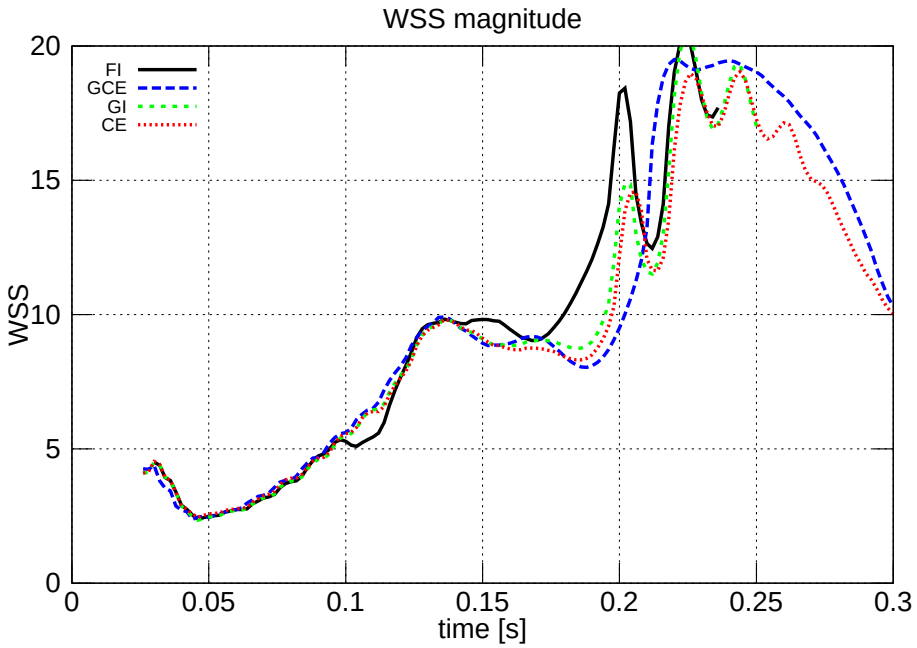
Figure 5.30: Time history of the WSS magnitude (in $^{\mathrm{dyn}}/_{\mathrm{cm}^2}$) averaged on the location A of Figure 5.27.



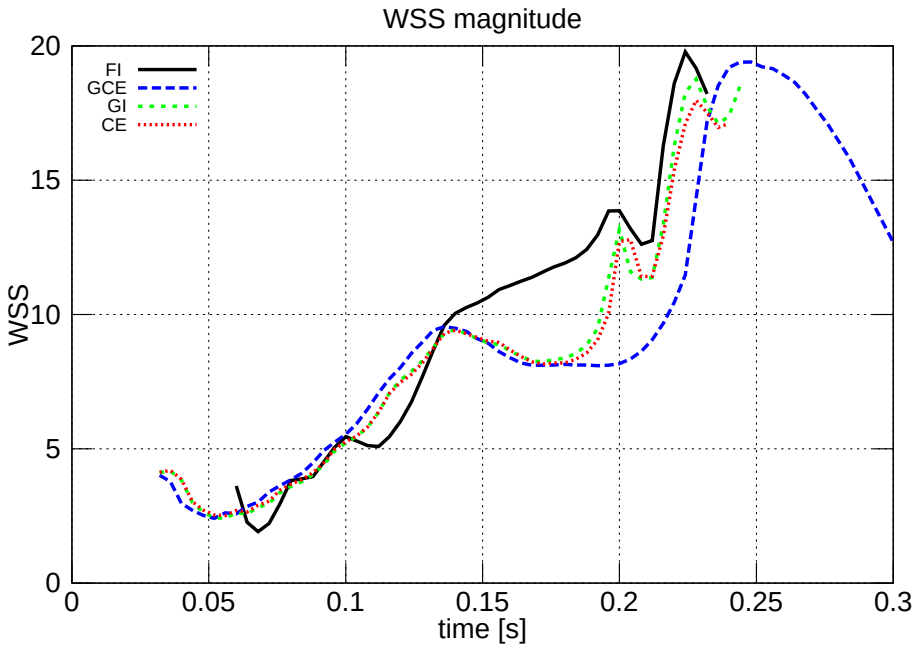
(a) $\delta t = 5 \cdot 10^{-4}$ s



(b) $\delta t = 10^{-3}$ s

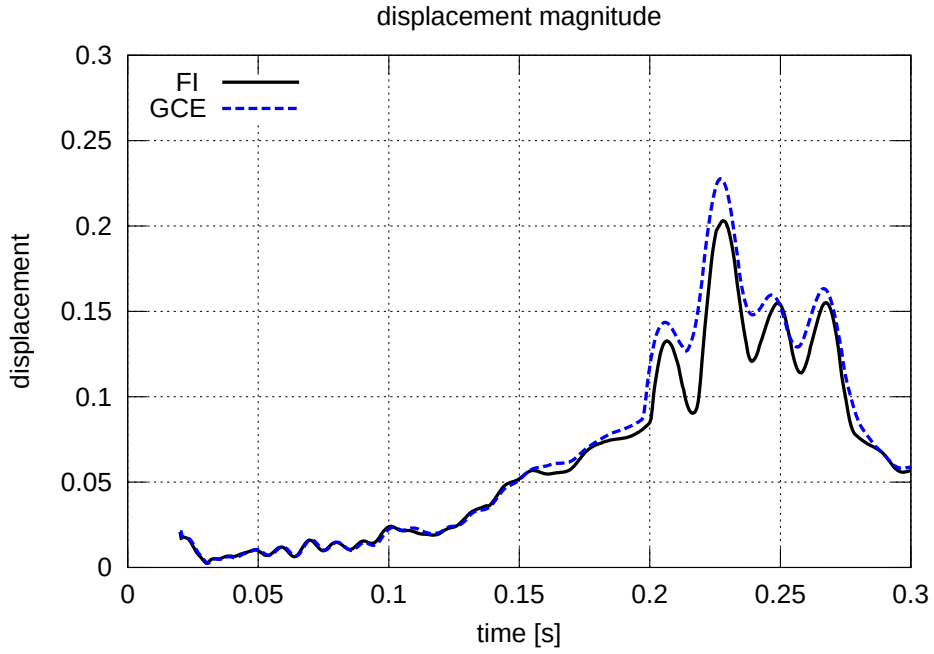


(c) $\delta t = 2 \cdot 10^{-3}$ s

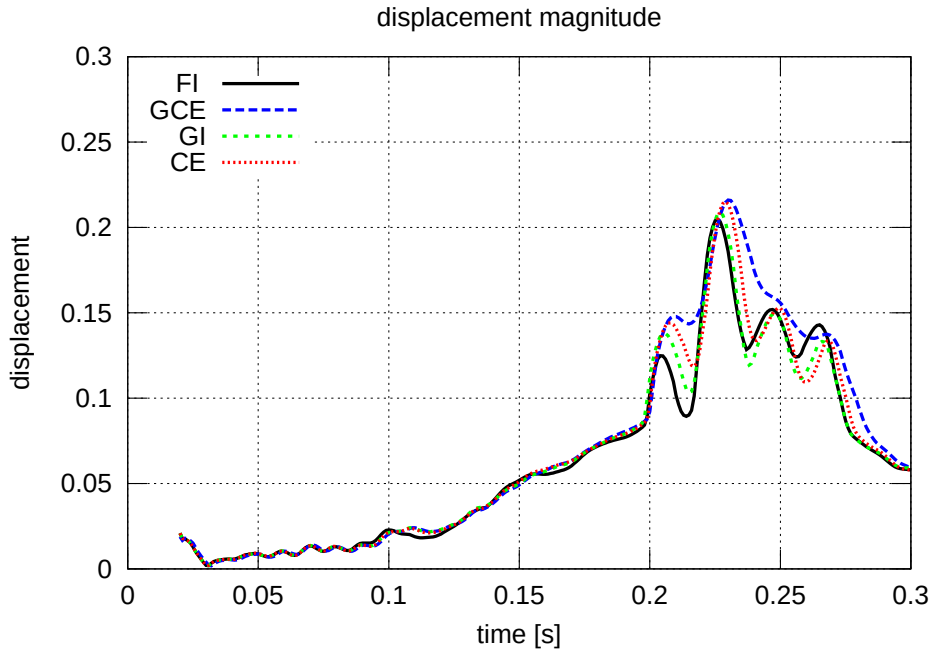


(d) $\delta t = 4 \cdot 10^{-3}$ s

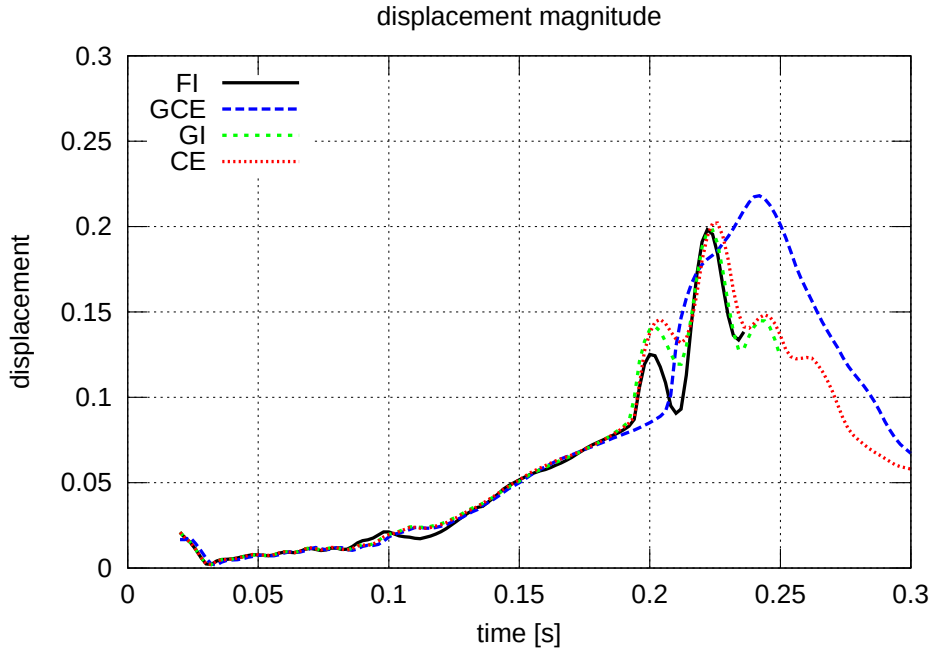
Figure 5.31: Time history of the WSS magnitude (in $^{\mathrm{dyn}}/_{\mathrm{cm}^2}$) averaged on the location B of Figure 5.27.



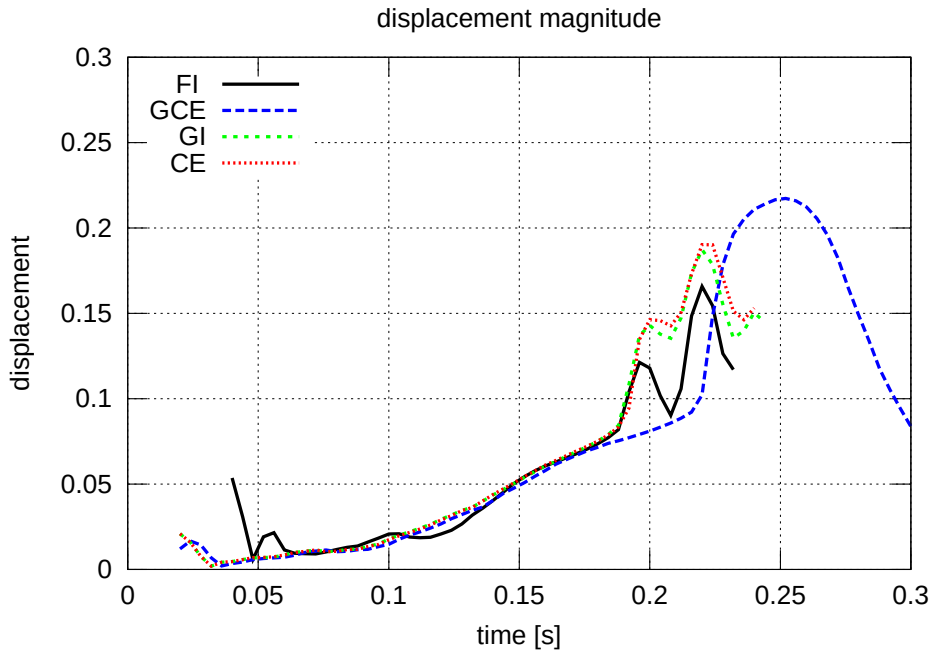
(a) $\delta t = 5 \cdot 10^{-4}$ s



(b) $\delta t = 10^{-3}$ s

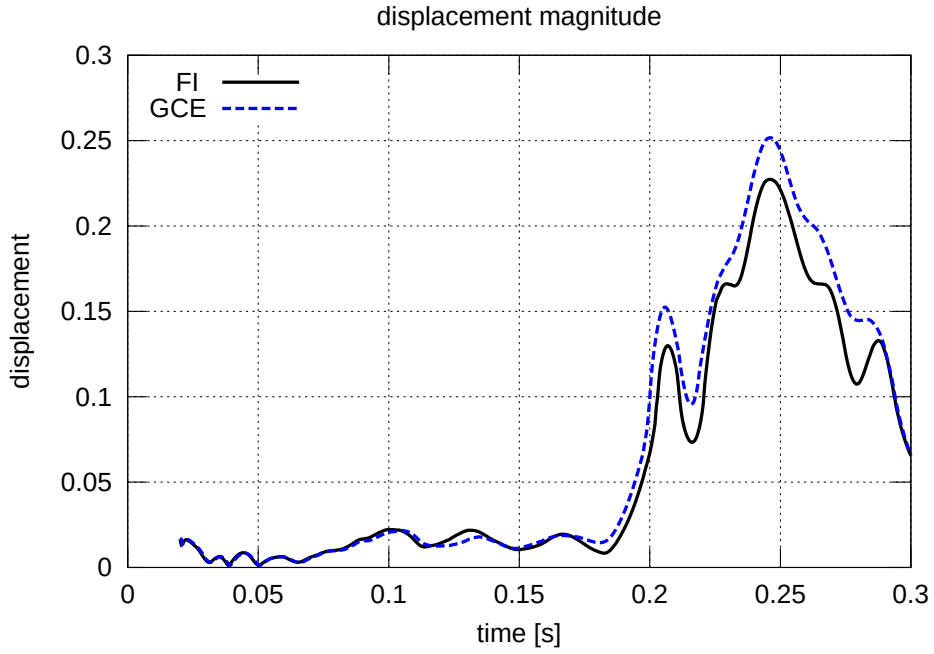


(c) $\delta t = 2 \cdot 10^{-3}$ s

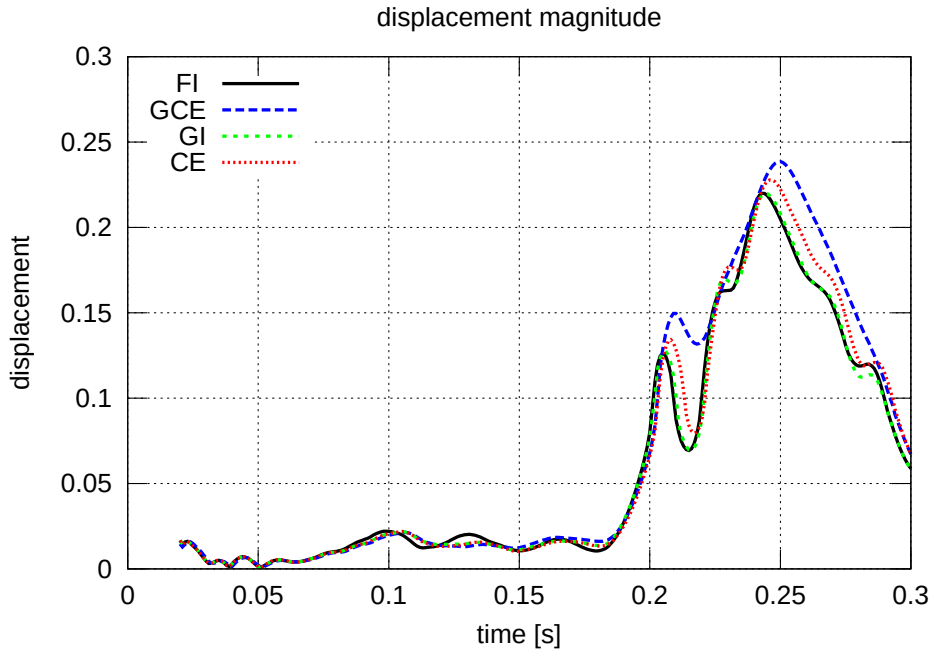


(d) $\delta t = 4 \cdot 10^{-3}$ s

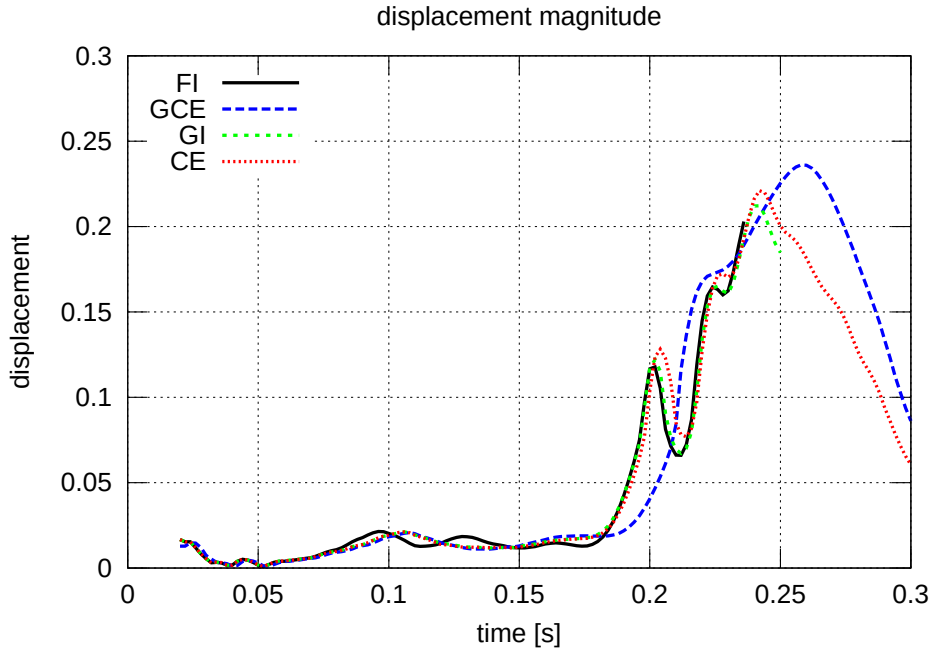
Figure 5.32: Time history of the displacement magnitude (in cm) averaged on the location A of Figure 5.27.



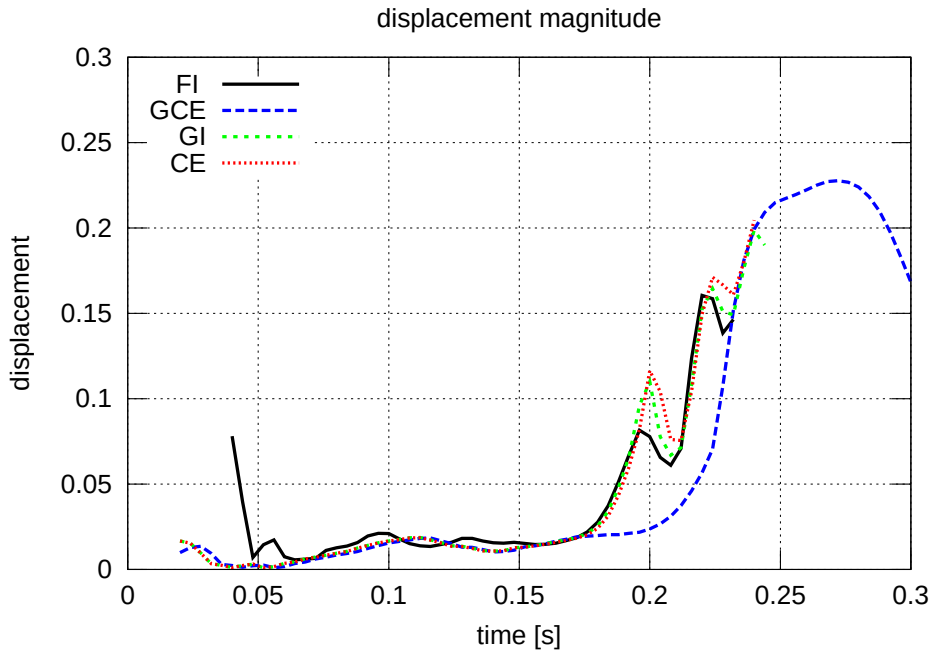
(a) $\delta t = 5 \cdot 10^{-4}$ s



(b) $\delta t = 10^{-3}$ s

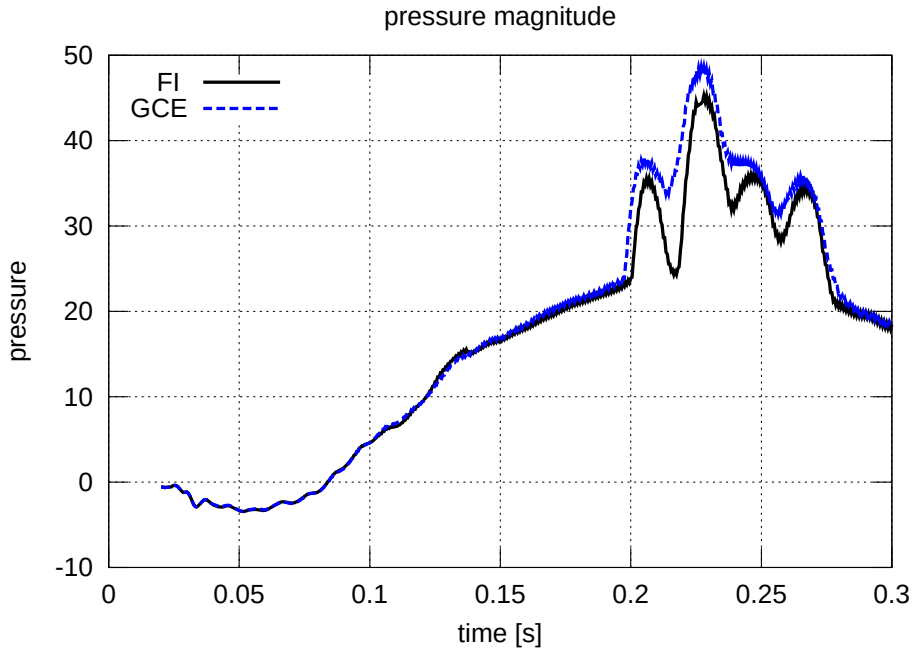


(c) $\delta t = 2 \cdot 10^{-3}$ s

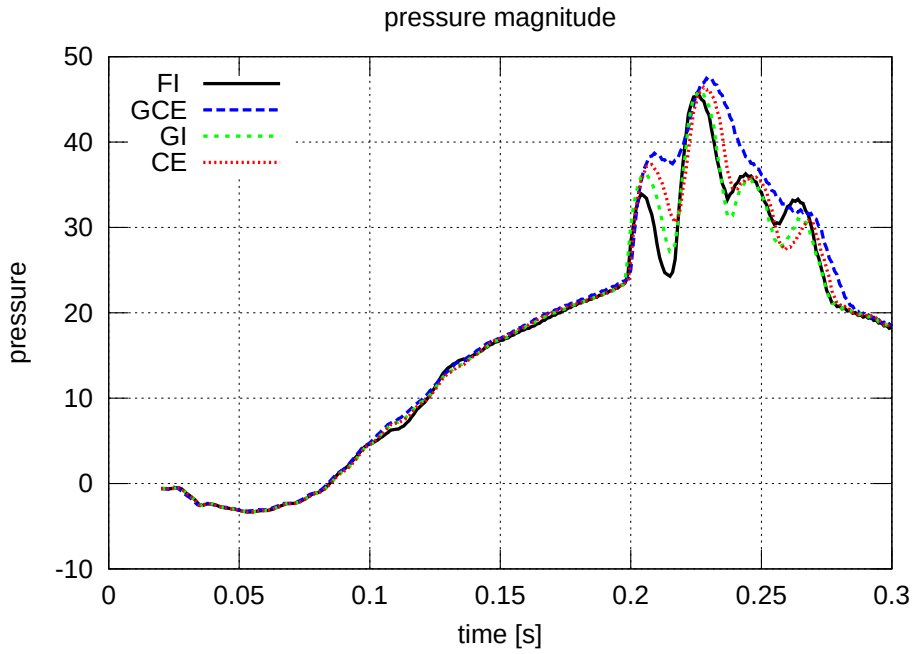


(d) $\delta t = 4 \cdot 10^{-3}$ s

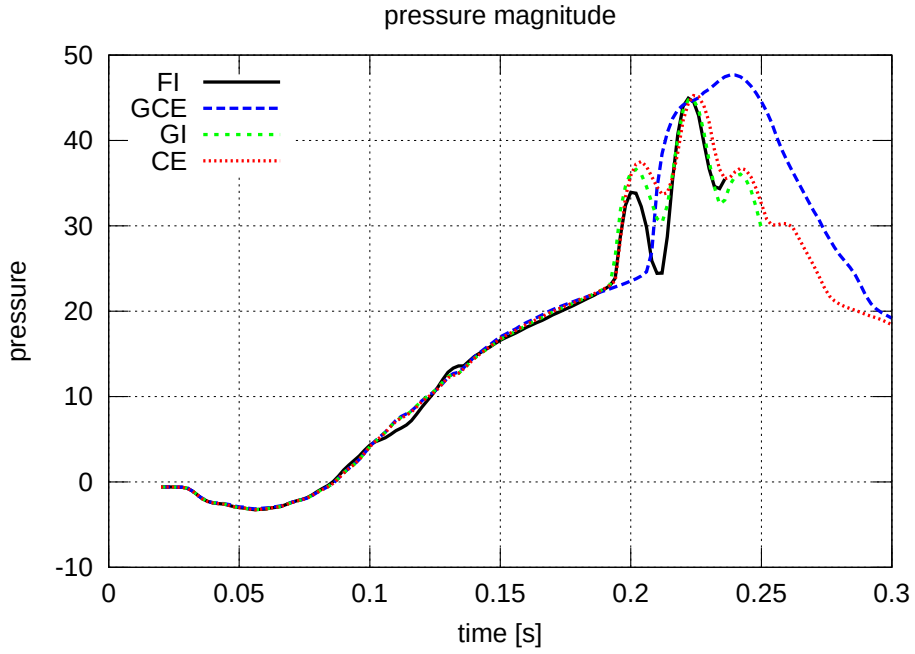
Figure 5.33: Time history of the displacement magnitude (in cm) averaged on the location B of Figure 5.27.



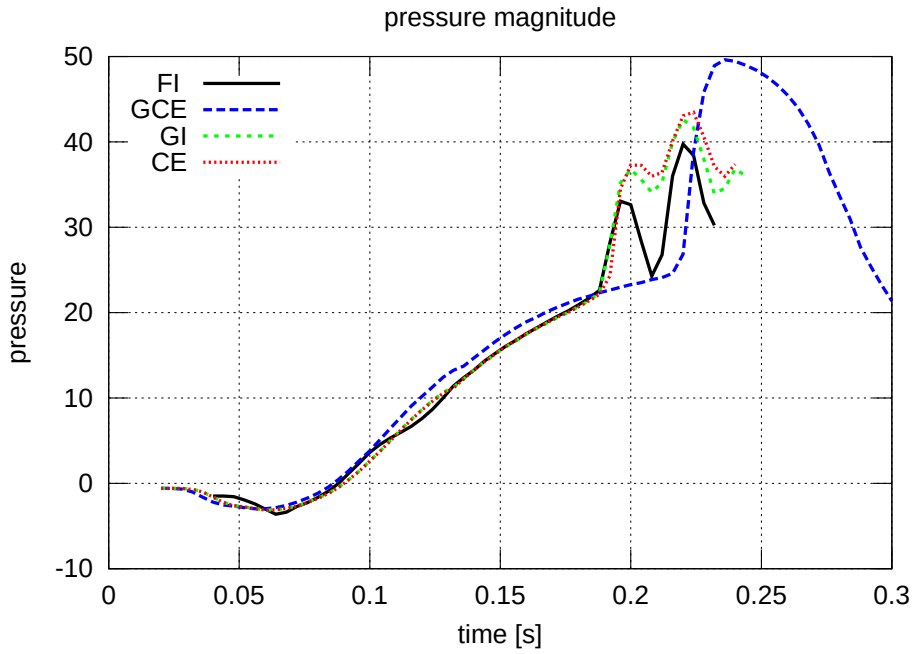
(a) $\delta t = 5 \cdot 10^{-4}$ s



(b) $\delta t = 10^{-3}$ s

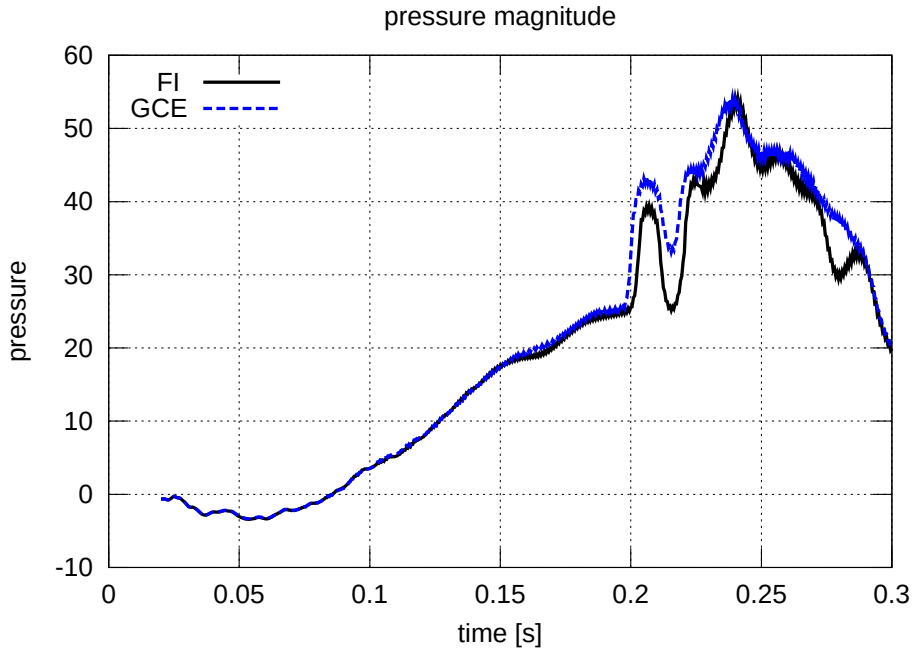


(c) $\delta t = 2 \cdot 10^{-3}$ s

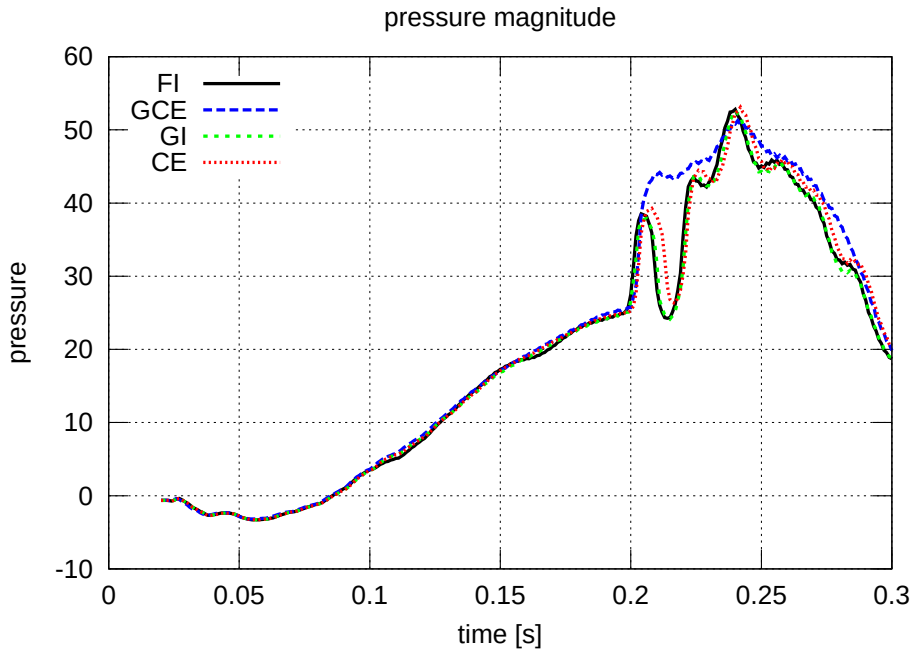


(d) $\delta t = 4 \cdot 10^{-3}$ s

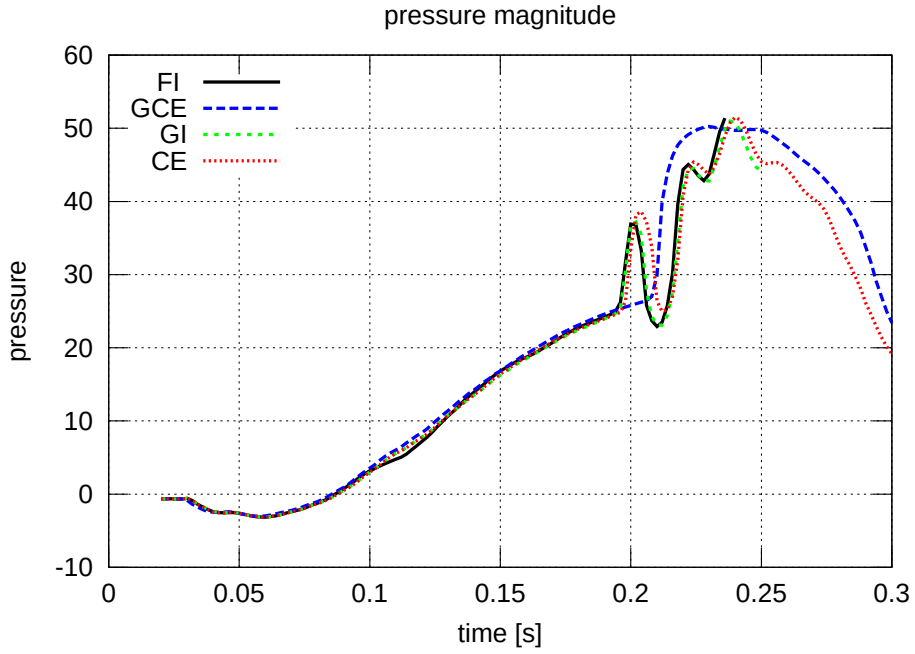
Figure 5.34: Time history of the pressure (in mmHg) averaged on the location A of Figure 5.27.



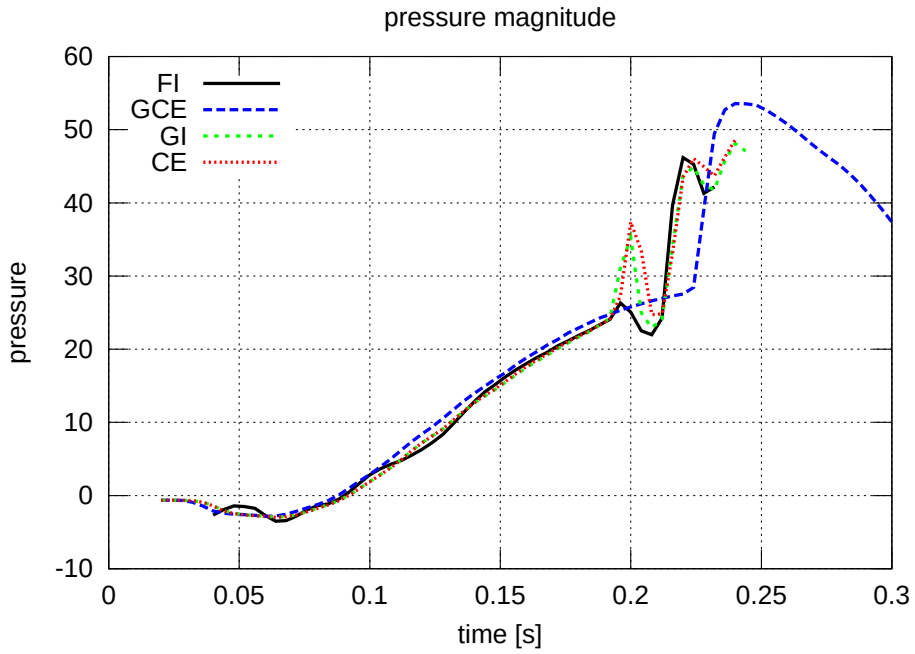
(a) $\delta t = 5 \cdot 10^{-4}$ s



(b) $\delta t = 10^{-3}$ s

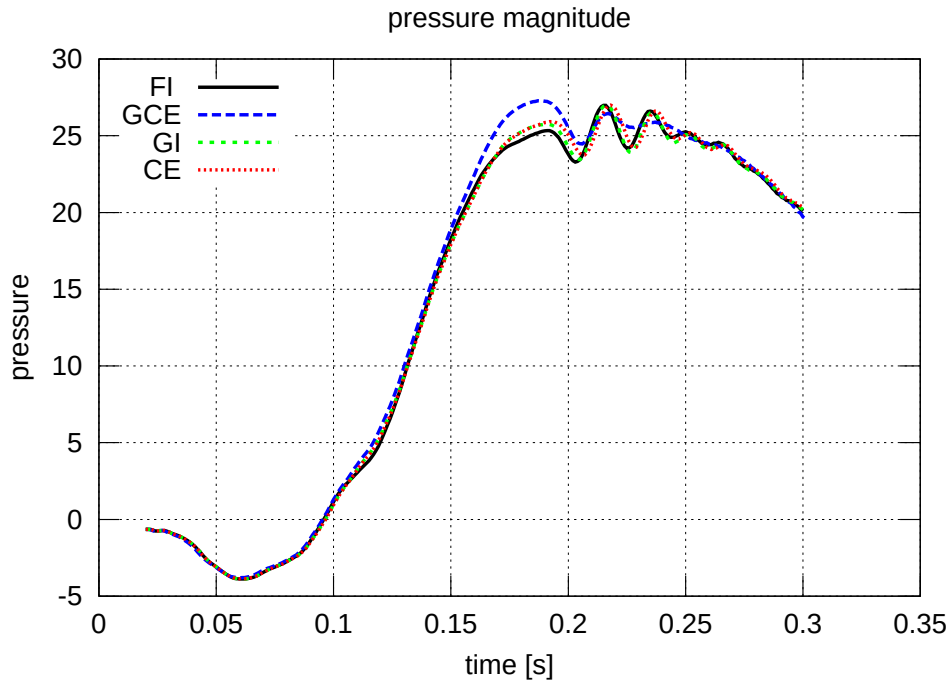


(c) $\delta t = 2 \cdot 10^{-3}$ s

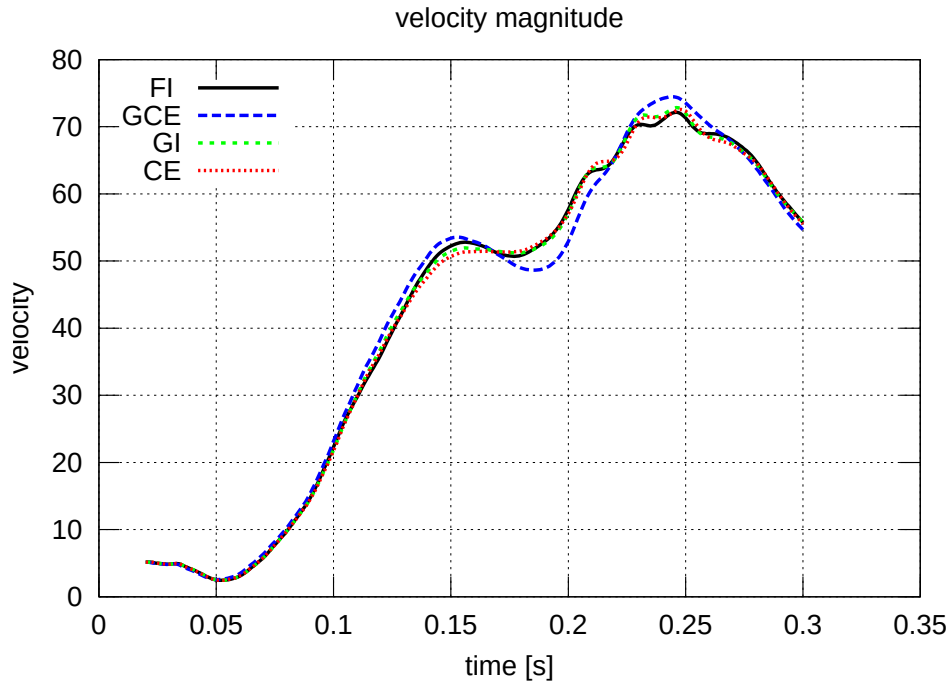


(d) $\delta t = 4 \cdot 10^{-3}$ s

Figure 5.35: Time history of the pressure (in mmHg) averaged on the location B of Figure 5.27.



(a) mean pressure



(b) mean velocity

Figure 5.36: Values of the mean pressure (in mmHg) and mean velocity (in $^{cm}/_{s}$) on a section of the descending aortic arch, for the different time discretizations and with a fixed time step of $\delta t = 10^{-3}$.

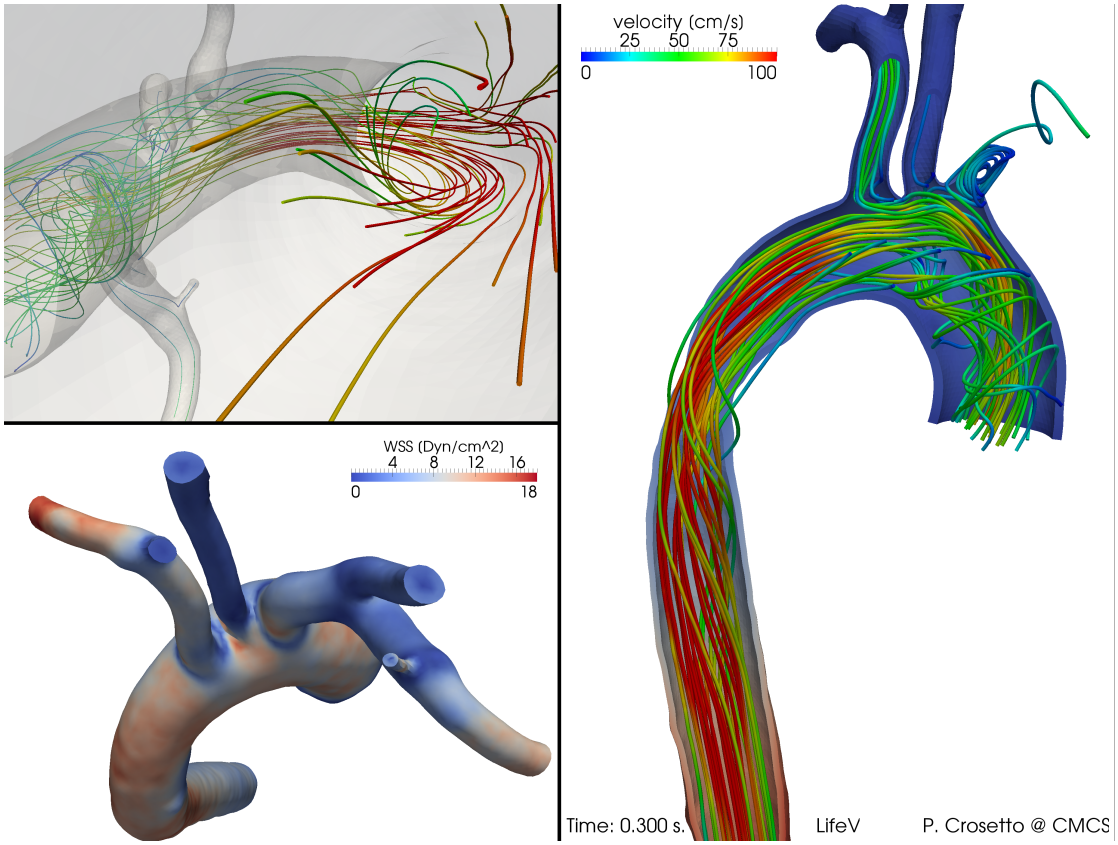


Figure 5.37: Aorta simulation. Streamlines and WSS at the end of the systolic phase in the second heartbeat.

5.5.2 Exact versus Inexact Newton Method

In the cases where the geometry is handled implicitly a Newton method is employed. The number of Newton iterations as expected varies depending on the time step chosen and on the phase of the heartbeat considered. In particular in proximity of the systolic peak, due to the large displacements and velocities, the nonlinearities due to both convective term and geometry become important, and the number of Newton iterations increases. Due to the fact that the convergence of Newton is not assured when the initial guess is far from the solution, for large time steps in some cases we experienced convergence problems in proximity of the systolic peak. Since the first order methods have usually a larger convergence ball then the Newton method (see e.g. [ESW05, Ch. 7]), a fixed point or inexact Newton method could be employed to reach a closer initial state for the Newton method[27]. Here we consider only first order extrapolations and discretizations. Testing more accurate time discretizations is currently under investigation.

We investigate in the simulation of the GI solution the effect of taking into account the shape derivatives in the Jacobian of the FSI problem. The number of Newton iterations performed during systole (i.e., from $t = 0.16\,\mathrm{s}$ to $t = 2.4\,\mathrm{s}$) in both cases is reported in Figure 5.38 for a time step of $\delta t = 0.004\,\mathrm{s}$, the relative tolerance (see Section 3.1) is set to $\epsilon_r = 10^{-4}$. Using the exact Jacobian matrix in this case not only improves the solution in terms of nonlinear iterations: also the performances in terms of CPU time are better. We report the computational cost per time level in Figure 5.38. We recall that if a smart implementation of the shape derivatives block assembly is exploited (see Section §3.4), then increasing the number of processors the overhead becomes negligible with respect to e.g. the matrix-vector multiplications required for the GMRES solution. At the first time iteration considered, both exact and inexact Newton perform three iterations (Figure 5.38 on the left), however the time spent in both cases (Figure 5.38 on the right) is almost the same. In our algorithm the main extra computational cost for the construction of the Jacobian J_{FI} with respect to the matrix of the system (computed to evaluate the residual) is in the shape derivatives block computation. This block assembly performed once per Newton step has a computational cost of about $0.8\,\mathrm{s}$ for the simulation here presented, which is negligible with respect to the total time ($\approx 1\%$ of the time per Newton iteration) and since it does not require inter processors communication (like all the matrix assembling steps) it is almost perfectly scalable.

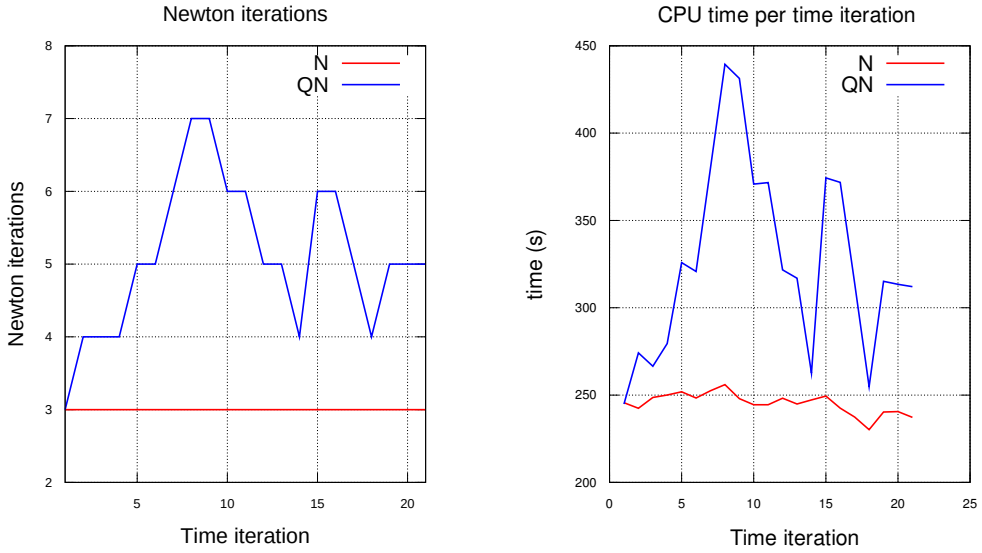
We remark that for both the inexact and exact Newton schemes the preconditioner used is P_{AS-DN} (4.2.20), thus the shape derivatives block is not neglected in the preconditioner. Using this preconditioner the number of iterations and the time needed for the solution of the linear Jacobian system is not remarkably affected by the presence of the shape derivatives block.

Remark 5.5.1. *These results are somehow in disagreement with what is stated in Bazilevs et al. [BGH+09], where an inexact Newton is employed to solve the geometrical nonlinearity, neglecting the shape derivative terms, for the simulation a Ventricular Assisted Device (VAD)*

[27] Many efficient ways to find an accurate initial state for Newton have been studied in literature. For instance in [BC10b] a nonlinear coarse problem is solved at each time iteration just to initialize the nonlinear solver (leading to the *two levels Newton* method), in [GK10, TDV07] efficient extrapolation techniques based on POD are used in order to accelerate the convergence of a linear or nonlinear solver by improving the initial guess. In [KGF+09, KW09] a vector extrapolation technique allows to move gradually from fixed point iterations to Newton.

implanted in a thoracic aorta. There the authors do not find great advantages in considering the shape derivatives terms in the Jacobian matrix. We conclude that the debate is still open and so far there is no method which behaves better than the others in all the situations.

Remark 5.5.2. *The Newton scheme used is not completely exact, because the fluxes are prescribed as integrals on moving boundaries (the vessel terminations are not clamped), and the shape derivatives of these terms (cf. Section §3.4.2) are not taken into account in our Jacobian.*



(a) (b)

Figure 5.38: Aorta simulation: (a) number of Newton iterations and (b) CPU time per time step. Exact Newton (N) is compared versus inexact Newton (QN) for an interval during the systolic peak (from $t = 0.16\,\mathrm{s}$ to $t = 0.24\,\mathrm{s}$) for a GI simulation with time step $\delta t = 0.004\,\mathrm{s}$

	aortic arch			Thoracic aorta	
	lower	middle	upper	interior	exterior
Rigid walls	2.4	3.7	2.6	3.7	4.3
Compliant walls	2.7	2.2	1.9	3.5	3.3

Table 5.4: Time average of WSS ($^{\mathrm{dyn}}/_{\mathrm{cm}^2}$) at different regions of the aorta. Values are reported for 3D compliant and rigid walls simulations.

5.5.3 FSI (GCE) versus Rigid Walls and 1D

In this subsection we complete the comparisons of Section §5.4.5 between two different of FSI simulations and a rigid walls one. Below we compare the FSI simulation also with a 1D simulation in terms of pressure and flux on different sections. Furthermore we provide a table with the differences in WSS for the compliant and rigid walls cases.

In order to compare a compliant FSI simulation with a standard Navier-Stokes simulation in a domain with rigid walls, we set up a problem with similar boundary conditions as those described above. We impose the fluxes at all the outlets, and we substitute the inlet flux with a time dependent uniform normal stress condition on the inlet section. Indeed due to the mass conservation constraint in the rigid walls case we cannot impose arbitrary fluxes at all the terminations, since the total flux must sum to zero at every time iteration. Figure 5.39 reports the history of the velocities for the rigid walls case at different locations of the artery.

For this comparison in the FSI simulation instead of the flux inlet boundary condition we impose the normal stress, also obtained from the 1D simulation.

Note that the energy estimate for Navier-Stokes equations (in both Eulerian and ALE formulation) holds just for Neumann boundary conditions in the outward direction [QV94]. However we do not observe instabilities in the simulations presented. We remark also that these results were obtained with the GCE time discretization. Hence the qualitative comparison may be affected by the dissipative behavior of this scheme.

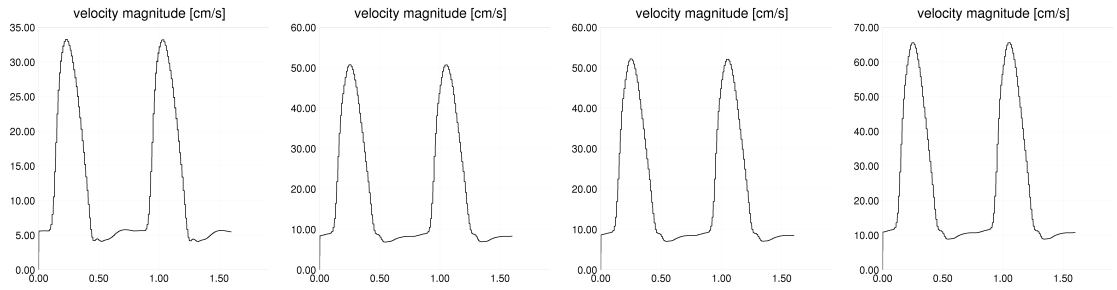


Figure 5.39: Rigid walls Navier–Stokes simulation. History of the mean velocity over the sections S_1, S_2, S_3 and S_4 represented in Figure 5.15. The imposition of an inlet normal stress instead of a flux leads to a nonzero positive flux through the aortic valve (Section S_1) also when the valve is supposed to be closed.

Another comparison that is certainly quite interesting from a modeling point of view, is between the 3D FSI and 1D wave propagation models. In Figure 5.40, pressure and flow waveforms obtained at 2 locations are superimposed. The qualitative features of the pressure

waveforms are in good agreement. The amplitude of the flow waveforms is slightly different, mainly in systole, at both aortic cross and thoracic aorta.

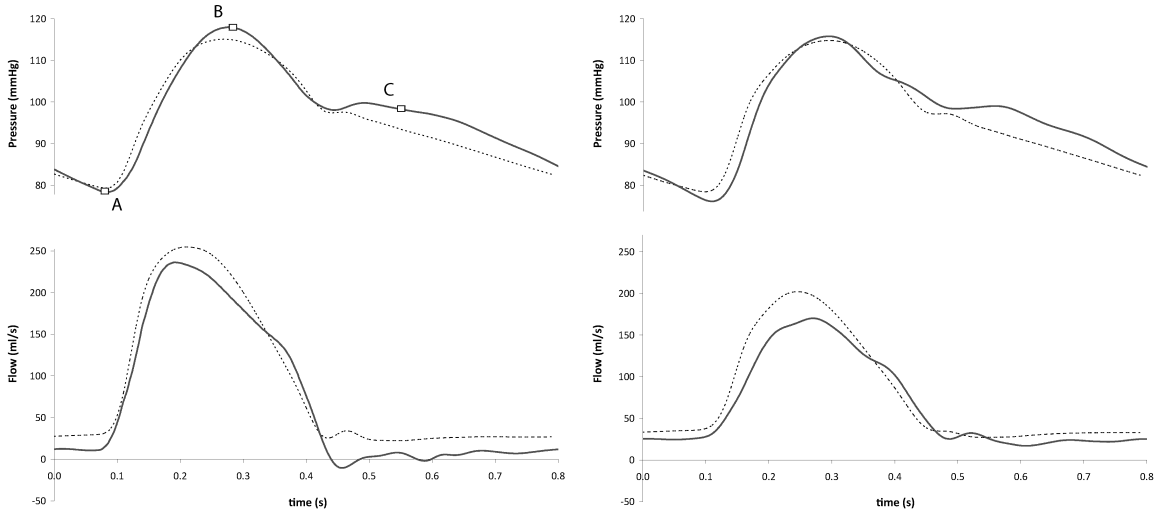


Figure 5.40: Pressure (top) and flow (bottom) waveforms computed by the FSI simulation (continuous line) and the 1D model (dashed line). Locations are at the aortic arch (left) and thoracic aorta (right). Fluxes are imposed at the inlet and all the outlets.

Scalability and Parallel Performances

6.1 Introduction

In this chapter we investigate the parallel performance of the FSI solver in LifeV. In the previous chapter we already reported some results concerning CPU time and computational cost. Here we complete these informations by studying the behavior of the solver when varying the number of MPI processes and/or the problem size.

To organize the discussion we report first a definition of scalable algorithm. Indeed depending on the field the concepts of weak and strong scalability may have slightly different meanings. We report then weak and strong scalability results for some of the preconditioners described in Section §4.3 on benchmark and physiological geometries.

6.2 Scalability and Results

In parallel computing a very important concept is the one of *scalability*, which is a measure of the capability of an algorithm to perform when increasing workload and/or the number of parallel processes.

Although the term *scalability* is widely used throughout literature in several application fields, many different interpretations and operational definitions coexist. This issue is addressed e.g. in Keyes [Key98] where a general definition is given which makes abstraction from the specific application considered. There the definition of scalability requires a *performance metric* which is a function of several parameters π_i. The performance function chosen strongly influences the type of scalability considered, which is usually reached for a bounded region in the parameters space.

The performance function that we consider is the *parallel efficiency*. It is usually measured in floating point operations per seconds when the hardware scalability is investigated. In our application we define the parallel efficiency of an operation as the inverse of the time spent for such operation. We express the performance η in terms of two parameters: the *workload* and the *processors number*. The workload in our case is the number of degrees of freedom of the problem. Two types of scalability, listed below, are usually considered.

- Strong scalability: in a strongly scalable algorithm doubling the number of processors (and keeping fixed the other parameters) the efficiency doubles as well. In practical cases this type of scalability is never reached because of the hardware limitations: the overall time is the sum of the time spent for the computations and the time spent for the communications between the processors. While for the former operations the

performance can actually double when doubling the number of processors, for the latter this does not hold true (see also the considerations in [Key98], where the overall efficiency is split into *algorithmic* and *implementation* efficiency).

- Weak scalability: in a weakly scalable model the efficiency is the same as long as the ratio $\frac{workload}{\#processors}$ is constant. The lack of weak scalability can be caused by either a loss of performance of the *computing* part, performance that should not decrease when increasing the number of processes, or by an increase of the *latency time* introduced by the hardware communication; this time should ideally remain constant when increasing the number of processors.

We remark that different definitions of performance lead to different types of scalability. For instance in domain decomposition often the scalability is interpreted as the independence of the convergence rate of an algorithm from the number of subdomains, or from the characteristic mesh size h. In this case the performance quantity is related to the number of iterations (instead of time), this type of scalability does not imply the scalability in the sense that we just introduced.

In our simulations the number of subdomains in the domain decomposition framework corresponds to the number of MPI processes. To measure the weak scalability it makes sense to double the length of our computational domain (the vessel) when doubling the CPUs. In this case the weak scalability measures the capability of the algorithm to increase the problem physical size.

Another important indicator for the parallel efficiency is the *speedup* σ, whose definition is only related to the timing T:

$$\sigma = \frac{T_{serial}}{T},$$

where T_{serial} is the time spent in the serial execution. We say that an algorithm is strongly scalable if $\sigma = \#processors$.

6.2.1 Strong Scalability Tests

In this section we report the numerical behavior of the preconditioner choices presented so far. In two of the cases considered, i.e., the GCE and the CE systems described in Chapter 4, our preconditioning strategy shows to be effective in terms of computational time. As a general comment, splitting the matrix into factors before building the operators P_{AS} yields better computational times and lower number of GMRES iterations (see Figures 6.2(a) to 6.7(b)). The theoretical justification stems from the analysis in Section §4.3.4.

A one level AAS preconditioner is used in all the simulations (for the different factors or for the whole matrix), it has a two layers overlap between the partitions (this choice is motivated by empirical observations). The finite elements loop to assemble the system matrix is almost perfectly scalable, in both weak and strong senses. The same holds true for the computation of the AAS preconditioners, which involves the LU factorization of the local matrices. The main lack of scalability is due to the increase of GMRES iterations when the processors number grows. This can be probably avoided by employing multilevel domain decomposition preconditioners for the factors (instead of one level AAS, cf. Section §4.3.1) and keeping the same modular approach described here.

The behavior of our preconditioners is tested on unstructured 3D cylindrical meshes with different characteristic lengths h, and on physiological 3D geometries as well, computed

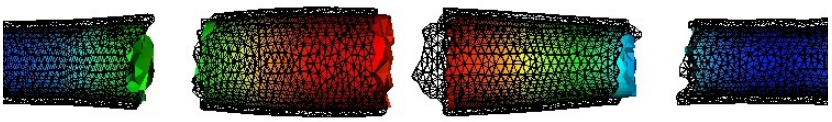


Figure 6.1: The cylindrical geometry used partitioned into 4 subdomains. One processor holds one solid and one fluid subdomain. However, in general, the two are not matching on the interface (despite the impression given by the picture) and the partitioning is done independently for the two meshes.

through segmentation of medical images (e.g. CT scans or MRI). The finite elements chosen are $P1-P1$ tetrahedra for the fluid, stabilized using the interior penalty technique described in [PB05], cf. Section §2.7.1, and P1 tetrahedra for the structure.

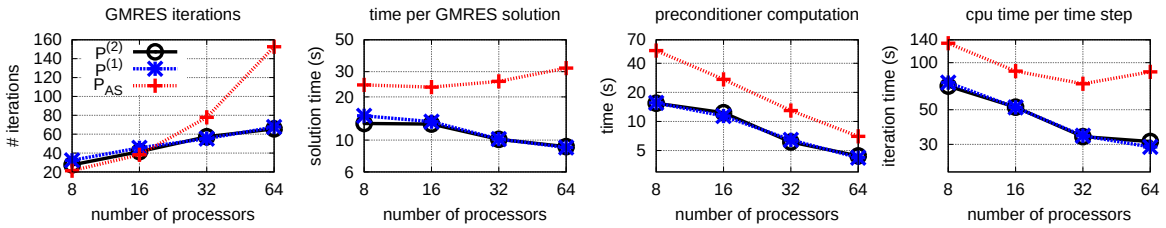
The tests reported in this section, unless otherwise specified, were performed on the cluster Callisto at EPFL.

Strong Scalability for GCE

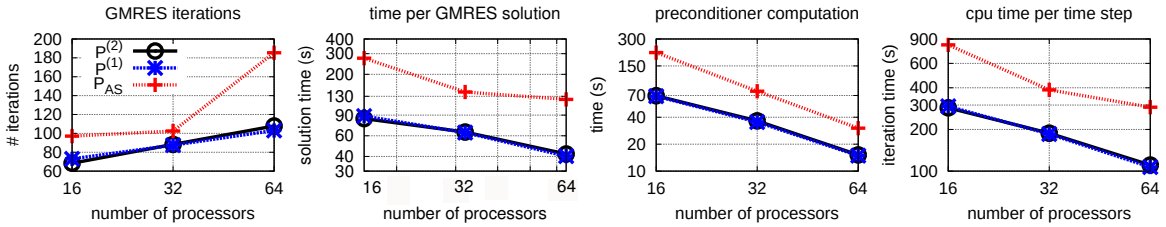
A benchmark geometry, similar to the one proposed in [GV03] (and used e.g. in [DDFQ06, KGF$^+$09, FM05]) consists of a straight cylinder of length 10 cm and with radius 0.5 cm representing the fluid domain, surrounded by a structure of constant thickness 0.1 cm (see Figure 6.1). The inlet boundary condition on the fluid domain is a pressure step function, taking the constant value $p_{in} = 1.33 \cdot 10^4 \,\mathrm{dyn/cm^2}$ for $t \leq 0.003\,\mathrm{s}$. The boundary conditions for the outlet and for the external structure are of Neumann homogeneous type. The time step used is $\delta t = 10^{-3}\,\mathrm{s}$ while the parameters characterizing the model are fluid viscosity $\mu_f = 0.03\,\mathrm{P}$, the Young modulus $\epsilon = 3 \cdot 10^6 \,\mathrm{dyn/cm^2}$, the Poisson ratio $\nu = 0.3$ and the mass densities of fluid and structure $\rho_f = 1.0\,\mathrm{g/cm^3}$ and $\rho_s = 1.2\,\mathrm{g/cm^3}$. Because of the GCE time discretization adopted, the problem is linear. The tolerance for the GMRES solver is 10^{-6}. The values reported in the following are the average over the first 30 time iterations.

The simulations run on the benchmark geometry show that the modular preconditioners P_{AS-DN1} and P_{AS-DN2} ($P^{(1)}$ and $P^{(2)}$ respectively in the figures) are more efficient than AAS (a classical additive Schwarz preconditioner on the whole matrix) in terms of computational time and GMRES iterations. Furthermore the gap between these preconditioning strategies increases with the processors number. This result is validated on two different meshes composed by a total of $169'267$ and $578'594$ tetrahedra, corresponding to respectively $216'441$ and $630'468$ degrees of freedom (in Figures 6.2(a) and 6.2(b)), and varying the ratio between the mass densities $r = \frac{\rho_f}{\rho_s}$ (in Figure 6.3).

We observe that in view of the analysis performed in Section §4.3.4 the mild growth of the GMRES iterations in the modular cases can be explained with a boundedness of the singular value σ_Σ^{max}. We recall that the preconditioners P_{AS-DN1} and P_{AS-DN2} are split into factors, and each factor contains either the fluid or the structure block. The lack of scalability, due to the increasing number of GMRES iterations, is related to our choice of using the one level AAS preconditioner for all the factors in the tests performed. Thanks to Proposition 4.3.1 the quality of the preconditioner is influenced mainly by the preconditioners chosen for the different factors. Thus an improvement of the preconditioner can be achieved by improving the preconditioners for the subproblems.



(a) Simulations run on the coarser tube mesh.



(b) Simulations run on the finer tube mesh.

Figure 6.2: GCE — Strong scalability. $P^{(1)}$ and $P^{(2)}$ correspond to P_{AS-DN1} and P_{AS-DN2} respectively.

Other strong scalability results performed on different architectures, with different space discretizations, are resumed in Figures 6.4(a), 6.4(b) and 6.4(c). These simulations are run on a segment of thoracic aorta, and the linear system is preconditioned by P_{AS-DN1}. Figures 6.4(a) to 6.4(c) show the total CPU time per time step, the time to compute the preconditioner, the time to solve the linear problem and the number of GMRES iterations at each time step. The number of tetrahedra is $578'594$.

Figures 6.4(a) and 6.4(b) refer to the same simulation on the Cray XT4 and on the Blue Gene/P, where the finite elements used for the fluid subproblem are the stabilized P1-P1 elements. The total number of unknowns in this case is $630'468$. With the same number of processors, the simulation on an XT4 takes about 3.5 times less CPU time than on a BlueGene/P (this is a consequence of the different CPU frequency); in contrast, the scalability of the overall simulation is better on the BlueGene/P. Figure 6.4(c) represents the same type of simulation with inf–sup stable P1Bubble-P1 (*mini*) finite elements for the fluid problem. On BlueGene/P we notice an improvement of the scalability with 512 processors. The CPU time is smaller than for P1-P1 since the stencil of the matrix is smaller.

We remark that in these simulations the number of GMRES iterations does not increase when the number of processors increases.

Strong Scalability for CE

We run the same simulation as for the GCE on the straight cylinder. The tolerance of the linear solver is set to 10^{-6} while the relative tolerance for the Newton method is $\epsilon^r = 10^{-5}$.

The preconditioners compared in Figure 6.5(a) are $P_{AS}(J_{CE})$, P_{AS-QN} (without splitting into factors) and P_{AS-GS} (two factors splitting) among those listed in Section §4.3.3. We observe that the preconditioner P_{AS-QN} is cheaper to compute than $P_{AS}(J_{CE})$, while it has

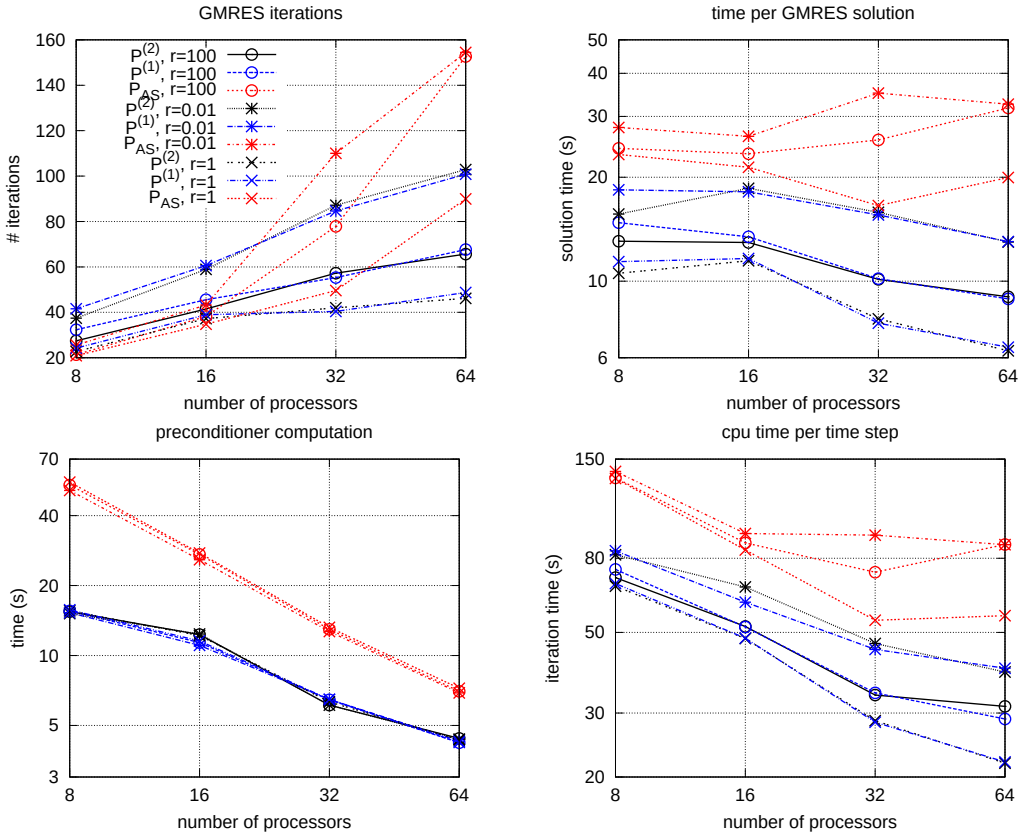
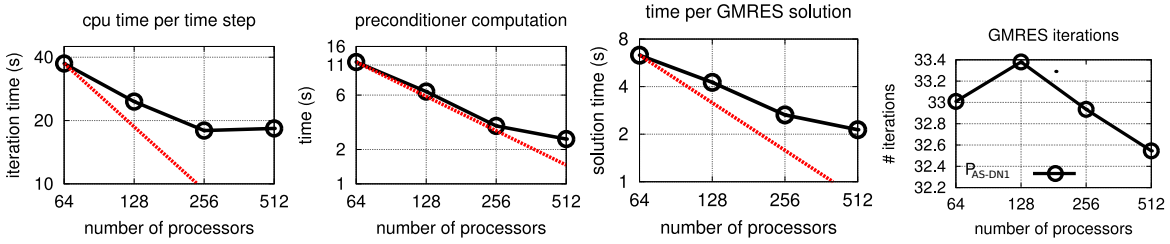
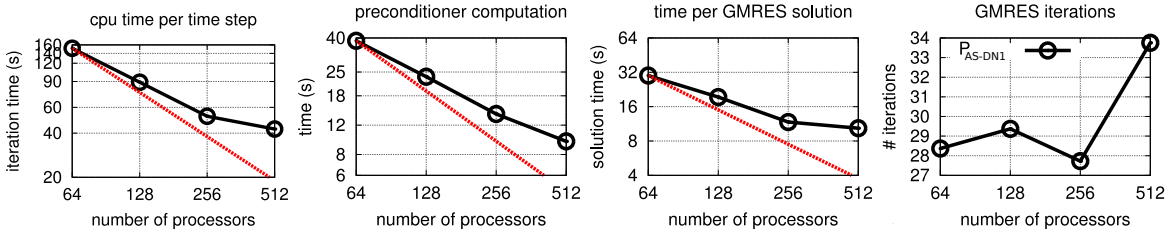


Figure 6.3: GCE — Comparison on the coarser tube mesh between the results obtained for different values of the ratio $r = \rho_f/\rho_s$.

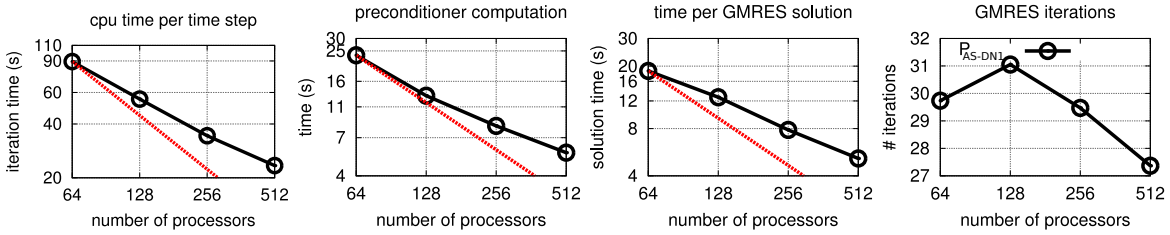
the same behavior in terms of GMRES iterations. Although P_{AS-GS} is cheaper to compute than the other choices it worsen faster when increasing the number of processors. This can be due to the growth of the singular value σ_Σ in the estimate (4.3.10a), when augmenting the number of processors. Thus P_{AS-GS} seems not to be a choice well suited for a massively parallel framework.



(a) stabilized P1P1 finite elements for fluid sub-problem, results on Cray XT4.

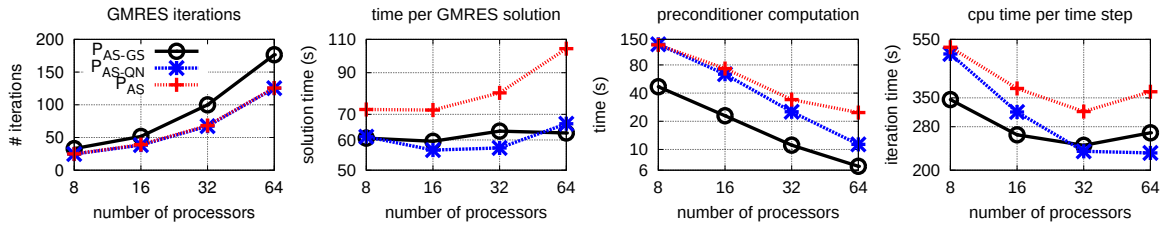


(b) stabilized P1P1 finite elements for fluid sub-problem, results on BlueGene/P.

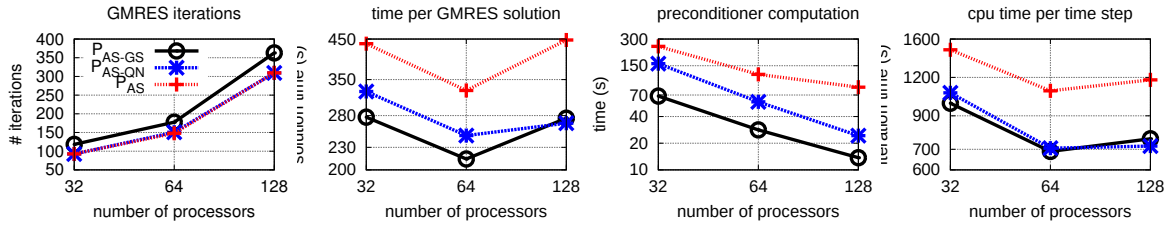


(c) P1BubbleP1 finite elements for fluid sub-problem,results on BlueGene/P.

Figure 6.4: GCE time discretization. The dashed line represents the ideal case.



(a) coarser tube mesh.



(b) finer tube mesh.

Figure 6.5: CE — Strong scalability on a straight cylinder.

Among the preconditioners presented in Figures 6.5(b) and 6.5(a) the most convenient

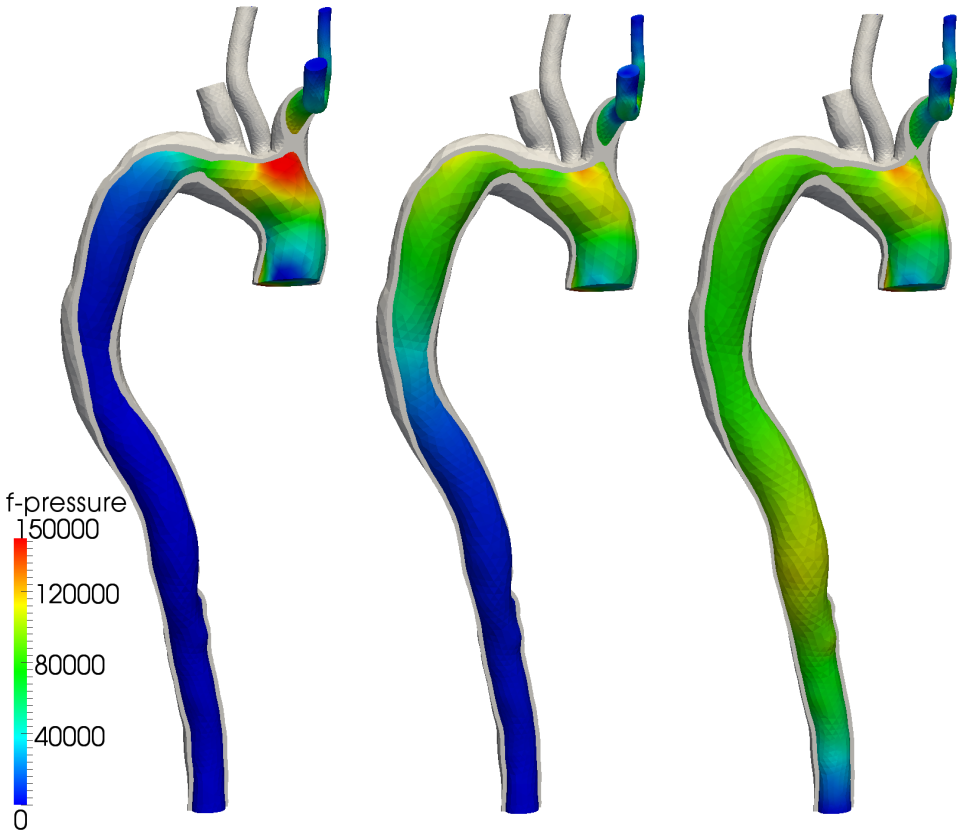


Figure 6.6: Blood pressure (in $^{\mathrm{dyn}}/_{\mathrm{cm}^2}$) and the deformation of an aorta at $t = 0.015\,\mathrm{s}$, $t = 0.03\,\mathrm{s}$, and $t = 0.075\,\mathrm{s}$.

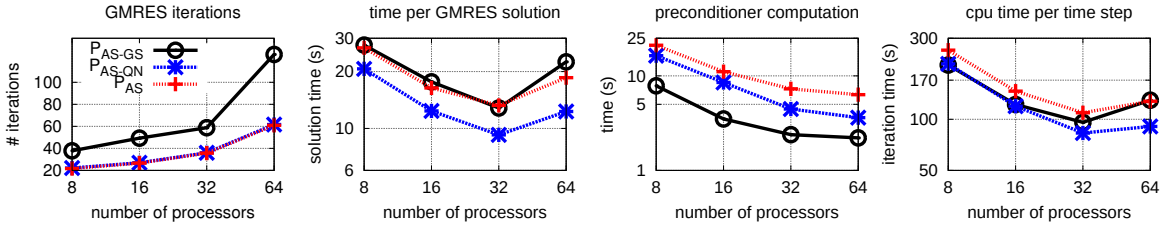
choice turns out to be P_{AS-QN}. In Section §6.2.2 another preconditioner featuring a three factor splitting is considered as well, and it is shown to improve the performances in terms of both computational time and iterations number with respect to the preconditioners considered up to now.

Physiological geometries

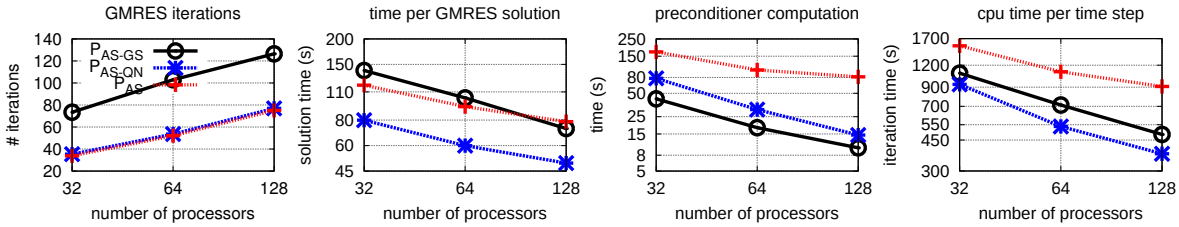
The preconditioners devised for the CE system are tested on physiological geometries (Figure 6.6), obtained through segmentation of medical images using VMTK[28], see [Bon09].

The physical parameters chosen for the models (1.4.9) and (1.5.11) are $\mu_f = 0.35\,\mathrm{P}$, $\epsilon = 4 \cdot 10^6\,{}^{\mathrm{dyn}}/_{\mathrm{cm}^2}$, $\nu = 0.48$, $\rho_s = 1.2\,{}^{\mathrm{g}}/_{\mathrm{cm}^3}$ and $\rho_f = 1.0\,{}^{\mathrm{g}}/_{\mathrm{cm}^3}$. Since the purpose of these simulations is mainly to test the behavior of the preconditioners rather than to observe the dynamic of the flow during an entire heartbeat we only consider the mean values over the first 30 time steps (0.03 s). The geometry considered in these simulations represents an aorta,

[28] `http://www.vmtk.org`



(a) coarser mesh.



(b) finer mesh.

Figure 6.7: CE — Simulations on the aorta on Figure 6.6.

starting from the aortic valve, including the aortic arch and the thoracic aorta. The inlet pressure imposed at the aortic valve is taken from physiological measurements corresponding to the beginning of an heartbeat cycle $p_{in} \approx 1.1 \cdot 10^5 \, \mathrm{dyn/cm^2}$. The boundary conditions imposed at the outflows and on the external wall are of Neumann homogeneous type. Since no spurious reflection waves are originated in the first 30 time steps (see e.g. [Nob01]), there is no need to impose absorbing boundary conditions at the outflows. The blood is considered at rest at the beginning of the simulation. The simulations are performed on two different meshes, with respectively 105′810 and 380′690 tetrahedra for a total of 135′000 and 486′749 degrees of freedom. We keep the same time step $\delta t = 10^{-3}$ s used in the previous simulations.

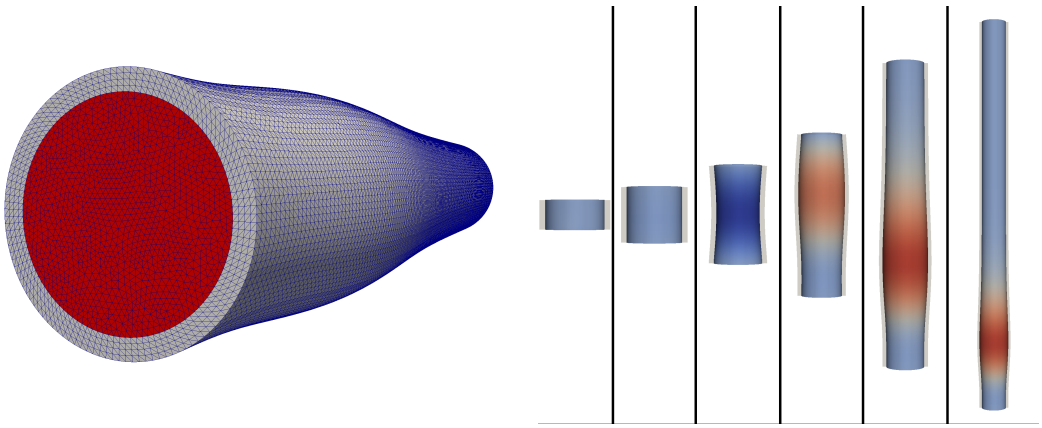
We can notice that the number of iterations relative to P_{AS-GS} worsens faster than the others when the number of processors increases (Figures 6.7(a) and 6.7(b)). This phenomenon was already observed in the previous benchmark tests.

The increase of the global computational time when passing from 32 to 64 processors in the coarser case is partly due to the relatively small mesh size of the problem addressed and partly to the communication time affecting both the GMRES solution and the preconditioner computation.

6.2.2 Weak Scalability Test

We want to set up the same benchmark test with a similar geometry, the same physical parameters and boundary conditions as in the previous section, but keeping approximately constant the ratio $\frac{\#dof}{\#processors}$. To this aim we build a series of meshes of the tube in Figure 6.8(a) by varying its length. The new meshes are built with the free mesh generator Gmsh[29]. The fluid part is generated by building a 2D Delaunay mesh of a circle and extruding it in the axial direction. The structured solid mesh is generated extruding the lateral surface in the normal direction. The solid is composed of four layers (see Figure 6.8(a)). We used for these simulations the Cray XT5 supercomputer Rosa.

[29] `http://geuz.org/gmsh/`



(a) Benchmark mesh (b) Geometries

Figure 6.8: Mesh (a) and geometries (b) used for the weak scalability test. Deformation at time $t = 0.01\,\text{s}$.

The different geometries considered have a length varying from 0.5 cm to 16 cm and are represented in Figure 6.8(b). The smaller problem is solved using either 12 or 24 MPI processes, then the number of processors is doubled every time the length of the tube doubles.

The finite elements matrices in this case are considerably worse conditioned than in the case considered for the strong scalability, the characteristic element size being smaller. The dimensions for the fluid and the solid meshes considered are resumed in Tables 6.4(a) and 6.4(b) respectively. The larger problem considered, corresponding to the 16 cm pipe in Figure 6.8(b), features 3′517′839 degrees of freedom. The results plotted are the mean values over the first 10 time iterations.

(a) fluid meshes.

Length	# Elements	# Vertices
0.5 cm	75’480	14’487
1 cm	149’520	27’405
2 cm	301’920	53’997
4 cm	603’840	106’677
8 cm	1’207’680	212’037
16 cm	2’415’360	422’757

(b) solid meshes.

Length	# Elements	# Vertices
0.5 cm	27’840	6’380
1 cm	55’680	12’180
2 cm	111’360	23’780
4 cm	222’720	46’980
8 cm	445’440	93’380
16 cm	890’880	186’180

Table 6.5: Table resuming the characteristics of the meshes used for the weak scalability test, represented in Figure 6.8(b).

Weak Scalability for GCE

In this case the number of linear iterations using the classical AAS preconditioner increases much faster. The new preconditioning strategies proposed show to be more robust. We show

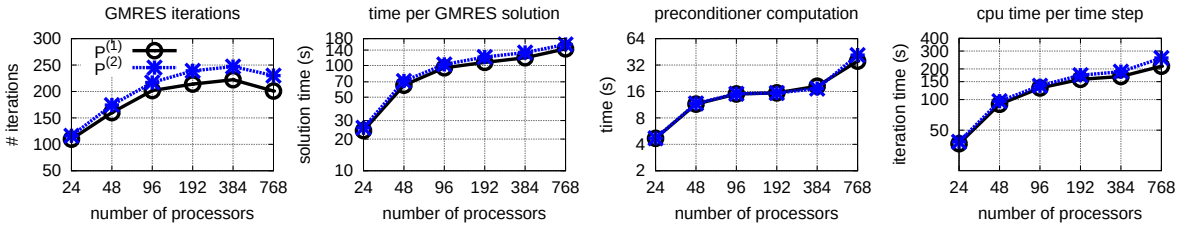


Figure 6.9: GCE — Weak scalability results for the number of processors ranging from 24 to 768.

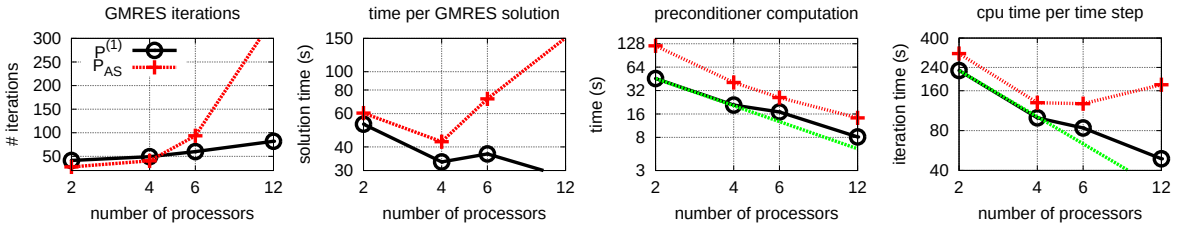


Figure 6.10: GCE — Strong scaling for the smallest geometry: even with few processors the number of iterations increases very fast for P_{AS}, which prevents scalability.

the behavior of the AAS preconditioner compared to P_{AS-DN1} on the smaller problem for a number of processors ranging from 2 to 12 in Figure 6.10. We observe that the number of iterations for the P_{AS}, though it is smaller when considering 2 cores, dramatically increases with the number of processors.

We can notice from the weak scalability plot of Figure 6.9 that the GMRES iterations remain bounded up to 768 processors. The iteration number comparison also shows that the performances of P_{AS-DN1} are slightly better than those of P_{AS-DN2}, however the same trend is observed in both cases. We notice also the difference of performance between the 24 processors case and the 48 processors one: the simulation does not scale in this step. This is probably due on one hand to the fact that 24 processors fit in two nodes of the cluster, and the communication between processors in the same node is faster. On the other hand on the 0.5 cm tube considered the pressure pulse imposed at the inlet exits the geometry during the first time steps, i.e, the characteristics of the simulation are different than in the other cases. We also remark that for the largest simulation the memory of one node is not sufficient when all the 12 processors are used. For this reason in that case we use only 6 cores per node. The loss of performance when reaching 768 processors is investigated more in detail in the next paragraph. In conclusion from the "CPU time per time step" plot we see that up to 384 processors the code scalability is acceptable. Here we did not represent the scalability of the assembling phase for the fluid block, of the solution of the harmonic extension equation and of the HDF5 post processing. However the behavior of these steps is accounted for in the "CPU time per time step" plot in Figure 6.9.

In Figure 6.11 we plot the results of three simulations obtained by varying the sizes of the local problems. We call N_i the ratio between the length of the geometry i among those in Figure 6.8(b) and the length of the shortest pipe (0.5 cm). These simulations have the following specifics:

1. $12 \cdot N_i$ processors for the geometry i, using 12 processors per node;

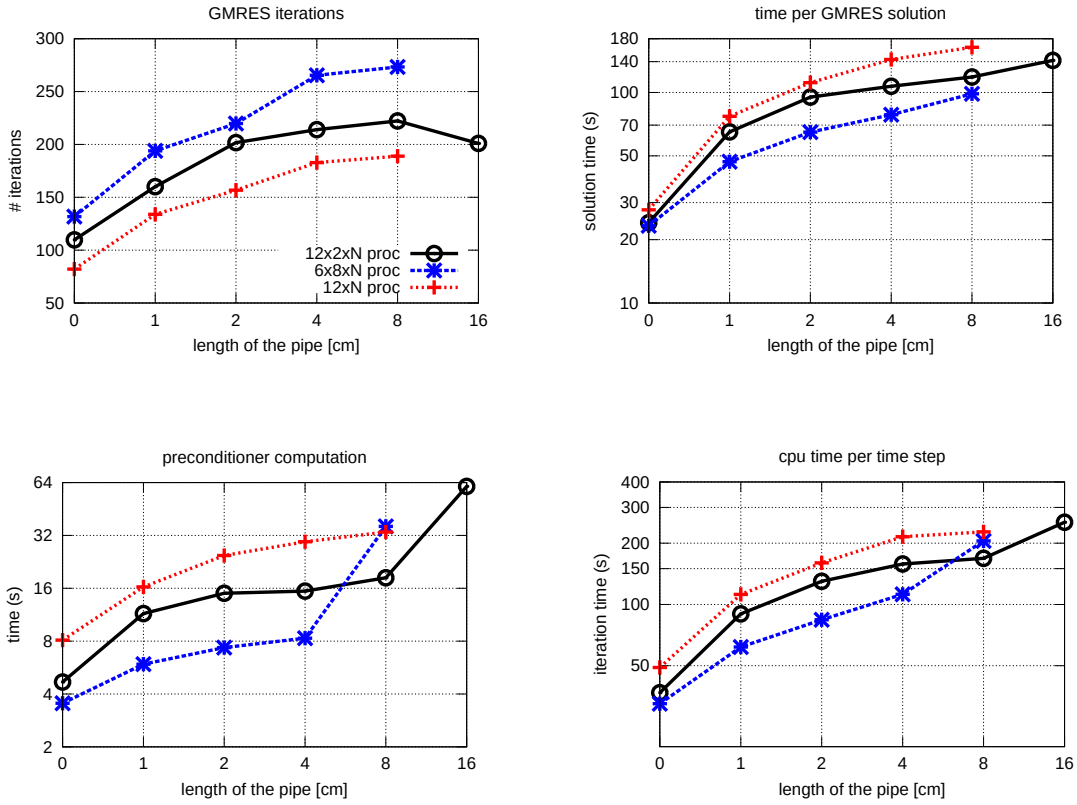


Figure 6.11: GCE — Weak scalability for different sizes of the local problems. N varies with the different geometries considered from 1 (corresponding to the shortest pipe) to 32 (corresponding to the longest one).

2. $24 \cdot N_i$ processors for the geometry i, also using 12 processors per node, except for the largest simulation, for which 6 processors per node are used due to memory issues, as already mentioned;

3. $48 \cdot N_i$ processors for the geometry i, with 6 processors per node (so that only one half of the processors in each node is used).

We remark from the results obtained that the GMRES iterations increase with the number of processors, as expected. The preconditioner computation has the same unexpected breakdown that has already been observed in Figure 6.9. This seems to be related to the number of processors employed: passing from 386 to 768 processors the preconditioner computation does not scale in any of the simulations performed, while it shows a nearly optimal strong scalability up to 386 processors. This introduces a bottleneck that still needs to be investigated. For the time being we conclude that on a Cray XT5 increasing the number of processors there is a gain in performance only below 768 processors. Further investigations are planned also on different platforms.

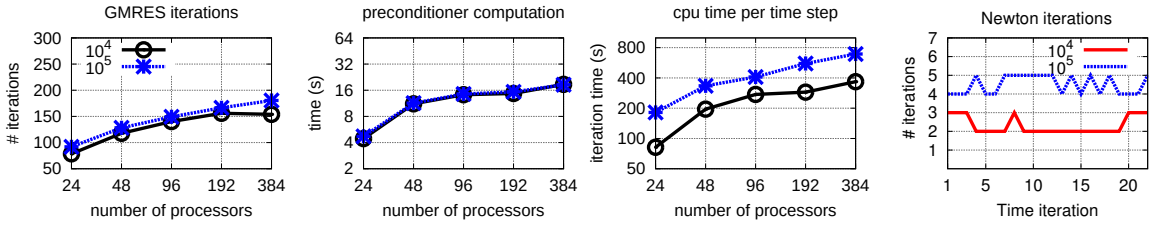


Figure 6.12: CE — We compare two simulations with different inlet pressure imposed. The first three plots represent the simulations run on the 0.5 cm to 8 cm tubes in figure 6.8(b), showing the GMRES iterations, preconditioner computation time, CPU time per time step. The last figure shows the number of Newton iterations for the 4 cm tube case.

Weak scalability for CE

We carry out a weak scalability test for the CE time discretization similar to the one for the GCE case. The tolerance for the linear system is set to 10^{-7}, while the relative tolerance for Newton is $\epsilon^r = 10^{-5}$. To estimate the influence of the geometric nonlinearity on the algorithm we compare two simulations with two different values for the inlet stress condition: $\sigma_f \cdot \mathbf{n} = 1.33 \cdot 10^4\,\mathrm{dyn/cm^2}$ and $\sigma_f \cdot \mathbf{n} = 1.33 \cdot 10^5\,\mathrm{dyn/cm^2}$ (for the first 3 ms, then zero). The time step chosen is $\delta t = 5 \cdot 10^{-4}$ s. With this choice the Newton scheme converges for all the test cases considered, while some convergence issue may appear when the initial state is far from the initial solution (i.e., when the timestep is large).

The P_{AS-QN} preconditioner for CE, which shows a good behavior on the strong scalability tests, in this case inherits the ill conditioning of the fluid-structure block, and a comparison between P_{AS-QN} and P_{AS-DN} produces a similar plot as the one shown in Figure 6.10. For this reason in the simulations presented here the preconditioner P_{AS-DN} (cf. Section §4.3.3) has been adopted.

In Figure 6.12 we compare the two different inlet stress conditions. The comparison is made on the first five benchmark geometries of Figure 6.8(b). The last geometry is not used because of the issue explained in the previous subsection, which affects the scalability from the 8 cm to the 16 cm pipe. The comparisons of Figure 6.12 are obtained using the maximum number of cores per nodes. The Newton iterations required to reach the specified tolerance increase slightly when the value of the inlet condition increases 10 times, which means that the method is robust. Furthermore we stress the fact that the computational time required to build the preconditioner is the same as for the GCE case, even though the matrix is larger. This thanks to the three factors splitting of the preconditioner.

Conclusion

In summary, we report the good behavior of the preconditioners obtained by composing a block Gauss–Seidel preconditioning strategy with a classical domain decomposition one. The factorization leading to such preconditioners is done at algebraic level, for both the GCE system and the CE (or all the other geometry–implicit variants) time discretizations. In particular for the CE system we show that a three factors splitting, which can be obtained by neglecting only an identity block in the system matrix (see (4.2.20)), leads to the best results in term of both time for the computation of the preconditioner and number of iterations for the linear system. We use a classical one level algebraic additive Schwarz preconditioner for

the local problems. This choice may affect the scalability of the preconditioner, while another choice (e.g. multilevel preconditioners, c.f. Section §4.3), also in view of Proposition 4.3.1, can improve further the spectral properties of the strategy proposed.

Conclusions

We report here an outline of this thesis, highlighting the original contributions, the concluding remarks and the limitations of the current approach.

This work is divided in three parts, containing two chapters each. The equations for the fluid and structure models are derived in the first chapter, with particular focus on the fluid equations in a moving domain and their ALE representation, which is the one adopted in this context.

An overview of the state of the art concerning the FSI models, coupling strategies, stability issues, time and space discretization, is provided in the second chapter. The description in this chapter is focalized on the (time and space) discretization methods used in the current work.

The original contributions of this thesis, detailed in the second and third parts, concern mainly the following four aspects:

- The nonlinear solver implementation, in particular the detailed description of the derivation and assembling of the shape derivatives terms for different formulations, with observations on the implementation efficiency.
- The study of suitable parallel preconditioners, their implementation in a general framework allowing to exploit already existing preconditioners which are specific for each field, the analysis of the conditioning of the system when these preconditioners are employed.
- The FSI simulation of physiological and pathological situations, with quantification of indicators of clinical interest, the comparisons on these physiological simulations between different space and time discretizations, different models and different algorithms for handling the nonlinearities.
- The numerical study of the parallel performance for some of the preconditioners introduced on high performance supercomputers.

Concerning the solution of the nonlinearity we cannot conclude that one method is better than the others in all possible situations. We found that for the FSI simulations in the aortic arch both the solution of the geometrical nonlinearity and of the convective term are key points in order to obtain accurate results, in fact in proximity of the systolic peak, where the all the nonlinearities are important and the deformations are large, the the fully–implicit (FI) scheme produces significantly different results than the other cases for time steps larger than 1 ms. We also remark that the computational cost of all the geometry–implicit variants is approximately the same, thus the most efficient alternatives are, according to our tests, the geometry–convective explicit scheme for a fast stable solution that can be inaccurate for large time steps or when the nonlinearity is large, and the fully implicit one.

We compared also an exact and an inexact variants of the Newton algorithm, finding that the extra cost of considering the complete Jacobian can be negligible (if a proper implementation of the shape derivatives assembling and a proper preconditioner for the Jacobian system are employed) and scalable when increasing the number of processors. On the other hand in some cases (e.g. during systole in the aortic arch) we found the convergence for the inexact Newton scheme to be considerably slower than for exact Newton. Our conclusion is that using the exact Jacobian matrix is the best choice, provided that a method to obtain a close enough initial state is available, in order for the Newton method to converge.

Concerning the parallel preconditioners for FSI we can conclude that the strategy introduced, featuring the splitting into factors and the separation of the different fields, greatly improves a more naive approach which does not account for the block structure of the system. In particular we stress that the cost of computing one of the preconditioners proposed for the FI system (P_{AS-DN}) is the same as for computing a preconditioner of the fluid block alone, and its parallel behaviour (also in view of Proposition 4.3.1) only depends on how well the fluid and structure blocks are approximated, while the coupling does not influence substantially the condition number of the preconditioned system.

Perspectives and Future Work

The present study has established a framework that can be used for further investigations, and the development of a robust and general parallel FSI solver opens the way for more complicated couplings, models, and for larger and more detailed geometries. In particular the code is mature enough to move towards a geometrical multiscale framework, which would allow the simulation of the arterial circulation for the entire body, focusing (i.e., simulating the 3D FSI) on the critical zones and using reduced models for the parts of the circulatory system which are not object of interest. Another necessary improvement to the model is the extension to nonlinear structures. This topic, as the previous one, is the subject of ongoing work, and would allow eventually to simulate with better accuracy the cases in which the nonlinearity of the structure, due to e.g. the activation of the collagen fibers, plays an important role (like in the process of aneurysm growth). The nonlinearity of the structure is also important for the mechanics of the heart (that features an anisotropic and nonlinear constitutive relation). In this regard the coupling of the FSI solver described in this thesis and an electro–mechanical model for the heart which is under development within the same library (LifeV) is one of the end goals. Eventually we showed that improving the preconditioner quality returns in improving the preconditioners for the single uncoupled problems. Thus the research (already ongoing in LifeV) in the field of parallel preconditioners for the Navier–Stokes equations can be easily plugged into the FSI framework, allowing eventually to efficiently move to massively parallel high performance platforms.

Bibliography

[ACCS06] C. Arthur Chang, D. Coutand, and S. Shkoller, *Navier-Stokes equations interacting with a nonlinear elastic fluid shell*, http://arxiv.org/abs/math.AP/0611670, 2006.

[ACF09] M. Astorino, F. Chouly, and M. A. Fernández, *Robin based semi-implicit coupling in fluid-structure interaction: Stability analysis and numerics*, SIAM J. Sci. Comput. **31** (2009), no. 6, 4041–4065.

[AK10] O. Axelsson and J. Karátson, *Condition number analysis for various form of block matrix preconditioners*, Electr. Trans. Numer. Anal. **36** (2010), 168–194.

[Ant02] L. Antiga, *Patient-specific modeling of geometry and blood flow in large arteries*, Ph.D. thesis, Politecnico di Milano, 2002, p. 190.

[AV89] O. Axelsson and P. S. Vassilevski, *Algebraic multilevel preconditioning methods, i*, Numer. Math. **56** (1989), 157–177.

[Axe94] O. Axelsson, *Iterative solution methods*, Cambridge University Press, New York, 1994.

[BA76] I. Babuska and A. Aziz, *On the angle condition in the finite element method*, SIAM J. Num. Anal. **13** (1976), 214–226.

[BA09] A. L. Braun and A. M. Awruch, *A partitioned model for fluid-structure interaction problems using hexahedral finite elements with one-point quadrature*, Int. J. Numer. Meth. Engng. **79** (2009), 505–549.

[Bal06] D. Balzani, *Polyconvex anisotropic energies and modelling of damage applied to arterial wall*, Ph.D. thesis, Universität Duisburg-Essen, Essen, 2006.

[BC09] A. T. Barker and X.-C. Cai, *NKS for fully coupled fluid-structure interaction with application*, Lect. Notes Comput. Sci. Eng. **70** (2009), 275–282.

[BC10a] ———, *Scalable parallel methods for monolithic coupling in fluid-structure interaction with application to blood flow modeling*, J. Comput. Phys. **229** (2010), no. 3, 642–659.

[BC10b] ———, *Two-level newton and hybrid schwarz preconditioners for fluid-structure interaction*, SIAM J. Sci. Comput. **32** (2010), no. 4, 2395–2417.

[BCA+10] C. Bekas, A. Curioni, P. Arbenz, C. Flaig, G. H. Van Lenthe, R. Müller, and A. J. Wirth, *Extreme scalability challenges in micro-finite element simulations of human bone*, Concurr. Comput. : Pract. Exper. **22** (2010), 2282–2296.

[BCHZ08] Y. Bazilevs, V. M. Calo, T. J. R. Hughes, and Y. Zhang, *Isogeometric fluid-structure interaction: theory, algorithms, and computations*, Comput. Mech. **43** (2008), no. 1, 3–37.

[BCT02] E. Bechet, J.-C. Cuilliere, and F. Trochu, *Generation of a finite element mesh from stereolithography (stl) files*, Computer-Aided Design **34** (2002), no. 1, 1–17.

[BdV04] H. Beirão da Veiga, *On the existence of strong solutions to a coupled fluid-structure evolution problem*, J. Math. Fluid. Mech. **6** (2004), 21–52.

[BF07] E. Burman and M. Fernndez, *Continuous interior penalty finite element method for the time-dependent navierstokes equations: space discretization and convergence*, Numerische Mathematik **107** (2007), 39–77, 10.1007/s00211-007-0070-5.

[BF09] E. Burman and M. A. Fernández, *Stabilization of explicit coupling in fluid-structure interaction involving fluid incompressibility*, Comput. Methods Appl. Mech. Engrg. **198** (2009), no. 5-8, 766–784.

[BFH04] E. Burman, M. . Fernndez, and P. Hansbo, *Edge stabilization for the incompressible Navier-Stokes equations: a continuous interior penalty method*, Tech. report, 2004, EPFL-IACS report 23.2004.

[BFOG97] M. Batdorf, L. Freitag, and C. Ollivier-Gooch, *A computational study of the effect of unstructured mesh quality on solution efficiency*, Proc. 13th AIAA Computational Fluid Dynamics Conf., 1997.

[BG04] D. Boffi and L. Gastaldi, *Stability and geometric conservation laws for ale formulations*, Comput. Methods Appl. Mech. Engrg. **193** (2004), no. 42-44, 4717–4739.

[BGH+09] Y. Bazilevs, J. Gohean, T. Hughes, R. Moser, and Y. Zhang, *Patient-specific isogeometric fluid-structure interaction analysis of thoracic aortic blood flow due to implantation of the Jarvik 2000 left ventricular assist device*, Comput. Methods Appl. Mech. Engrg. **198** (2009), no. 45-46, 3534–3550.

[BGL05] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta Numerica **14** (2005), 1–137.

[BLG00] H. Borouchaki, P. Laug, and P. George, *Parametric surface meshing using a combined advancing-front generalized delaunay approach*, Int. J. Numer. Meth. Engng. **49** (2000), 223–259.

[BLS10] A. Barker, C. Lanning, and R. Shandas, *Quantification of hemodynamic wall shear stress in patients with bicuspid aortic valve using phase-contrast mri*, Ann. Biomed. Eng. **38** (2010), 788–800.

[BNV08] S. Badia, F. Nobile, and C. Vergara, *Fluid-structure partitioned procedures based on Robin transmission conditions*, J. Comput. Phys. **227** (2008), no. 14, 7027–7051.

[Bon09] J. Bonnemain, *From medical images to numerical simulations*, Tech. report, MATHICSE, 2009.

[BQQ08a] S. Badia, A. Quaini, and A. Quarteroni, *Modular vs. non-modular preconditioners for fluid-structure systems with large added-mass effect*, Comput. Methods Appl. Mech. Engrg. **197** (2008), no. 49-50, 4216–4232.

[BQQ08b] ———, *Splitting methods based on algebraic factorization for fluid-structure interaction*, SIAM J. Sci. Comput. **30** (2008), no. 4, 1778–1805.

[BRvdV09] B. Beulen, M. Rutten, and F. van de Vosse, *A time-periodic approach for fluid-structure interaction in distensible vessels*, J. Fluids Struct. **25** (2009), no. 5, 954–966.

[BT96] A. Burleson and V. Turitto, *Identification of quantifiable hemodynamic factors in the assessment of cerebral aneurysm behavior*, Thromb. Haemost. **76** (1996), 118–123.

[CBD+07] U. Catalyurek, E. Boman, K. Devine, D. Bozdag, R. Heaphy, and L. Riesen, *Hypergraph-based dynamic load balancing for adaptive scientific computations*, Proc. of 21st International Parallel and Distributed Processing Symposium (IPDPS'07), IEEE, 2007.

[CDEG05] A. Chambolle, B. Desjardins, M. J. Esteban, and C. Grandmont, *Existence of weak solutions for the unsteady interaction of a viscous fluid with an elastic plate*, Journal of Mathematical Fluid Mechanics **7** (2005), 368–404.

[CDFQ11] P. Crosetto, S. Deparis, G. Fourestey, and A. Quarteroni, *Parallel Algorithms for Fluid-Structure Interaction Problems in Haemodynamics*, SIAM J. Sci. Comput. (2011), accepted.

[CGN05] P. Causin, J.-F. Gerbeau, and F. Nobile, *Added-mass effect in the design of partitioned algorithms for fluid-structure problems*, Comput. Methods Appl. Mech. Engrg. **194** (2005), no. 42-44, 4506 – 4527.

[CH93] J. Chung and G. M. Hulbert, *A time integration algorithm for structural dynamics with improved numerical dissipation: The generalized-α method*, Journal of Applied Mechanics **60** (1993), no. 2, 371–375.

[CK02] X.-C. Cai and D. E. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput. **24** (2002), no. 1, 183–200 (electronic).

[CMM08] G.-H. Cottet, E. Maitre, and T. Milcent, *Eulerian formulation and level set models for incompressible fluid-structure interaction*, M2AN Math. Model. Numer. Anal. **42** (2008), no. 3, 471–492.

[CP08] C. Chevalier and F. Pellegrini, *Pt-scotch: A tool for efficient parallel graph ordering*, Parallel Comput. **34** (2008), 318–331.

[CRD+11] P. Crosetto, P. Reymond, S. Deparis, D. Kontaxakis, N. Stergiopulos, and A. Quarteroni, *Fluid-structure interaction simulation of aortic blood flow*, Computers & Fluids **43** (2011), no. 1, 46–57, Symposium on High Accuracy Flow Simulations. Special Issue Dedicated to Prof. Michel Deville.

[Cur04] A. Curnier, *Mécanique des solides déformables vol.1 cinématique, dynamique, énergétique*, Presses polytechniques et universitaires romandes, Lausanne, 2004.

[CZ10] Y. Cheng and H. Zhang, *Immersed boundary method and lattice boltzmann method coupled fsi simulation of mitral leaflet flow*, Computers & Fluids **39** (2010), no. 5, 871–881.

[Dav06] T. A. Davis, *Direct methods for sparse linear systems*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2006.

[DAV10] J. Degroote, S. Annerel, and J. Vierendeels, *A multi-solver quasi-newton method for the partitioned simulation of fluid-structure interaction*, IOP Conference Series: Materials Science and Engineering **10** (2010), no. 1, 12–20.

[DBHV08] J. Degroote, P. Bruggeman, R. Haelterman, and J. Vierendeels, *Stability of a coupling technique for partitioned solvers in fsi applications*, Computers & Structures **86** (2008), no. 23-24, 2224–2234.

[DBV09] 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 **87** (2009), no. 11-12, 793 – 801, Fifth MIT Conference on Computational Fluid and Solid Mechanics.

[DDFQ06] S. Deparis, M. Discacciati, G. Fourestey, and A. Quarteroni, *Fluid-structure algorithms based on Steklov-Poincaré operators*, Comput. Methods Appl. Mech. Engrg. **195** (2006), no. 41-43, 5797–5812.

[DDFQ07] ———, *Heterogeneous domain decomposition methods for fluid-structure interaction problems*, Domain decomposition methods in science and engineering XVI, Lect. Notes Comput. Sci. Eng., vol. 55, Springer, Berlin, 2007, pp. 41–52.

[DDQ06] S. Deparis, M. Discacciati, and A. Quarteroni, *A domain decomposition framework for fluid-structure interaction problems*, Proceedings of the Third International Conference on Computational Fluid Dynamics (ICCFD3), 2006, pp. 41–58.

[Dep04] S. Deparis, *Numerical analysis of axisymmetric flows and methods for fluid-structure interaction arising in blood flow simulation*, Ph.D. thesis, École Polytechnique Fédérale de Lausanne, Lausanne, 2004.

[DFF03] S. Deparis, M. A. Fernández, and L. Formaggia, *Acceleration of a fixed point algorithm for fluid-structure interaction using transpiration conditions*, M2AN Math. Model. Numer. Anal. **37** (2003), no. 4, 601–616.

[DGH82] J. Donea, S. Giuliani, and J. Halleux, *An arbitrary lagrangian-eulerian finite element method for transient dynamic fluid-structure interactions*, Comput. Methods Appl. Mech. Engrg. **33** (1982), no. 1-3, 689–723.

[DP07] W. G. Dettmer and D. Perić, *A fully implicit computational strategy for strongly coupled fluid-solid interaction*, Arch. Comput. Methods Eng. **14** (2007), no. 3, 205–247.

[EHS+06] H. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro, *Block preconditioners based on approximate commutators*, SIAM J. Sci. Comput **27** (2006), 1651–1668.

[EHS+08] H. Elman, V. E. Howle, J. Shadid, R. Shuttleworth, and R. Tuminaro, *A taxonomy and comparison of parallel block multi-level preconditioners for the incompressible navier-stokes equations*, J. Comput. Phys. **227** (2008), 1790–1808.

[EHST03] H. Elman, V. E. Howle, J. Shadid, and R. Tuminaro, *A parallel block multi-level preconditioner for the 3d incompressible navier-stokes equations*, J. Comput. Phys **187** (2003), 504–523.

[ESW05] H. Elman, D. Silvester, and A. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, ch. 8, Oxford University Press, Oxford, 2005.

[FB71] F. Fontan and E. Baudet, *Surgical repair of tricuspid atresia*, Thorax **26** (1971), no. 3, 240–248.

[FFT00] T. Fanion, M. Fernández, and P. L. Tallec, *Deriving adequate formulations for fluid structure interaction problems: from ALE to transpiration*, Rév. Européenne Élém. Finis **9** (2000), no. 6-7, 681–708.

[FGG01] C. Farhat, P. Geuzaine, and C. Grandmont, *The discrete geometric conservation law and the nonlinear stability of ale schemes for the solution of flow problems on moving grids*, J. Comput. Phys. **174** (2001), no. 2, 669–694.

[FGG07] M. A. Fernández, J.-F. Gerbeau, and C. Grandmont, *A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid*, Internat. J. Numer. Methods Engrg. **69** (2007), no. 4, 794–821.

[FGNQ01] L. Formaggia, J. F. Gerbeau, F. Nobile, and A. Quarteroni, *On the coupling of 3d and 1d navier-stokes equations for flow problems in compliant vessels*, Comput. Methods Appl. Mech. Engrg. **191** (2001), no. 6-7, 561–582.

[FGNQ02a] L. Formaggia, J.-F. Gerbeau, F. Nobile, and A. Quarteroni, *Numerical treatment of defective boundary conditions for the Navier–Stokes equations.*, SIAM J. Numer. Anal. **40** (2002), no. 1, 376–401 (English).

[FGNQ02b] L. Formaggia, J. Gerbeau, F. Nobile, and A. Quarteroni, *Numerical treatment of defective boundary conditions for the navier-stokes equations*, SIAM J. Numer. Anal. **40** (2002), 376–401.

[FL00] C. Farhat and M. Lesoinne, *Two efficient staggered algorithms for the serial and parallel solution of three-dimensional nonlinear transient aeroelastic problems*, Comput. Methods Appl. Mech. Engrg. **182** (2000), 499–515.

[Fla89] H. Flanders, *Differential forms with applications to the physical sciences*, Dover books on mathematics, Dover Publications, 1989.

[FLT98] C. Farhat, M. Lesoinne, and P. L. 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*, Comput. Methods Appl. Mech. Engrg. **157** (1998), no. 1-2, 95–114.

[FM05] M. Fernández and M. Moubachir, *A Newton method using exact jacobians for solving fluid-structure coupling*, Computers and Structures **83** (2005), no. 2-3, 127–142.

[FQV09] L. Formaggia, A. Quarteroni, and A. Veneziani (eds.), *Cardiovascular mathematics: Modeling and simulation of the circulatory system*, Modeling, Simulation and Applications, vol. 1, Springer, Milan, 2009.

[Fri72] I. Fried, *Condition of finite element matrices generated from nonuniform meshes*, AIAA Journal **10** (1972), 219–221.

[Fun97] Y. C. Fung, *Biomechanics : circulation*, 2nd ed ed., Springer, New York, 1997.

[FVCJ+06] C. A. Figueroa, I. E. Vignon-Clementel, K. E. Jansen, T. J. Hughes, and C. A. Taylor, *A coupled momentum method for modeling blood flow in three-dimensional deformable arteries*, Comput. Methods Appl. Mech. Engrg. **195** (2006), no. 41-43, 5685–5706, John H. Argyris Memorial Issue. Part II.

[FvdZG06] C. Farhat, K. G. van der Zee, and P. Geuzaine, *Provably second-order time-accurate loosely-coupled solution algorithms for transient nonlinear computational aeroelasticity*, Comput. Methods Appl. Mech. Engrg. **195** (2006), no. 17-18, 1973–2001.

[FVV08] L. Formaggia, A. Veneziani, and C. Vergara, *A new approach to numerical solution of defective boundary value problems in incompressible fluid dynamics*, SIAM J. Sci. Comput. **46** (2008), no. 6, 2769–2794.

[FWR07] C. Förster, W. A. Wall, and E. Ramm, *Artificial added mass instabilities in sequential staggered coupling of nonlinear structures and incompressible viscous flows*, Comput. Methods Appl. Mech. Engrg. **196** (2007), no. 7, 1278–1293.

[GAS06] A. Gray, E. Abbena, and S. Salamon, *Modern differential geometry of curves and surfaces with mathematica, third edition (studies in advanced mathematics)*, Chapman & Hall/CRC, 2006.

[GCF94] Y. Gobin, J. Counord, and P. Flaud, *In vitro study of haemodynamics in a giant saccular aneurysm model: influence of flow dynamics in the parent vessel and effects of coil embolisation.*, J. Neuroradiology **36** (1994), 530–536.

[GGCC09] G. Guidoboni, R. Glowinski, N. Cavallini, and S. Canic, *Stable loosely-coupled-type algorithm for fluid-structure interaction in blood flow*, J. Comput. Phys. **228** (2009), no. 18, 6916–6937.

[GGNV10] L. Gerardo-Giorda, F. Nobile, and C. Vergara, *Analysis and optimization of robin–robin partitioned procedures in fluid-structure interaction problems*, SIAM J. Sci. Comput. **48** (2010), no. 6, 2091–2116.

[GK10] L. Grinberg and G. E. Karniadakis, *Extrapolation–based acceleration of iterative solvers: Application to simulation of 3d flows*, Commun. Comput. Phys. **9** (2010), no. 3, 607–626.

[GKW10] M. W. Gee, U. Küttler, and W. A. Wall, *Truly monolithic algebraic multigrid for fluid–structure interaction*, Int. J. Numer. Meth. Engng. **85** (2010), 987–1016.

[GN93] A. Green and P. Naghdi, *A direct theory of viscous fluid flow in pipes. I. Basic general developments*, Phil. Trans. R. Soc. Lond., Ser. A **342** (1993), 525–542.

[GNS93] A. Green, P. Naghdi, and M. Stallard, *A direct theory of viscous fluid flow in pipes. II. Flow of incompressible viscous fluid in curved pipes*, Phil. Trans. R. Soc. Lond., Ser. A **342** (1993), 543–572.

[GSP92] V. Graves, C. Strother, and R. A. Partington, CR, *Flow dynamics of lateral carotid artery aneurysms and their effects on coils and balloons: an experimental study in dogs.*, AJNR Am J Neuroradiol **13** (1992), 189–196.

[GV03] J. Gerbeau and M. Vidrascu, *A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows*, M2AN Math. Model. Numer. Anal. **37** (2003), no. 4, 663–680.

[GWM06] F. Gao, M. Watanabe, and T. Matsuzawa, *Fluid-structure interaction within a layered aortic arch model.*, J. Biol. Phys. **32** (2006), no. 5, 435–454.

[GZG93] D. Giddens, C. Zarins, and S. Glagov, *The role of fluid mechanics in localisation and detection of atherosclerosis.*, J Biomech Engng **115** (1993), 588–594.

[Han05] P. Hansbo, *Nitsche's method for interface problems in computational mechanics*, GAMM-Mitt. (2005), no. 28/2, 183–206.

[Hei04] M. Heil, *An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems*, Comput. Methods Appl. Mech. Engrg. **193** (2004), no. 1-2, 1–23.

[HGO00] G. Holzapfel, T. Gasser, and R. Ogden, *A new constitutive framework for arterial wall mechanics and a comparative study of material models*, Journal of Elasticity (2000), 1–48.

[HHB08] M. Heil, A. L. Hazel, and J. Boyle, *Solvers for large-displacement fluid-structure interaction problems: segregated versus monolithic approaches*, Computational Mechanics **43** (2008), no. 1, 91–101.

[HL73] T. J. R. Hughes and J. Lubliner, *On the one-dimensional theory of blood flow in the larger vessels*, Math. Biosci. **18** (1973), 161–170.

[HO06] G. A. Holzapfel and R. W. Ogden, *Mechanics of biological tissue*, Springer Verlag, 2006.

[Hro01] J. Hron, *Fluid structure interaction with applications in biomechanics*, Ph.D. thesis, Charles University, Prague, 2001.

[HSSB02] G. A. Holzapfel, M. Stadler, and C. A. J. Schulze-Bauer, *A layer-specific three-dimensional model for the simulation of balloon angioplasty using magnetic resonance imaging and mechanical testing*, Ann. Biomed. Eng. **30** (2002), no. 6, 753–767.

[HT06] J. Hron and S. Turek, *A monolithic fem/multigrid solver for an ale formulation of fluid-structure interaction with applications in biomechanics*, Fluid-Structure Interaction (H.-J. Bungartz and M. Sch afer, eds.), Lecture Notes in Computational Science and Engineering, vol. 53, Springer Berlin Heidelberg, 2006, pp. 146–170.

[Hug74] T. J. R. Hughes, *A study of the one-dimensional theory of arterial pulse propagation*, Report 74-13, University of California Berkeley, Structural Engineering Laboratory, December 1974.

[HWD04] B. Hübner, E. Walhorn, and D. Dinkler, *A monolithic approach to fluid-structure interaction using space-time finite elements*, Comput. Methods Appl. Mech. Engrg. **193** (2004), no. 23-26, 2087 – 2104.

[IBM08] IBM journal of Research and Development staff, *Overview of the ibm blue gene/p project*, IBM J. Res. Dev. **52** (2008), 199–220.

[IN02] Y. Ito and K. Nakahashi, *Direct surface triangulation using stereolithography data*, AIAA Journal **40** (2002), no. 3, 490–496.

[IOP03] S. R. Idelsohn, E. Oate, and F. D. Pin, *A lagrangian meshless finite element method applied to fluid-structure interaction problems*, Computers & Structures **81** (2003), no. 8-11, 655–671, K.J Bathe 60th Anniversary Issue.

[JGLJ11] R. Jaiman, P. Geubelle, E. Loth, and X. Jiao, *Combined interface boundary condition method for unsteady fluid-structure interaction*, Comput. Methods Appl. Mech. Engrg. **200** (2011), no. 1-4, 27–39.

[JWH00] K. E. Jansen, C. H. Whiting, and G. M. Hulbert, *A generalized-α method for integrating the filtered navier-stokes equations with a stabilized finite element method*, Comput. Methods Appl. Mech. Engrg. **190** (2000), no. 3-4, 305–319.

[Key98] D. E. Keyes, *How scalable is domain decomposition in practice?*, Proceedings of the 11th Intl. Conf. on Domain Decomposition Methods (1998), 286–297.

[KGF+09] U. Küttler, M. Gee, C. Förster, A. Comerford, and W. A. Wall, *Coupling strategies for biomedical fluid–structure interaction problems*, Commun. Numer. Meth. Engng. **26** (2009), 305–321.

[Kla98] A. Klawonn, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comput. **19** (1998), no. 1, 172–184.

[KM09] J. Kraus and S. Margenov, *Robust algebraic multilevel methods and algorithms*, Walter de Gruyter, 2009.

[Kor90] J. Korsgaard, *On the representation of symmetric tensor-valued isotropic functions*, International Journal of Engineering Science **28** (1990), no. 12, 1331–1346.

[KR10] A. Klawonn and O. Rheinbach, *Highly scalable parallel domain decomposition methods with an application to biomechanics*, Z. angew. Math. Mech. **90** (2010), 5–32.

[Krz10] P. Krzyżanowski, *On block preconditioners for saddle point problems with singular or indefinite (1, 1) block*, Numer. Linear Algebra Appl. (2010), 1099–1506.

[KVCF+09] H. Kim, I. Vignon-Clementel, C. Figueroa, J. LaDisa, K. Jansen, J. Feinstein, and C. Taylor, *On coupling a lumped parameter heart model and a three-dimensional finite element aorta model*, Ann Biomed Eng. (2009), 2153–69.

[KW08a] U. Küttler and W. A. Wall, *The dilemma of domain decomposition approaches in fluid-structure interactions with fully enclosed incompressible fluids*, Domain decomposition methods in science and engineering XVII, Lect. Notes Comput. Sci. Eng., vol. 60, Springer, Berlin, 2008, pp. 575–582.

[KW08b] U. Küttler and W. A. Wall, *Fixed-point fluid-structure interaction solvers with dynamic relaxation*, Computational Mechanics **43** (2008), no. 1, 61–72.

[KW09] ———, *Vector extrapolation for strong coupling fluid-structure interaction solvers*, Journal of Applied Mechanics **76** (2009), no. 2.

[KWO+97] R. Krams, J. Wentzel, J. Oomen, R. Vinke, J. Schuurbiers, P. de Feyter, P. Serruys, and C. Slager, *Evaluation of endothelial shear stress and 3D geometry as factors determining the development of atherosclerosis and remodeling in human coronary arteries in vivo. combining 3D reconstruction from angiography and IVUS (ANGUS) with computational fluid dynamics.*, Arterioscl., Thromb. Vascul. Biol. **17** (1997), no. 10, 2061–2065.

[Kwo08] Y. W. Kwon, *Coupling of lattice boltzmann and finite element methods for fluid-structure interaction application*, Journal of Pressure Vessel Technology **130** (2008), no. 1.

[Lan82] G. J. Langewouters, *Visco-elasticity of the human aorta in vitro in relation to pressure and age*, Ph.D. thesis, Free University, Amsterdam, 1982.

[LB03] P. Laug and H. Boruchaki, *Interpolating and meshing 3d surface grids*, Int. J. Numer. Meth. Engng. **58** (2003), 209–225.

[LDG+08] Y. Liu, C. Dang, M. Garcia, H. Gregersen, and G. S. Kassab, *Surrounding tissues affect vessel mechanics*, AJP - Heart and Circulatory Physiology (2008), no. 294, 514–523.

[Lio69] J.-L. Lions, *Quelques mèthodes de rèsolution des problèmes aux limites non linèaires*, Dunod, Paris, 1969.

[LM01] P. Le Tallec and J. Mouro, *Fluid structure interaction with large structural displacements*, Comput. Methods Appl. Mech. Engrg **190** (2001), 3039–3067.

[LPRM02] B. Levy, S. Petitjean, N. Ray, and J. Maillot, *Least squares conformal maps for automatic texture atlas generation*, Computer Graphics (Proceedings of SIGGRAPH 02), 2002, pp. 362 – 371.

[LSS+09] P. T. Lin, J. N. Shadid, M. Sala, R. S. Tuminaro, G. L. Hennigan, and R. J. Hoekstra, *Performance of a parallel algebraic multilevel preconditioner for stabilized finite element semiconductor device modeling*, J. Comput. Phys. **228** (2009), no. 17, 6250–6267.

[MBDQ11] A. C. I. Malossi, P. J. Blanco, S. Deparis, and A. Quarteroni, *Algorithms for the partitioned solution of weakly coupled fluid models for cardiovascular flows*, Accepted in Int. J. Numer. Method. Biomed. Eng., 2011.

[MCdWV+10] E. Marchandise, C. Carton de Wiart, W. Vos, C. Geuzaine, and J.-F. Remacle, *High quality surface remeshing using harmonic maps. Part II: Surfaces with high genus and of large aspect ratio*, Int. J. Numer. Meth. Engng. (2010), accepted.

[MCG+11] E. Marchandise, P. Crosetto, C. Geuzaine, J.-F. Remacle, and E. Sauvage, *Quality open source mesh generation for cardiovascular flow simulations*, Modelling Physiological Flow (D. Ambrosi, A. Quarteroni, and G. Rozza, eds.), Springer Series on Modeling, Simulation and Applications, Springer-Verlag, 2011, accepted.

[MCW+10] E. Marchandise, G. Compère, M. Willemet, G. Bricteux, C. Geuzaine, and J.-F. Remacle, *Quality meshing based on stl triangulations for biomedical simulations*, Int. J. Numer. Meth. Engng. **83** (2010), 876–889.

[MN99] J. R. Magnus and H. Neudecker, *Matrix differential calculus with applications in statistics and econometrics*, 2nd ed., John Wiley & Sons, 1999.

[MNS06] H. G. Matthies, R. Niekamp, and J. Steindorf, *Algorithms for strong coupling procedures*, Comput. Methods Appl. Mech. Engrg. **195** (2006), no. 17-18, 2028–2049.

[MPGW10] U. Mayer, A. Popp, A. Gerstenberger, and W. Wall, *3d fluidstructure-contact interaction based on a combined xfem fsi and dual mortar contact approach*, Computational Mechanics **46** (2010), 53–67, 10.1007/s00466-010-0486-0.

[MS03] H. G. Matthies and J. Steindorf, *Partitioned strong coupling algorithms for fluid-structure interaction*, Computers & Structures **81** (2003), no. 8-11, 805–812, K.J Bathe 60th Anniversary Issue.

[MW01] D. P. Mok and W. A. Wall, *Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures*, Trends in Computational Structural Mechanics (K. Schweizerhof and W. Wall, eds.), K.U. Bletzinger, CIMNE, Barcelona, 2001.

[MWR01] D. P. Mok, W. A. Wall, and E. Ramm, *Accelerated iterative substructuring schemes for instationary fluid-structure interaction*, Computational Fluid and Solid Mechanics (K. Bathe, ed.), Elsevier, 2001, pp. 1325–1328.

[MXA+11] P. Moireau, N. Xiao, M. Astorino, C. A. Figueroa, D. Chapelle, C. A. Taylor, and J. F. Gerbeau, *External tissue support and fluid-structure simulation in blood flows.*, to appear in Biomechanics and Modeling in Mechanobiology (2011).

[NO05] W. Nichols and M. O'Rourke, *Mcdonald's blood flow in arteries: Theoretical, experimental and clinical principles*, 5th ed., Hodder Arnold, London, 2005.

[Nob01] F. Nobile, *Numerical Approximation of Fluid-Structure Interaction Problems with Application to Haemodynamics*, Ph.D. thesis, École Polytechnique Fédérale de Lausanne, 2001.

[NOHM98] W. W. Nichols, M. F. O'Rourke, C. Hartley, and D. A. McDonald, *Mcdonald's blood flow in arteries : theoretical, experimental, and clinical principles*, 4th ed., Arnold ; Oxford University Press, New York, 1998.

[Not10] Y. Notay, *An aggregation-based algebraic multigrid method*, etna **37** (2010), 123–146.

[NSP99] N. Noori, R. Scherer, and K. Perktold, *lood flow in distal end-to-side anastomoses with ptfe and a venous patch: results of an in vitro flow visualisation study.*, Eur J Vasc EndoVasc Surg **18** (1999), 191–200.

[NV08] F. Nobile and C. Vergara, *An effective fluid-structure interaction formulation for vascular dynamics by generalized Robin conditions*, SIAM J. Sci. Comput. **30** (2008), no. 2, 731–763.

[OnO11] E. Oñate and R. Owen, *Particle-based methods: Fundamentals and applications*, Computational Methods in Applied Sciences, vol. 25, Springer, 2011.

[OOL04] J. T. Ottesen, M. S. Olufsen, and J. K. Larsen, *Applied mathematical models in human physiology*, Society for Industrial and Applied Mathematics, Philadephia, PA, 2004.

[PB05] N. Parolini and E. Burman, *Subgrid edge stabilization for transport equations*, Tech. report, 2005, EPFL-IACS report 09.2005.

[PP10] G. Pena and C. Prud'hom, *Construction of a high order fluid-structure interaction solver*, J. Comput. Appl. Math. **234** (2010), 2358–2365.

[PPQ10] G. Pena, C. Prud'homme, and A. Quarteroni, *High order methods for the approximation of the incompressible navier-stokes equations in a moving domain*, Tech. report, 2010, MOX report 25.

[PR95] K. Perktold and G. Rappitsch, *Computer simulation of local blood flow and vessel mechanics in a compliant carotid artery bifurcation model*, J. Biomech. **28** (1995), no. 7, 845 – 856.

[PS09] M. Pennacchio and V. Simoncini, *Algebraic multigrid preconditioners for the bidomain reaction–diffusion system*, Appl. Numer. Math. **59** (2009), 3033–3050.

[PW11] L. F. Pavarino and O. B. Widlund, *BDDC and FETI-DP preconditioners for spectral element discretizations of almost incompressible elasticity*, Spectral and High Order Methods for Partial Differential Equations (J. S. Hesthaven and E. M. Ronquist, eds.), Lecture Notes in Computational Science and Engineering, vol. 76, Springer Berlin Heidelberg, 2011, pp. 479–486.

[QQ07] A. Quaini and A. Quarteroni, *A semi-implicit approach for fluid-structure interaction based on an algebraic fractional step method*, Math. Models Methods Appl. Sci. **17** (2007), no. 6, 957–983.

[QSS00] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical mathematics*, Texts in Applied Mathematics, vol. 37, Springer-Verlag, New York, 2000.

[Qua09] A. Quaini, *Algorithms for fluid-structure interaction problems arising in hemodynamics*, Ph.D. thesis, EPFL, Lausanne, 2009.

[QV94] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*, Springer Series in Computational Mathematics, vol. 23, Springer-Verlag, Berlin, 1994. MR 95i:65005

[QV99] A. Quarteroni and A. Valli, *Domain decomposition methods for partial differential equations*, Oxford Science Publications, Oxford, 1999.

[RBP+10] P. Reymond, Y. Bohraus, F. Perren, F. Lazeyras, and N. Stergiopulos, *Validation of a person specific 1-d model of the systemic arterial tree*, XII Mediterranean Conference on Medical and Biological Engineering and Computing 2010 (R. Magjarevic, J. H. Nagel, P. D. Bamidis, and N. Pallikarakis, eds.), IFMBE Proceedings, vol. 29, Springer Berlin Heidelberg, 2010, pp. 578–579.

[RCD+11] P. Reymond, P. Crosetto, S. Deparis, A. Quarteroni, and N. Stergiopulos, *Physiological aspects of fluid structure interaction and their effects on blood flow in a person-specific aorta.*, Submitted, 2011.

[RGCM10] J.-F. Remacle, C. Geuzaine, G. Compère, and E. Marchandise, *High quality surface remeshing using harmonic maps*, Int. J. Numer. Meth. Engng. **83** (2010), 403–425.

[RMP+09] P. Reymond, F. Merenda, F. Perren, D. Rufenacht, and N. Stergiopulos, *Validation of a one-dimensional model of the systemic arterial tree*, Am J Physiol Heart Circ Physiol **297** (2009), no. 1, H208–222.

[RRDH08] J. F. Rodríguez, C. Ruiz, M. Doblaré, and G. A. Holzapfel, *Mechanical stresses in abdominal aortic aneurysms: Influence of diameter, asymmetry, and material anisotropy*, J. Biomech. Eng. **130** (2008), no. 2, 21–23.

[RS87] J. W. Ruge and K. Stüben, *Algebraic multigrid*, Multigrid methods, Frontiers Appl. Math., vol. 3, SIAM, Philadelphia, PA, 1987, pp. 73–130.

[RSFW09] E. Ramm, M. Scheven, C. Förster, and W. A. Wall, *Interaction of incompressible flows and thin-walled structures*, ECCOMAS Multidisciplinary Jubilee Symposium (J. Eberhardsteiner, C. Hellmich, H. A. Mang, and J. Priaux, eds.), Computational Methods in Applied Sciences, vol. 14, Springer Netherlands, 2009, pp. 219–233.

[RVS10] M. Rehman, C. Vuik, and G. Segal, *Block preconditioners for the incompressible stokes problem*, Large-Scale Scientific Computing (I. Lirkov, S. Margenov, and J. Wasniewski, eds.), Lecture Notes in Computer Science, vol. 5910, Springer Berlin / Heidelberg, 2010, pp. 829–836.

[Sal06] M. Sala, *An object-oriented framework for the development of scalable parallel multilevel preconditioners*, ACM Trans. Math. Softw. **32** (2006), 396–416.

[Sar02] M. Sarkis, *Partition of unity coarse spaces and schwarz methods with harmonic overlap*, Lect. Notes Comput. Sci. Eng. **23** (2002), 75–92.

[SD00] A. M. Shaaban and A. Duerinckx, *Wall shear stress and early atherosclerosis*, American Journal of Roentgenology **174** (2000), 1657–1665.

[SH07] G. Scovazzi and T. Hughes, *Lecture notes on continuum mechanics on arbitrary moving domains*, Tech. report, Sandia National Laboratories, Albuquerque, NM, 2007.

[Sim04] V. Simoncini, *Block triangular preconditioners for symmetric saddle-point problems*, Appl. Numer. Math. **49** (2004), no. 1, 63–80.

[SKK02] K. Schloegel, G. Karypis, and V. Kumar, *Parallel static and dynamic multiconstraint graph partitioning*, Concurrency and Computation: Practice and Experience **14** (2002), no. 3, 219–240.

[SMS07] D. Szczerba, R. McGregor, and G. Szekely, *High quality surface mesh generation for multi-physics bio-medical simulations*, Computational Science – ICCS 2007, vol. 4487, Springer Berlin, 2007, pp. 906–913.

[STB03] K. Stein, T. Tezduyar, and R. Benney, *Computational methods for modeling parachute systems*, Computing in Science and Engineering **5** (2003), 39–46.

[SZ92] J. Sokolowski and J. P. Zolesio, *Introduction to shape optimization : shape sensitivity analysis*, Berlin ; New York : Springer-Verlag, 1992 (English), Includes bibliographical references (p. [240]-250).

[TDV07] D. Tromeur-Dervout and Y. Vassilevski, *Pod acceleration of fully implicit solver for unsteady nonlinear flows and its application on grid architecture*, Adv. Eng. Softw. **38** (2007), 301–311.

[Tez04] T. E. Tezduyar, *Finite element methods for fluid dynamics with moving boundaries and interfaces*, Encyclopedia of Computational Mechanics, John Wiley & Sons, Ltd, 2004.

[TF01] H. M. Tufo and P. F. Fischer, *Fast parallel direct solvers for coarse grid problems*, J. Parallel Distrib. Comput. **61** (2001), 151–177.

[TSS06] T. E. Tezduyar, S. Sathe, and K. Stein, *Solution techniques for the fully discretized equations in computation of fluid-structure interactions with the space-time formulations*, Comput. Methods Appl. Mech. Engrg. **195** (2006), no. 41-43, 5743–5753.

[TW05] A. Toselli and O. Widlund, *Domain decomposition methods—algorithms and theory*, Springer Series in Computational Mathematics, vol. 34, Springer-Verlag, Berlin, 2005.

[TY01] P.-S. Tsai and C. B. Yucha, *Noninvasive measurements of central arterial pressure and distensibility by arterial applanation tonometry with a generalized transfer function: Implications for nursing*, Heart & Lung: The Journal of Acute and Critical Care **30** (2001), no. 6, 437–444.

[VdV03] H. A. Van der Vorst, *Iterative Krylov methods for large linear systems*, Cambridge Monographs on Applied and Computational Mathematics, vol. 13, Cambridge University Press, Cambridge, 2003.

[Ver10] C. Vergara, *Nitsches method for defective boundary value problems in incompressibile fluid-dynamics*, J. Sci. Comput. **45** (2010), no. 1-3.

[VMB96] P. Vanek, J. Mandel, and M. Brezina, *Algebraic multigrid by smoothed aggregation for second and fourth order elliptic problems*, Computing **56** (1996), no. 3, 179–196.

[vPVS+09] J. van Prehn, K. Vincken, S. Sprinkhuizen, M. Viergever, J. van Keulen, J. van Herwaarden, F. Moll, and L. Bartels, *Aortic pulsatile distention in young healthy volunteers is asymmetric: Analysis with ecg-gated mri*, European Journal of Vascular and Endovascular Surgery **37** (2009), no. 2, 168 – 174.

[VV87] R. N. Vaishnav and J. Vossoughi, *Residual stress and strain in aortic segments*, J. Biomech. **20** (1987), no. 3, 235 – 237, 239.

[VV05] A. Veneziani and C. Vergara, *Flow rate defective boundary conditions in haemodynamics simulations*, Int. J. Numer. Meth. Engng. **47** (2005), no. 8-9, 803–816.

[WCLB08] H. Wang, J. Chessa, W. K. Liu, and T. Belytschko, *The immersed/fictitious element method for fluid-structure interaction: volumetric consistency, compressibility and thin members*, Internat. J. Numer. Methods Engrg. **74** (2008), no. 1, 32–55.

[WHMW07] D. Wang, O. Hassan, K. Morgan, and N. Weatheril, *Enhanced remeshing from stl files with applications to surface grid generation*, Commun. Numer. Meth. Engng **23** (2007), 227–239.

[WMZ06] J.-Z. Wu, H.-Y. Ma, and M.-D. Zhou, *Vorticity and vortex dynamics*, 1st ed., Springer, May 2006.

[Zei91] E. Zeidler, *Nonlinear functional analysis and its applications, part i: Fixed-point theorems*, Acta Applicandae Mathematicae **24** (1991), 312–314.

[ZFDR08] A. R. Zhao, M. L. Field, K. Digges, and D. Richens, *Blunt trauma and acute aortic syndrome: a three-layer finite-element model of the aortic wall*, European Journal of Cardio-Thoracic Surgery **34** (2008), no. 3, 623–629.