

Reduction techniques for PDEs built upon Reduced Basis and Domain Decomposition Methods with applications to hemodynamics

Présentée le 23 avril 2021

Faculté des sciences de base
Groupe SCI SB SD
Programme doctoral en mathématiques

pour l'obtention du grade de Docteur ès Sciences

par

Luca PEGOLOTTI

Acceptée sur proposition du jury

Prof. D. Kressner, président du jury
Prof. S. Deparis, directeur de thèse
Prof. P. Gervasio, rapporteuse
Prof. P. Blanco, rapporteur
Prof. A. Buffa, rapporteuse

To my parents,
Anna and Renato.

Acknowledgements

My deepest gratitude goes to my thesis director, Prof. Simone Deparis, who gave me the opportunity to pursue my studies in his group. I have greatly enjoyed working under his supervision during the past years and each and every exchange of ideas with him has been a moment of learning. By showing his constant support and trust in my abilities, he contributed to making me a better researcher and to further develop my critical thinking. Without his guidance, this thesis would certainly not have been possible.

Special thanks to the members of the jury: Prof. Daniel Kressner, who kindly agreed to take the role of president, Prof. Annalisa Buffa, Prof. Paola Gervasio, and Prof. Pablo Blanco. Their useful comments helped me ameliorate this dissertation; moreover, our fruitful discussions suggested interesting directions in which the present work could be further developed.

I would also like to express my sincere appreciation to Prof. Alison Marsden, with whom I had the chance to work during the final months of my PhD, for the opportunity to join her group at Stanford University. It has been an invaluable experience to learn from her and other researchers with diverse backgrounds and skills. Among them, I would particularly like to thank Martin for our professional collaboration as well as for the time we got to spend outside the university.

The challenges encountered during my PhD would have been insurmountable without the relief of my friends. Jack and Edone surely take a special place. With them, I shared my first apartment and the first year in Lausanne; they turned what could have been a tough life-style change into a piece of cake. At work, I was fortunate enough to cross paths with the members of CMCS, Antobello, Bartz, Nico, and Spagaz, who taught me everything there is to know about being a PhD student (for example, the importance of *always* logging off the computer whenever not at my desk to avoid unpleasant surprises). I would particularly like to thank Nico and Spagaz, as well as Jean and Felipe, for being great office buddies. I am grateful to have met many extraordinary people in the Mathematics department. I would especially like to mention Alice, Ceci, Edino, Eva, Fabian, Giacomo, il Sergente, Ondine, Riccardo, Stefano, Sundar, with whom I shared happy everyday moments and some of my dearest memories, from the evenings at Sat or Luigia to the day-long hikes in the beautiful Swiss Alps. My most heartfelt gratitude goes to Katy and Luca, for being the perfect sport and mountain companions, consultants, friends, and Nicolino, for all these years of amazing friendship (and flat-sharing).

Acknowledgements

Lastly, I wish to thank my family. None of the (although small) accomplishments I have been fortunate to achieve so far, including this dissertation, would have been possible without the constant support of my parents, Anna and Renato, and my siblings, Giulia and Tommaso. My final thought goes to my grandparents, Anna, Mariuccia, and Vittorio, who I hope would be proud of the work contained in this thesis.

Lausanne, 29 December 2020

Luca Pegolotti

Abstract

This thesis focuses on the development and validation of a reduced order technique for cardiovascular simulations. The method is based on the combined use of the Reduced Basis method and a Domain Decomposition approach and can be seen as a particular implementation of the Reduced Basis Element method. Our contributions include the application to the unsteady three-dimensional Navier–Stokes equations, the introduction of a reduced coupling between subdomains, and the reconstruction of arteries with deformed elementary building blocks. The technique is divided into two main stages: the offline and the online phases. In the offline phase, we define a library of reference building blocks (e.g., tubes and bifurcations) and associate with each of these a set of Reduced Basis functions for velocity and pressure. The set of Reduced Basis functions is obtained by Proper Orthogonal Decomposition of a large number of flow solutions called snapshots; this step is expensive in terms of computational time. In the online phase, the artery of interest is geometrically approximated as a composition of subdomains, which are obtained from the parametrized deformation of the aforementioned building blocks. The local solution in each subdomain is then found as a linear combination of the Reduced Basis functions defined in the corresponding building block. The strategy to couple the local solutions is of utmost importance. In this thesis, we devise a nonconforming method for the coupling of Partial Differential Equations that takes advantage of the definition of a small number of Lagrange multiplier basis functions on the interfaces. We show that this strategy allows us to preserve the h -convergence properties of the discretization method of choice for the primal variable even when a small number of Lagrange multiplier basis functions is employed. Moreover, we test the flexibility of the approach in scenarios in which different discretization algorithms are employed in the subdomains, and we also use it in a fluid-structure interaction benchmark. The introduction of the Lagrange multipliers, however, is associated with stability problems deriving from the saddle-point structure of the global system. In our Reduced Order Model, the stability is recovered by means of supremizers enrichment.

In our numerical simulations, we specifically focus on the effects of the Reduced Basis and geometrical approximations on the quality of the results. We show that the Reduced Order Model performs similarly to the corresponding high-fidelity one in terms of accuracy. Compared to other popular models for cardiovascular simulations (namely 1D models), it also allows us to compute a local reconstruction of the Wall-Shear Stress on the vessel wall. The speedup with respect to the Finite Element method is substantial (at least one

Acknowledgements

order of magnitude), although the current implementation presents bottlenecks that are addressed in depth throughout the thesis.

Keywords: Cardiovascular applications, Navier–Stokes equations, Reduced Order Modeling, Reduced Basis method, Reduced Basis Element method, Domain Decomposition, nonconforming methods.

Résumé

Cette thèse se concentre sur le développement et la validation d'un modèle réduit appliqué aux simulations cardiovasculaires. La méthode est basée sur une combinaison de la méthode des bases réduites et d'une approche de décomposition de domaine, et elle peut être interprétée comme une implémentation spécifique de la *Reduced Basis Element method*. Nos contributions comprennent l'application aux équations de Navier–Stokes non-stationnaires en trois dimensions, l'introduction d'une stratégie de couplage réduit entre sous-domaines, et la reconstruction d'artères avec des composantes géométriques de base déformées. La technique se divise en deux étapes principales : une phase *offline* et une phase *online*. Dans la phase offline, nous définissons une librairie de composantes géométriques de base (par exemple, des tubes et des bifurcations) et nous associons à chacune une collection de fonctions réduites pour la vitesse et pour la pression. Ces fonctions réduites sont générées avec l'algorithme de décomposition orthogonale aux valeurs propres, appliqué à un grand nombre de solutions aux équations de Navier–Stokes, appelées *snapshots* ; cette étape est onéreuse en termes de temps de calcul. Dans la phase online, la géométrie de l'artère d'intérêt est approximée par une composition de sous-domaines obtenus à partir de la déformation paramétrée des composantes géométriques de base mentionnées ci-dessus. La solution locale dans chaque sous-domaine est ensuite trouvée sous forme de combinaison linéaire des fonctions de base réduites définies dans la composante géométrique correspondante. La stratégie de couplage des solutions locales est évidemment importante. Dans cette thèse, nous concevons une méthode non conforme pour le couplage d'équations différentielles partielles qui tire parti, pour la discrétisation des multiplicateurs de Lagrange, de la définition d'un petit nombre de fonctions de base sur les interfaces. Nous montrons que cette stratégie permet de conserver les propriétés de convergence de la méthode de discrétisation choisie pour la variable primale, même lorsqu'un petit nombre de fonctions de base est employé sur les interfaces. De plus, nous testons la flexibilité de l'approche dans des scénarios dans lesquels différents algorithmes de discrétisation sont employés dans les sous-domaines, et nous l'utilisons également dans un benchmark d'interaction fluide-structure. L'introduction des multiplicateurs de Lagrange, cependant, est associée à des problèmes de stabilité découlant de la structure en points-selles du système global. Dans notre modèle réduit, la stabilité est récupérée au moyen de l'enrichissement des *supremizers*.

Dans nos simulations numériques, nous nous concentrons spécifiquement sur les effets de la base réduite et des approximations géométriques sur la qualité des résultats. Nous

Acknowledgements

montrons que le modèle d'ordre réduit fonctionne de manière similaire au modèle haute fidélité correspondant, en termes de précision. Comparé à d'autres modèles populaires de simulation cardiovasculaire (à savoir les modèles 1D), il permet également une reconstruction locale des contraintes de cisaillement sur la paroi du vaisseau. L'accélération par rapport à la méthode des éléments finis est substantielle (au moins un ordre de grandeur), bien que l'implémentation actuelle présente des goulets de étranglement qui sont abordés en profondeur tout au long de la thèse.

Mots clés : Applications cardiovasculaires, équations de Navier–Stokes, modèles d'ordre réduit, méthode des bases réduites, Reduced Basis Element method, décomposition de domaine, méthodes non conformes.

Contents

Acknowledgements	v
Abstract (English/Français)	vii
Contents	xi
List of Figures	xx
List of Tables	xxi
List of Acronyms	xxiii
Introduction	1
1 Mathematical models for hemodynamics	11
1.1 The strong formulation of the incompressible Navier–Stokes equations . . .	12
1.2 Weak formulation and discretization in space of the Navier–Stokes equations	12
1.3 Discretization in time of the Navier–Stokes equations	15
1.3.1 Backward Differentiation Formula schemes	16
1.3.2 Rosenbrock schemes	18
1.4 Elements of the theory of saddle-point problems	24
1.4.1 Existence, uniqueness, stability and convergence results	26
1.4.2 Numerical estimation of the inf-sup constant	29
1.4.3 The case of the Stokes equations	30
1.5 Stabilization of the Navier–Stokes equations by Variational Multiscale modeling	31
1.6 Fluid-structure interaction with deformable domains	33
1.6.1 Strong formulation	33
1.6.2 Space discretization	35
1.7 Concluding remarks	36
2 Non-conforming coupling of PDEs	39
2.1 An introduction to Domain Decomposition methods	40
2.2 Theory of primal hybrid methods	42
2.3 Discretization of the primal hybrid formulation	45

Contents

2.3.1	Generalization to multiple subdomains	48
2.4	Relationship with other nonconforming methods	49
2.4.1	Relationship with the mortar method	49
2.4.2	Relationship with INTERNODES	50
2.4.3	Relationship with the three-field method	50
2.5	Reinterpreting the inf-sup condition of the coupled problem	52
2.6	Numerical results	53
2.6.1	Choice of basis functions for the Lagrange multipliers	54
2.6.2	The Poisson problem on two-way partitioned domains	56
2.6.3	The steady Navier–Stokes equations on three-way partitioned domains	63
2.7	A time-dependent fluid-structure interaction benchmark	71
2.8	Concluding remarks	77
3	Elements of the Reduced Basis method	79
3.1	Motivation	79
3.2	The Reduced Basis method in a nutshell	80
3.2.1	The offline phase: basis construction	81
3.2.2	The online phase: solution of the reduced problem	85
3.2.3	On the efficiency of the online phase: the Discrete Empirical Interpolation Method	86
3.3	The Reduced Basis method in parametrized domains	88
3.3.1	The Piola transformation	89
3.4	Supremizers enrichment for saddle-point problems	90
3.5	The steady Stokes equations in a deformable pipe	93
3.6	Concluding remarks	100
4	Modular Model Order Reduction of blood flow	101
4.1	An overview of Reduced Order Methods for networks of blocks with recurring features	102
4.2	Modular Domain Decomposition of arteries	103
4.2.1	Parametrized geometrical maps for the reference building blocks	105
4.2.2	The continuous Navier-Stokes equations on modular geometries	108
4.2.3	Discretization of the primal hybrid formulation of the flow problem	109
4.3	Efficient solution of the global nonlinear system	114
4.3.1	Bottlenecks of the global preconditioner	115
4.4	The Reduced Basis Element method for flow in arteries	118
4.4.1	Supremizers for pressure and coupling Lagrange multipliers	122
4.4.2	Assembly and solution of the global reduced system	123
4.5	Concluding remarks	126

5	Results on artificial and physiological geometries	129
5.1	Automatic generation of modular geometries	130
5.1.1	Generation of meshes for cardiovascular simulations	130
5.1.2	A greedy algorithm for geometric assembly	131
5.1.3	Limitations and alternatives	135
5.2	Online phase on an artificial geometry	136
5.3	Online phase on the aorta and iliac arteries	141
5.4	Online phase on the aortic arch	144
5.5	Concluding remarks	150
6	Conclusions	151
6.1	Perspectives and future work	152
	Curriculum Vitae	173

List of Figures

1	Geometry of the aortic arch, with detail of the computational mesh on the right.	2
1.1	Convergence in time for various Rosenbrock schemes. The solid lines in the convergence plots for the pressure refer to the pressure computed with the correction at the end of the timestep by solving Eq. (1.10), while the dot-dashed lines are obtained without correction.	20
1.2	On the top, iso-contour of the Q-criterion with $Q_{\text{iso}} = 6$ of the solution of the flow past a cylinder obtained using the ROS2 scheme with $\Delta t = 5 \cdot 10^{-3}$ s at $t = 4$ s. On the bottom, magnitude of the velocity field of the same solution at $t = 4$ s on a plane perpendicular to the cylinder and passing at half the height of the domain, with detail of the computational mesh.	22
1.3	Drag and lift coefficients. The red solid line refers to c_D (left) and c_L (right) obtained with the ROS2 scheme with fixed $\Delta t = 5 \cdot 10^{-3}$ s. The black dashed line refers to the reference solution reported in [Joh06].	24
1.4	Total GMRES iterations (i.e., sum of the iterations of each stage) when solving the Navier–Stokes equations by using ROS3Pw with adaptive timesteps. The tolerance of the linear solver is set to 10^{-11} with stopping criterion based on the residual norm relative to the initial one.	25
1.5	Some generalized Taylor–Hood elements in two and three dimensions. The velocity and pressure elements are on the top and on the bottom, respectively. For each element, the position of the mesh node is marked with red dots.	30
1.6	Scheme of the ALE mapping.	34
2.1	Example of partitioned domain $\Omega = \Omega_1 \cup \Omega_2$	42
2.2	Example of a three-way partition of Ω with three interfaces; each interface corresponds to a Lagrange multiplier space.	48
2.3	Condition number of the discretized system for the Poisson problem on two subdomains (see Section 2.6.2) vs number of basis functions for the Lagrange multiplier space N_λ^δ . On the left, we consider the nonorthonormal basis functions in Eq. (2.17), while on the right we consider their orthonormalization.	55

List of Figures

2.4 Contour lines of the solution (top row) and derivatives of the solution at both sides of the interface, approximated Lagrange multiplier and exact derivative at the interface (bottom row) when $N_\lambda^\delta = 1$ (left column), $N_\lambda^\delta = 3$ Fourier modes (middle column) and $N_\lambda^\delta = 5$ Fourier modes (right column) are used to characterize the space Λ^δ 57

2.5 Convergence of error in broken norm with respect to the mesh size h and number of basis functions on the interface N_λ^δ with conforming (left) and nonconforming meshes (right). The black dashed line shows the convergence of the error with a conforming mesh with comparable mesh size. 58

2.6 Decaying of the inf-sup constant $\tilde{\beta}$ with respect to N_λ^δ 59

2.7 On the left, subdomains of the cylinder with corresponding meshes (sizes: $h_1 = 1.61 \cdot 10^{-1}$ and $h_2 = 1.76 \cdot 10^{-1}$). On the right, the solution u obtained with $N_\lambda^\delta = 28$ 60

2.8 Convergence of the H^1 -error in the solution u in Ω_1 and Ω_2 (left) and convergence of the L^2 -error in the Lagrange multiplier λ (right). In Ω_1 and Ω_2 we use linear and quadratic finite elements, respectively. 61

2.9 GMRES iterations for solving the (preconditioned) linear system on the problem presented in Section 2.6.2 vs number of basis functions for the Lagrange multiplier space. 62

2.10 Three-way partition of the model bifurcation. The displayed mesh corresponds to the size $h = 1.32 \cdot 10^{-1}$ 63

2.11 On the left, solution of the Navier–Stokes computed on a mesh with size $h = 7.13 \cdot 10^{-2}$ and employing $N_\omega^\delta = 4$ frequencies (corresponding to $N_\lambda^\delta = 54$ basis functions for Λ^δ). On the right, convergence of the error (as defined in Eq. (2.21)) with respect to the mesh size h , for different values of N_λ^δ 66

2.12 Error distribution on the bifurcation when employing $N_\omega^\delta = 0$ (top row), $N_\omega^\delta = 2$ (middle row) and $N_\omega^\delta = 4$ (bottom row) and a coarse mesh (left column, $h = 9.29 \cdot 10^{-2}$) and a fine mesh (right column, $h = 3.11 \cdot 10^{-2}$). . 67

2.13 Global solution obtained by considering 6×12 elements and P3-P2 B-Splines basis functions in Ω_1 , a FE mesh with size $h = 7.16 \cdot 10^{-2}$ and P2-P1 basis functions in Ω_2 , 5×10 elements and P3-P2 B-Splines basis functions in Ω_3 , and $N_\omega^\delta = 5$ frequencies for the discretization of Λ^δ . The control points are marked with red dots. 69

2.14 Convergence of the error in Ω_3 with respect to the mesh size h and number of basis functions on the interface N_λ^δ . The basis functions in Ω_3 are P2-P1 B-Splines (left) and P3-P2 (right). 70

2.15 Error in Ω_1 (left) and Ω_2 (right) with respect to the mesh size h in Ω_3 . In the latter subdomain, we employ P3-P2 B-Splines basis functions. 70

2.16 Sketch of the fluid (inner cylinder) and structure (outer layer) domains. . 72

2.17	Velocity solution of the fluid velocity on the deformed domain (displacement is magnified by 10 times) at $t = 4 \cdot 10^{-3}$ s and detail of the computational mesh. The global mesh is conforming.	73
2.18	Pressure wave in the fluid domain at three timesteps and for different discretization choices of the Lagrange multiplier space on the fluid-structure interface. The deformation of the domain is amplified by 10 times for demonstration purposes.	74
2.19	Velocity magnitude and pressure along the centerline of the fluid domain.	75
2.20	L^2 norms of velocity, pressure and displacement for different degrees of sizes of the Lagrange multiplier space on the interface.	76
3.1	Illustration of the RB method in the case $N_\mu = 1$. Image inspired by [QMN15].	81
3.2	Comparison of singular values computed with the <code>svd</code> method in MATLAB and with the correlation matrix approach presented in Algorithm 3.1.	85
3.3	Effect of the Piola transformation. The vector field defined on the reference domain $\hat{\Omega}$ is mapped without scaling on top and with scaling by the Piola transformation on the bottom.	89
3.4	Magnitude of the velocity field of the steady Stokes equations on the reference domain $\hat{\Omega}$ and on the deformed one $\Omega([\mu_1, \mu_2])$. The geometrical parameters are the outlet diameter μ_1 and the bending angle μ_2	93
3.5	Decay of the first 40 velocity (left) and pressure (right) singular values.	95
3.6	Diagram of the RB algorithm. The green boxes are specific to the application of the steady Stokes equations in parametrized domains.	96
3.7	RB velocity field magnitude (left column), RB pressure distribution (middle column) and velocity error magnitude with respect to FE solution (right column), for three different choices of the geometrical parameters. First row: $\mu_1 = -0.21$ and $\mu_2 = 1.08$. Second row: $\mu_1 = -0.10$ and $\mu_2 = 0.83$. Third row: $\mu_1 = -0.41$ and $\mu_2 = 0.92$	97
3.8	Top left: L^2 error in pressure (L2p) and H^1 error in velocity (H1u), with and without scaling of the velocity by the Piola transformation vs POD tolerance (the same for pressure and velocity). Top right: H^1 errors in velocity vs basis size, with and without Piola scaling. Bottom left: L2p and H1u errors vs online runtime in seconds. Bottom right: L2p and H1u errors with and without supremizers vs POD tolerance.	99
4.1	Sketch of the Domain Decomposition of a target geometry. Each block in the target geometry is found from the parametrized geometrical deformation of a small number of reference building blocks.	103
4.2	Types of reference building blocks and affine transformations. On the left, tubes. On the right, bifurcation.	105

List of Figures

4.3	Deformation of the model bifurcation. On the right, we show examples in which the geometrical parameter—i.e., the rotation angles of the outlets—is (top) and is not (bottom) included in the admissible set.	106
4.4	Basis functions P_k^n on the unit disk \mathcal{D} , for $n \leq 3$, mapped onto the target interface $\Gamma^{[ij]}$. Minimum and maximum values are plotted in blue and yellow, respectively. The basis function P_0^0 is constant.	110
4.5	Parallel computation of a single block of the Schur complement \mathcal{S}	116
4.6	Solution time of a single linear system preconditioned as discussed in Section 4.3 (right-top) and corresponding number of FGMRES iterations (right-bottom), as functions of number of subdomains N_Ω in the geometry of the aorta and iliac arteries (left). Blue solid lines: $\tilde{\mathcal{A}}^{-1}$ approximated with a single application of SIMPLE for each subdomain; red dashed lines and green dash-dotted lines: $\tilde{\mathcal{A}}^{-1}$ solved with GMRES and tolerance $5 \cdot 10^{-1}$ and 10^{-2} , respectively; circles: $N_\lambda^\delta = 3$; diamonds = $N_\lambda^\delta = 84$	117
4.7	Offline phase on an artificial geometry featuring $N_\Omega = 9$ subdomains and $N_{\text{bb}} = 4$ building blocks (left) and singular values decay for the velocity and pressure reduced bases built on the bifurcation (right). The colored dots on the plot on the right refer to different values of POD tolerances ε_u and ε_p	118
4.8	First four modes of velocity (left) and pressure (right) for the four building blocks. From top to bottom: bifurcation (B), tubes with aspect ratio length/diameter 1:1 (T1), 1:2 (T2) and 1:3 (T3).	120
4.9	Average over time of the RB velocity solutions—normalized with respect to the first coefficient—in the four building blocks with $\varepsilon_u = 4 \cdot 10^{-3}$ and $\varepsilon_p = 8 \cdot 10^{-5}$. The left and right dashed lines in every plot correspond to the indices 40 and 120.	125
4.10	Flowchart of the geometrical approximation and setup phases of the method.	127
4.11	Flowchart of the solve phase of the method.	128
5.1	Pipeline for the generation of meshes in cardiovascular simulations. From left to right and from top to bottom: centerline generation, segmentation, model lofting, and mesh generation.	132
5.2	After the tube is rotated, scaled and translated so that $\Gamma_{\text{in}} = \Gamma(t_{\text{in}})$ is satisfied, the two remaining parameters (the rotation angle along its axis α and the bending angle μ_3) are determined such that the center and the normal of the outlet Γ_{out} approximately match the centerline point $\mathbf{z}(\tilde{t})$ and the tangent $\dot{\mathbf{z}}(\tilde{t})$	134

5.3 The left and right columns—each composed of two sub-columns of plots—refer to time $t = 0.15$ s and $t = 0.25$ s, respectively. First row: velocity magnitude volume plot of the RB solution (sub-column left) and magnitude of the point-wise velocity error w.r.t. the FE solution (sub-column right). Second row: pressure plot of the RB solution (sub-column left) and absolute value of the point-wise pressure error w.r.t. the FE solution (sub-column right). Third row: magnitude of the WSS of the RB solution (sub-column left) and magnitude of the point-wise WSS error w.r.t. the FE solution (sub-column right). The RB solution corresponds to the choice $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$ 137

5.4 Error in velocity e_u (left) and error in pressure e_p (right), computed as in Eq. (5.1), in function of the POD tolerances for velocity and pressure ε_u and ε_p 138

5.5 Errors of reduced velocity and pressure against the FE solution vs time. The colored lines refer to different choices of N_c for the approximation of the nonlinear convective term. The black dashed lines show the reference errors obtained without approximation of the convective term with $\varepsilon_u = 4 \cdot 10^{-3}$ and $\varepsilon_p = 8 \cdot 10^{-5}$ (which are the same tolerances used in the simulations corresponding to the colored lines) and $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$ 140

5.6 On the left, qualitative comparison of the reference mesh with the decomposed one. On the right, quantitative estimation of the distance between the two. 141

5.7 The left and right columns refer to time $t = 0.9$ s and $t = 1.25$ s. First row: velocity magnitude volume plot of the RB and reference solutions. Second row: pressure plot of the RB and reference solutions. Third row: magnitude of the WSS of the RB and reference solutions. The RB solution corresponds to the choice $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$ 143

5.8 Average WSS (in dyn/cm²) on the three regions marked on the figure on the left (top row), and pressure p (in dyn/cm²) and flow rates Q (in cm³/s) at inlet and outlets, respectively (bottom row). The black dashed line refers to the reference solution computed by SimVascular, whereas the 4 colored lines are obtained with different RB settings. RB1: $\varepsilon_u = 4 \cdot 10^{-3}$, RB2: $\varepsilon_u = 8 \cdot 10^{-3}$ and $N_c = 80$, RB3: $\varepsilon_u = 6.4 \cdot 10^{-2}$ and $N_c = 40$, RB4: $\varepsilon_u = 6.4 \cdot 10^{-2}$ and $N_c = 20$. In all cases, $\varepsilon_p = 10^{-5}$ 145

5.9 On the left, geometry decomposed into blocks. On the right, pattern of the global matrix (the dimensions of the individual blocks are in scale) when employing the RB tolerances $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$. LC: left common carotid artery. LS: left subclavian artery. RC: right common carotid artery. RS: right subclavian artery. 146

5.10 Velocity magnitude (left), pressure (center) and WSS on the vessel boundary (right), at $t = 0.95$ s and $t = 1.25$ s (top and bottom row, respectively). The results are obtained with POD tolerances $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$ 148

List of Figures

- 5.11 Flow rates at the inlet and outlets, and velocity field magnitude at $t = 1$ s across several slices of the geometry. RB1: $\varepsilon_u = 10^{-2}$. RB2: $\varepsilon_u = 10^{-3}$. . 149

List of Tables

1.1	Number of stages, theoretical and measured orders of the selected Rosenbrock schemes. The numerical rate of convergence are approximated as slopes of the linear regression lines computed from the errors in Fig. 1.1.	21
1.2	Number of timesteps, maximum timestep size Δt_n^{\max} , average timestep size $\overline{\Delta t_n}$ and c_L^{\max} as a function of the tolerance τ of the time adaptivity scheme.	25
3.1	Bases sizes as functions of POD tolerances.	95
4.1	Velocity RB sizes $N_{u,i}$ with respect to different POD tolerances ε_u , for 4 different building blocks (B: bifurcation, T1, T2 and T3: tubes with aspect ratios diameter/length 1:1, 1:2, 1:3, respectively).	119
4.2	Pressure RB sizes $N_{p,i}$ with respect to different POD tolerances ε_p , for 4 different building blocks (B: bifurcation, T1, T2 and T3: tubes with aspect ratios diameter/length 1:1, 1:2, 1:3, respectively).	120
5.1	Overall speedups w.r.t. the FE solution and, in parenthesis, speedups of the solve part of the online phase, i.e. speedup relative to the total running time excluding the setup part in which the reduced bases are loaded and the constant matrices are assembled and projected onto the reduced spaces.	139

List of Acronyms

ALE	Arbitrary Lagrangian Eulerian
BDF	Backward Differentiation Formulas
CFD	Computational Fluid Dynamics
dof	degree of freedom
DEIM	Discrete Empirical Interpolation Method
FE	Finite Element
FOM	Full Order Model
IGA	Isogeometric Analysis
MDEIM	Matrix Discrete Empirical Interpolation Method
MOR	Model Order Reduction
PDE	Partial Differential Equation
POD	Proper Orthogonal Decomposition
RB	Reduced Basis
SUPG	Streamline–Upwind Petrov–Galerkin
RBE	Reduced Basis Element
ROM	Reduced Order Model
VMS	Variational Multiscale
WSS	Wall-Shear Stress

Introduction

Cardiovascular disease is currently the leading cause of death worldwide. According to the data of the World Health Organization¹, this class of diseases has been responsible for around 31% of all the deaths in 2016. Under the broad definition of cardiovascular disease are various pathological conditions—such as coronary heart disease, stroke, or heart failure—which are fostered by lifestyles that are rapidly becoming prevalent in many areas of the developed and developing world. Risk factors include, for example, unhealthy diet, obesity, lack of physical exercise, air pollution, and many more. Therefore, it is not surprising that the study of the cardiovascular system is nowadays an extremely active area of research.

From a global perspective, the cardiovascular system resembles a (complex) mechanical system of interconnected pipes, arteries, and veins, whose purpose is to distribute oxygenated blood over all the districts of the body and to collect it once oxygen has been released. Blood flow is propelled by the action of the heart, which can be interpreted as a mechanical pump. The utmost importance of the mechanical properties of the cardiovascular system is also supported by the fact that many of the aforementioned pathological conditions are caused by abnormal patterns in the blood dynamics and their effects on the vessel walls. For example, atherosclerosis, which is a disease caused by the formation of lipidic plaques on the inner walls of blood vessels, has been shown to be related to the Wall-Shear Stress (WSS) exerted by the fluid [CG05, SD00].

The numerical approximation of blood flow by Computational Fluid Dynamics (CFD) has gained considerable attention during the last twenty years as a valuable quantitative aid for the study and diagnosis of cardiovascular disease [BBC⁺14, FTM17]. This tendency is also promoted by the exponential rise in the computational capabilities of supercomputers (as predicted by Moore's law [Moo65]), which allows researchers and scientists to run simulations with ever-increasing complexity.

Blood dynamics is typically modeled by means of the incompressible Navier–Stokes equations. These are a set of nonlinear Partial Differential Equations (PDEs) derived from the application to continuums of basic principles of continuity of mass, energy,

¹[https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds))

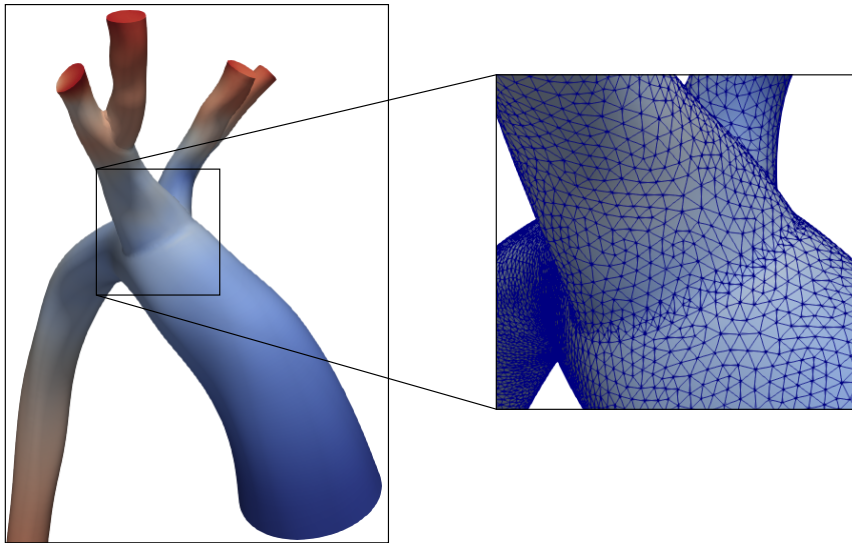


Figure 1 – Geometry of the aortic arch, with detail of the computational mesh on the right.

and momentum. To attempt the numerical simulation of these equations, it is necessary to consider a suitable finite-dimensional approximation. Here, this is performed by means of the Finite Element (FE) method, although other methods—e.g., Finite Volume and Isogeometric Analysis (IGA)—are also popular in the context of fluid simulations. The basic underlying idea of the FE method is to fill the domain of interest with nonoverlapping polyhedra (e.g., tetrahedra or hexahedra) to form a mesh [SF73, Cia02]. The infinite-dimensional variables of interest, namely velocity and pressure in the case of the Navier–Stokes equations, are approximated as combinations of a finite number of basis functions with local support on the elements of the computational mesh. In Fig. 1 we show, as an example, a geometry of the aortic arch and a detail of its triangular surface mesh.

However, accurate results typically require very fine mesh sizes and a large number of degrees of freedom (dofs). The latter is proportional to the size of the linear systems to be solved at each timestep. The computational time associated with large-scale simulations is, consequently, often prohibitive in daily clinical practice. Additionally, certain *multi-query* scenarios require solving these problems multiple times for a variety of data (e.g., variable boundary conditions or physical parameters of blood and/or artery wall). This is a recurring theme, for instance, whenever one is interested in parameter identification or in quantifying the uncertainty associated with the quantities to be varied [SKCT16, SM11, CQR13].

The FE method is one of the viable techniques to define what in this thesis we refer to as

Full Order Models (FOMs) or high-fidelity models. *Reduced Order Models* (ROMs) aim at drastically reducing the number of dofs compared to the high-fidelity ones. Reducing the number of variables involved comes at the cost of larger approximation errors, but this decreased accuracy is generally still well within the bounds of acceptability (depending, of course, on the application at hand).

Among the ROMs employed in the context of cardiovascular modeling are the popular 0D/1D models [MB13, Mal12, MEM15]. These models consider a coarse approximation of systems of arteries as electric circuits (0D) or as segments in which the quantities of interest are found as averages across the section of the vessels (1D). Geometrical multiscale models often prove to be remarkably accurate in approximating flow rates and pressure drops [BBM⁺18], albeit the strong geometrical approximations inevitably entail a significant loss of local details. For this reason, algorithms to couple geometrical multiscale models with full three-dimensional simulations—to be employed in the regions in which higher-quality solutions are required—have been devised [MBC⁺13, SMK⁺12, MVCF⁺13].

The main goal of this thesis is the formalization and assessment of a ROM for cardiovascular simulations allowing to preserve the three-dimensional features of the flow whilst still considerably reducing the size of the system. In particular, we base our strategy on a combination of a Domain Decomposition approach with the Reduced Basis (RB) method. The RB method has been introduced in [ASB78] for nonlinear structural analysis and has been further developed, e.g., in [Bal96, FR83, NP83, Por85]. This reduction strategy has received much attention in the last two decades; see [QMN15, HRS16] for general overviews. Our method is a particular implementation of the Reduced Basis Element (RBE) method [MR04, MR02]. The domain of interest is approximated as a composition of subdomains obtained from the parametrized deformation of a small number of elementary building blocks. In the cardiovascular context, these are, for example, reference cylinders and bifurcations. Instead of relying on the FE space associated with each of these building blocks, the global solution to the Navier–Stokes equations is found by combining the local solutions in each subdomain, which in turn are retrieved as a linear combination of a small number of spectral basis functions defined in every building block.

A critical part of the algorithm is evidently the coupling strategy. In this thesis, we present an approach based on spectral basis functions defined on the interfaces of the subdomains. From the theoretical standpoint, the method is based on the same concepts that lead to the well-known mortar method [Ber89, BMR05]. The main advantage of considering spectral basis functions is that the space associated with the coupling functions—the Lagrange multipliers—is independent of either of the neighboring meshes, which results in the possibility of freely tuning the degree of coupling.

In the following sections, we first present the main objectives of this work. Then, we highlight the principal original contributions of the manuscript and discuss its outline. We conclude with general remarks on the implementation of the code that has been

employed in the numerical results presented throughout the thesis.

Objectives

As mentioned, the main goals of the thesis are the development of a ROM for cardiovascular simulations based on the RBE method and the assessment of its performance on physiological geometries. In particular, we address the following research questions:

- *How to couple solutions defined on globally nonconforming spaces?*
In the literature, the task of combining solutions defined in separate domains and on a priori incompatible functional spaces is a well-known problem. The most common approach introduces, for each interface, a set of basis functions for a Lagrange multiplier used to impose the kinematic and dynamic conditions (i.e., continuity of the solutions and relative stresses across the interface). In this thesis, we propose a method in which the space of the Lagrange multiplier is discretized by a set of spectral basis functions. We study the stability and convergence properties of this approach and put it in relationship with other popular methods for the nonconforming coupling of PDEs.
- *What are the main strengths and criticalities/bottlenecks of the ROM?*
Our ROM is a data-driven approach in which the local basis functions are computed by Proper Orthogonal Decomposition (POD) of large datasets of local solutions computed during an expensive offline stage. The goal of this step is to lower the computational burden of the solution of the flow problem given a new geometry (online stage). In this work, we aim at formalizing the general framework of the ROM by identifying the steps which characterize both the offline and online stages. We also focus on the advantages of our method compared with other reduced order modeling techniques and on the current bottlenecks that still need further investigation.
- *How does the ROM compare with respect to the FOM in terms of efficiency and accuracy?*
The main reason to consider reduced order modeling in the first place is to obtain substantial speedup with respect to the high-fidelity simulations. For the ROM to be beneficial in practical scenarios, however, the overall accuracy must not be excessively compromised. In the particular case of our ROM, the loss of accuracy is dictated by two factors: the geometrical approximation and the use of RB functions instead of the richer FE ones. The goal of our numerical experiments is to evaluate both the expected gain in performance and the accuracy loss associated with each of these aspects.

Contributions and outline of the thesis

The main contributions of the thesis are:

- We discuss the main results achieved during the study of a time-integration method applied to the Navier–Stokes equations, namely the Rosenbrock schemes [Ros63]. In particular, we focus on two critical points that affect these methods: order reduction and the issues related to the convergence of the linear solver. Although we do not use Rosenbrock schemes in the following parts of the thesis, these findings motivate the time discretization strategy that is considered in our numerical simulations.
- We perform an in-depth study of a coupling strategy based on spectral Lagrange multiplier basis functions. As we show in our numerical experiments, we are able to recover the convergence rates of the discretization method employed in the subdomains by considering a minimal number of basis functions. The method is tested on a variety of cases, ranging from FE-FE coupling (with global nonconforming meshes or different polynomials degrees in the subdomains) to FE-IGA coupling. Furthermore, the coupling strategy is validated on a classic fluid-structure interaction benchmark in the Arbitrary Lagrangian-Eulerian (ALE) framework, in which we show that a small number of spectral basis functions is sufficient to obtain results close to the ones achieved with a strong coupling of fluid and structure displacements.
- We propose a particular implementation of the RBE method [MR02] which takes advantage of the spectral basis functions discussed in the previous point. Contrarily to previous investigations of the RBE method in the cardiovascular context, moreover, we define analytical functions for the deformation of some of our building blocks (the tubes), and we perform the mapping of others (the bifurcations) by solving linear elasticity problems in which we prescribe the displacement of the boundaries so as to match the target geometry. In the RBE context, the typical approach is instead based on transfinite maps [LMR06b, LMR09, IQR12, JIR14].
- We develop an ad-hoc preconditioner that exploits the saddle-point nature of the coupled problem. The main idea is to rely on the same decomposition of the matrix system leading, e.g., to the SIMPLE preconditioner [SRV10]. The critical points are the computation of the Schur complement and the inversion of the matrices relative to each subdomain. Regarding this last aspect, our numerical tests show that the largest gains in terms of computational time are achieved when the inverses of the primal matrices—which in our application are the Jacobians of the Navier–Stokes equations in every subdomain—are approximated with a single application of the SIMPLE preconditioner. It is worth noting that it is important to focus on the efficiency of the solution of the global system even in the FE context since the offline phase of the ROM requires sampling snapshots on already decomposed geometries.

The development of an effective preconditioner is, therefore, fundamental to lower the cost associated with this computationally intensive phase of the method.

- We evaluate the performance of the ROM on artificial and physiological geometries. We highlight that, even though the RBE method has been originally proposed for cardiovascular applications, to our knowledge, this work represents its first assessment in the context of the unsteady three-dimensional Navier–Stokes equations. Indeed, previous works in this domain have focused primarily on the steady Stokes equations in two dimensions; see, e.g., [LMR06a, LMR06c, LMR07, IRQ10, IQR12, Iap12]. One of the main topics of interest is the effect of the geometric discretization on the accuracy of the method. The automatic generation of decomposed geometries, although of the utmost importance for the effective use of the methodology in practical scenarios, is only tangentially presented in Section 5.1.

The structure of the thesis is the following.

Chapter 1 deals with the mathematical models which are commonly used in cardiovascular applications. In particular, we start from the strong formulation of the Navier–Stokes equations and then present the discretization in space by the FE method. The discretization in time is performed by Backward Differentiation Formulas (BDF) schemes, but we also discuss the properties of the aforementioned Rosenbrock schemes. Furthermore, we introduce the concept of saddle-point problems and some related stability and convergence results, which play an important role in the approximation of the Navier–Stokes equations and other areas of the thesis. We conclude by presenting a fluid-structure multiphysics problem in ALE formulation in strong form and its space discretization.

Chapter 2 is about our nonconforming strategy for the coupling of PDEs. We recall that, in this context, the term *nonconforming* refers to the concept that the global functional space in the discretized problem (i.e., the composition of the local functional spaces in every subdomain) is not a subspace of the functional space in which the continuous solution resides. In practical terms, nonconforming coupling methods allow us to combine solutions defined on globally nonconforming meshes (that is, meshes featuring hanging nodes) or with different discretization methods (for example, FE-FE, FE-IGA, and FE-RB). In this chapter, we discuss how our discretization strategy for the global problem is based on the definition of spectral Lagrange multiplier basis functions on the interfaces. We show that this allows us to retain desirable convergence properties of the underlying discretization strategy for the primal variable of the problem. Finally, we draw a parallel between the theory of the nonconforming coupling of PDEs and the fluid-structure interaction problem discussed in the previous chapter, and we also employ an ad-hoc set of spectral basis functions to solve the benchmark problem proposed in [Nob01].

In **Chapter 3**, we present fundamental concepts of the RB method. We start by

providing a general overview of the method and focusing on the offline and online phases separately. Then, we specifically focus on problems featuring geometrical parameters. This is motivated by the fact that the subdomains in our ROM are obtained from parametrized geometrical deformations of reference building blocks. We address two particular aspects that are problematic when dealing with parametric deformations in fluid simulations: the lack of an affine decomposition and the need to perform the Piola transformation of the velocity field (in particular, the latter point is required to preserve the divergence-free property of the RB functions). Furthermore, we discuss the matter of the stability of the reduced system whenever dealing with saddle-point problems (such as the Stokes or the Navier–Stokes equations). In the last part of the chapter, we present a somewhat elementary two-dimensional example in which we aim at solving the steady Stokes equations in a parametrized tube. Despite its simplicity, this example requires putting into practice all the topics introduced in this chapter.

In **Chapter 4**, we present our ROM. We start with an overview of other instances of uses of the RBE method in the context of cardiovascular simulations. As mentioned, the existing literature focuses primarily on steady simulations of the Stokes equations. Then, we formalize the method by introducing a Domain Decomposition of the geometry of interest and by providing the weak formulation of the Navier–Stokes equations on modular geometries. We also address the efficient solution of the global problem when the FE method is employed in every subdomain. In the last part of the chapter, we first tackle the offline phase of our ROM and then move to the online part.

Chapter 5 focuses on numerical results computed on an artificial geometry—as a matter of fact, the same geometry that has been considered in the previous chapter during the basis generation phase of the algorithm—and then move on to simulations on physiological geometries. The goal of the simulations on the artificial geometry is to evaluate the effects of the sole reduction by the RB method on the accuracy and efficiency of the solution. Indeed, considering an already decomposed model as a benchmark allows us to establish a comparison with a FE simulation that shares the same geometry and mesh (consequently, the same number of FE dofs). With the second simulation, which is performed on the physiological geometry of an aorta with the iliac arteries, we aim to evaluate the effects of the geometrical approximation on the solution quality. The last simulation is performed on the geometry of an aortic arch: here, we show the feasibility of employing a mixture of RB and FE solutions in the decomposed geometry. The rationale to do so is that, in some of the subdomains, it could be desirable to consider exact geometries (e.g., whenever a bifurcation is not well approximated by the ones in our library of building blocks, or whenever we are dealing with pathological cases with abnormal geometries such as stenoses) which are not equipped with RB basis functions.

Finally, in **Chapter 6** we draw some conclusions and analyze the current limitations of the work presented in this thesis. We also discuss possible future directions for our study.

Introduction

The content of this thesis is based on two published papers ([DDM⁺19, DIP19]) and an additional one ([PPMD20]) that is currently under review for publication. Specifically: the part of Chapter 1 relative to the study of Rosenbrock schemes for the Navier–Stokes equations is an extract of [DDM⁺19], Chapter 2 is largely taken from [DIP19] (except for the last part, which deals with the application of the coupling strategy to fluid-structure interaction), and Chapter 4 and Chapter 5 are based on the topics and results presented in [PPMD20].

Implementation details

The numerical simulations in this thesis are carried out with many different codes and pieces of software.

The two-dimensional simulations in Chapter 2 and Chapter 3 are computed with a freely downloadable suite of MATLAB functions for FE simulations on structured and unstructured meshes². The code has been developed from scratch (i.e., it is not based on any pre-existing library) during the doctoral program duration. The simulations involving IGA in Chapter 2 are instead performed in GeoPDEs [Váz16], a free and open-source package written in OCTAVE and MATLAB for IGA.

Most of the three-dimensional simulations are obtained with LifeV³ [BDF⁺17], a C++ library for high performance numerical simulations with FEM. In some cases, we also rely on SimVascular⁴ [UWM⁺17], an open-source program for patient-specific modeling and blood flow simulations; we explicitly specify in the text whether a certain result is found with this specific software.

The results presented in Chapter 1 and relative to the Rosenbrock schemes required to implement and integrate that discretization strategy in LifeV.

The results presented in Chapter 5 are realized with a C++ library developed for the purposes of this thesis and based on LifeV. A comprehensive description of the code is out of the scope of this manuscript; here, we only report its main characteristics and capabilities.

- *File format of the modular geometries.* The geometrical characteristics of the modular geometries employed in Section 5 are encoded in `.xml` files. The elementary entity of this kind of format is a *node*, which is characterized by a variable number of properties. Furthermore, nodes are arranged in a tree-like structure (namely, a particular node can have one or multiple children or parents). The reasons for which this format is appropriate to our application are twofold: (i) every block of

²<https://github.com/lucapegolotti/feamat>

³<https://bitbucket.org/lifev-dev/lifev-release/wiki/Home>

⁴<http://simvascular.github.io/>

the modular geometry can be represented with a node in which the attributes are its corresponding geometrical parameters, and (ii) the modular geometry can be conceptually interpreted as a tree in which every block is followed by either one or multiple children; we restrict ourselves, however, to a single parent (namely a single inflow per subdomain). Generating a new modular geometry, therefore, is equivalent to generating the corresponding `.xml` file: in this thesis, this is done by following the greedy algorithm presented in Section 5.1, which has also been implemented in the library.

- *Polymorphic treatment of assemblers in building blocks.* The library has been developed by following a classic C++ paradigm, namely polymorphism. This has two main advantages. Firstly, the code is easily extendible to include new assemblers (in addition to those already implemented, which correspond to the Stokes and Navier–Stokes equations), as the assembler of the global system is agnostic about the specific type of local assembler in every subdomain (in particular, all the assemblers are children of a common abstract class). The second advantage is that, owing to the flexibility of the global assembler, it is possible to allocate instances of different assemblers in different subdomains (e.g., FE assemblers in certain parts of the domain and RB assemblers in others) by directly specifying the assembler type in the nodes of the `.xml` file. The results presented in Section 5.4 take advantage of this property.
- *Polymorphic treatment of Lagrange multiplier basis functions.* Similarly to the previous point, the basis functions on the interfaces share a common abstract interface, and this allows users to integrate new classes of functions in the code easily.
- *Dynamic processing of simulation data.* The simulation data regarding boundary conditions (e.g., inflow profile), physical properties of the fluid, and POD tolerances for the reduced bases, are passed to the simulation executables using formatted datafiles. This allows users to efficiently vary these parameters without the need to recompile the code.

The most computationally demanding numerical results presented in this thesis are found on the Deneb, Fidis, and Helvetios clusters and, for this, we gratefully acknowledge the Scientific IT and Application Support of EPFL⁵.

⁵<https://www.epfl.ch/research/facilities/scitas/>

1 Mathematical models for hemodynamics

This chapter is meant to set the theoretical basis for the topics that are treated in the rest of the thesis. In particular, we focus on the mathematical aspects that characterize the modeling and simulation of blood dynamics. Hemodynamics is typically approximated using the incompressible Navier–Stokes equations, a set of nonlinear PDEs featuring the velocity and pressure of the fluid as variables. A more comprehensive description of blood flow behavior in compliant arteries is obtained by combining the Navier–Stokes equations with elastic equations modeling the response of the vessel walls to the fluid stresses.

The structure of the chapter is the following. The continuous formulation of the Navier–Stokes equations is presented in Section 1.1, whereas their space and time discretizations are treated in Sections 1.2 and 1.3, respectively. The Navier–Stokes equations belong to a broad class of mathematical problems commonly referred to as saddle-point problems. These are discussed in a general fashion in Section 1.4, where we also touch on some of the main theorems regarding well-posedness, stability, and convergence properties. The importance of these concepts within the context of the thesis is also motivated by the fact that, in Chapter 2, we introduce a coupling strategy for PDEs based on a saddle-point formulation. In Section 1.5 we describe the Variational Multiscale (VMS) approach for the discretization of the Navier–Stokes equations. Finally, in Section 1.6 we introduce the idea of fluid-structure interaction, and we focus in particular on an Arbitrary Lagrangian-Eulerian (ALE) formulation in which the deformation of the structure determines the motion of the fluid domain.

The content of Section 1.3.2 is based on the published study [DDM⁺19], which focuses on the properties of the multi-stage Rosenbrock schemes for the time discretization of the Navier–Stokes equations.

1.1 The strong formulation of the incompressible Navier–Stokes equations

Let us consider a domain $\Omega \in \mathbb{R}^d$, $d = 2$ or $d = 3$, and let $\partial\Omega = \Gamma_D \cup \Gamma_N$ ($\Gamma_D \cap \Gamma_N = \emptyset$) be a partition of its boundary. In strong formulation, the incompressible Navier–Stokes equations read

$$\begin{aligned}
 \rho_f \frac{\partial \mathbf{u}}{\partial t} + \rho_f (\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla \cdot \sigma_f(\mathbf{u}, p) &= \mathbf{f} && \text{in } \Omega \times (0, T), \\
 \nabla \cdot \mathbf{u} &= 0 && \text{in } \Omega \times (0, T), \\
 \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_D \times (0, T), \\
 \sigma_f(\mathbf{u}, p) \mathbf{n} &= \mathbf{h} && \text{on } \Gamma_N \times (0, T), \\
 \mathbf{u} &= \mathbf{u}_0 && \text{for } t = 0,
 \end{aligned} \tag{1.1}$$

where $\mathbf{u} : \Omega \times (0, T) \mapsto \mathbb{R}^d$ and $p : \Omega \times (0, T) \mapsto \mathbb{R}$ are velocity and pressure of the fluid, ρ_f is the density, μ_f is the viscosity, $\varepsilon_f(\mathbf{u}) = (\nabla \mathbf{u} + \nabla \mathbf{u}^T)/2$ is the strain rate tensor, $\sigma_f(\mathbf{u}, p) = 2\mu_f \varepsilon_f(\mathbf{u}) - pI$ is the Cauchy stress tensor, $\mathbf{f} : \Omega \times (0, T) \mapsto \mathbb{R}^d$ is a forcing term, $\mathbf{g} : \Gamma_D \times (0, T) \mapsto \mathbb{R}^d$ and $\mathbf{h} : \Gamma_N \times (0, T) \mapsto \mathbb{R}^d$ are Dirichlet and Neumann data, \mathbf{n} is the normal unit vector to the boundary $\partial\Omega$, and $\mathbf{u}_0 : \Omega \mapsto \mathbb{R}^d$ is the prescribed initial condition. The first equation in Eq. (1.1) (*momentum equation*) represents the generalization of Newton’s second law of motion to continuums, and the second equation (*continuity equation*) is the incompressibility constraint.

The dynamics of the solution of the Navier–Stokes equations is described by the adimensional Reynolds number $\text{Re} = \rho_f D U / \mu_f$, where $D \in \mathbb{R}$ is a characteristic dimension of Ω (e.g., the radius of a pipe), and $U \in \mathbb{R}$ is a characteristic velocity magnitude. In particular, low values of Re correspond to laminar flows, whereas large values are typical of turbulent flows. In these limit cases the Navier–Stokes equations can be safely approximated by the (linear) Stokes equations, in which the inertial nonlinear term $(\mathbf{u} \cdot \nabla) \mathbf{u}$ is neglected, and by the nonlinear Euler equations, in which the viscous part $2\mu_f \nabla \cdot \varepsilon_f(\mathbf{u})$ is taken to be zero.

1.2 Weak formulation and discretization in space of the Navier–Stokes equations

The numerical solution of the Navier–Stokes equations by classical Galerkin methods such as the FE method entails transforming Eq. (1.1) into weak form. For details about the functional spaces we adopt in this section and in the rest of the thesis, we refer to [GR12]. Here, we limit ourselves to recall that

$$L^2(\Omega) := \left\{ \varphi : \Omega \mapsto \mathbb{R} : \int_{\Omega} |\varphi|^2 < \infty \right\},$$

1.2. Weak formulation and discretization in space of the Navier–Stokes equations

with the norm

$$\|\varphi\|_{L^2(\Omega)}^2 := \int_{\Omega} |\varphi|^2,$$

is the space of square-integrable functions defined in Ω , and that

$$H^1(\Omega) := \{\varphi \in L^2(\Omega) : \nabla\varphi \in [L^2(\Omega)]^d\},$$

is a Sobolev space with norm

$$\|\varphi\|_{H^1(\Omega)}^2 := \|\varphi\|_{L^2(\Omega)}^2 + \|\nabla\varphi\|_{L^2(\Omega)}^2.$$

We define $H_{\Gamma_D}^1(\Omega)$ as the linear manifold of functions in $H^1(\Omega)$ such that, if $\varphi \in H_{\Gamma_D}^1(\Omega)$, $\gamma_{\Gamma_D}\varphi = 0$. Moreover, we consider the notation $H_0^1(\Omega) := H_{\partial\Omega}^1(\Omega)$. Operator $\gamma_{\Gamma_D} : H^1(\Omega) \mapsto H^{1/2}(\Gamma_D)$ is called trace operator; see, e.g., [BBF13] for more information.

Let us denote $\mathcal{V} = [H^1(\Omega)]^d$, $\mathcal{V}_0 = [H_{\Gamma_D}^1(\Omega)]^d$ and $\mathcal{Q} = L^2(\Omega)$. We proceed formally by performing the scalar product of both sides of the momentum equation with a generic test function $\mathbf{v} \in \mathcal{V}_0$ and by integrating over Ω . This yields

$$\int_{\Omega} \rho_f \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} + \int_{\Omega} \rho_f [(\mathbf{u} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} - \int_{\Omega} (\nabla \cdot \sigma_f(\mathbf{u}, p)) \cdot \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v}, \quad (1.2)$$

which must hold true for every $\mathbf{v} \in \mathcal{V}_0$ and for every $t \in (0, T)$. Using Green formulae and exploiting the fact that \mathbf{v} vanishes on Γ_D , Eq. (1.2) is rewritten as

$$\int_{\Omega} \rho_f \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} + \int_{\Omega} \rho_f [(\mathbf{u} \cdot \nabla) \mathbf{u}] \cdot \mathbf{v} + \int_{\Omega} \sigma_f(\mathbf{u}, p) : \nabla \mathbf{v} = \int_{\Omega} \mathbf{f} \cdot \mathbf{v} + \int_{\Gamma_N} \mathbf{h} \cdot \mathbf{v}.$$

The continuity equation is treated in a similar fashion by multiplying both sides by a generic test function $q \in \mathcal{Q}$ and integrating over Ω .

Let us now introduce the following bilinear forms

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) &:= \langle 2\mu_f \varepsilon_f(\mathbf{u}), \nabla \mathbf{v} \rangle, \\ b(\mathbf{u}, p) &:= -\langle \nabla \cdot \mathbf{u}, p \rangle, \end{aligned}$$

and the trilinear form

$$c(\mathbf{u}, \mathbf{v}, \mathbf{w}) = \langle \rho_f [(\mathbf{u} \cdot \nabla) \mathbf{v}], \mathbf{w} \rangle,$$

for every $\mathbf{u} \in \mathcal{V}$, $\mathbf{v} \in \mathcal{V}$, $\mathbf{w} \in \mathcal{V}$, and $p \in \mathcal{Q}$. Thus, the weak formulation of the Navier–Stokes equations reads as follows.

(W1.1) *given \mathbf{f} , \mathbf{g} , \mathbf{h} , and \mathbf{u}_0 regular enough, find $\mathbf{u} \in L^2(\mathcal{V}; (0, T)) \cap C^0([L^2(\Omega)]^d; (0, T))$*

and $p \in L^2(\mathcal{Q}; (0, T))$, such that,

$$\begin{aligned} \langle \rho_t \frac{\partial \mathbf{u}}{\partial t}, \mathbf{v} \rangle + a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) &= \langle \mathbf{f}, \mathbf{v} \rangle + \langle \mathbf{h}, \mathbf{v} \rangle_{\Gamma_N} & \forall \mathbf{v} \in \mathcal{V}_0, \\ b(\mathbf{u}, q) &= 0 & \forall q \in \mathcal{Q}, \end{aligned}$$

and such that $\mathbf{u}|_{t=0} = \mathbf{u}_0$ and $\gamma_{\Gamma_D} \mathbf{u} = \mathbf{g}$.

In order to transform the infinite dimensional problem W1.1 into a finite dimensional one (which can be then numerically solved by a computer) we now consider the two finite dimensional subspaces $\mathcal{V}^h \subset \mathcal{V}$ and $\mathcal{Q}^h \subset \mathcal{Q}$. For each of these, we introduce a set of basis functions φ_i^h for $i = 1, \dots, N_u^h$ and ψ_i^h for $i = 1, \dots, N_p^h$ such that $\mathcal{V}^h = \text{span}\{\varphi_i^h\}_{i=0}^{N_u^h}$ and $\mathcal{Q}^h := \text{span}\{\psi_i^h\}_{i=0}^{N_p^h}$. The finite dimensional approximations of velocity and pressure $\mathbf{u}^h(\mathbf{x}, t) \in \mathcal{V}^h$ and $p^h(\mathbf{x}, t) \in \mathcal{Q}^h$ then read

$$\begin{aligned} \mathbf{u}^h(\mathbf{x}, t) &= \sum_{i=0}^{N_u^h} u_i^h(t) \varphi_i^h(\mathbf{x}), \\ p^h(\mathbf{x}, t) &= \sum_{i=0}^{N_p^h} p_i^h(t) \psi_i^h(\mathbf{x}). \end{aligned} \tag{1.3}$$

Coefficients $u_i^h(t)$ and $p_i^h(t)$ are typically called degrees of freedom (dofs). Here, we explicitly highlighted for the sake of clarity the dependance of each term on space and time, but this will be omitted in the following. The choice of discrete spaces \mathcal{V}^h and \mathcal{Q}^h clearly plays a crucial role in the accuracy of the approximation and it is a characteristic of the discretization method of choice. Moreover, in the case of saddle-point problems such as the Navier–Stokes equations, the quality of the discretization is critical to ensure that the well-posedness of the continuous problem—which for the three-dimensional Navier–Stokes equations is still not possible to prove in general cases—is preserved at the discrete level, as we shall see in Section 1.4. In most applications throughout the thesis we consider standard FE Lagrangian basis functions.

Let us now consider the projection of W1.1 onto the finite dimensional spaces \mathcal{V}^h and \mathcal{Q}^h . Specifically, we replace \mathbf{u} and p by their discrete counterparts defined in Eq. (1.3) and we restrict the test spaces to $\mathcal{V}_0^h = \mathcal{V}^h \cap \mathcal{V}_0$ and \mathcal{Q}^h . This leads to

$$\begin{aligned} \langle \rho_t \frac{\partial \mathbf{u}^h}{\partial t}, \mathbf{v}^h \rangle + a(\mathbf{u}^h, \mathbf{v}^h) + c(\mathbf{u}^h, \mathbf{u}^h, \mathbf{v}^h) + b(\mathbf{v}^h, p^h) &= \langle \mathbf{f}, \mathbf{v}^h \rangle + \langle \mathbf{h}, \mathbf{v}^h \rangle_{\Gamma_N}, \\ b(\mathbf{u}^h, q^h) &= 0. \end{aligned} \tag{1.4}$$

which must hold true for all $\mathbf{v}^h \in \mathcal{V}_0^h$ and $q^h \in \mathcal{Q}^h$. The discrete initial condition for \mathbf{u}^h , which we denote \mathbf{u}_0^h , can be found, e.g., as L^2 -projection of the continuous one \mathbf{u}_0 onto the space \mathcal{V}^h . Regarding the imposition of the Dirichlet data, i.e., $\gamma_{\Gamma_D} \mathbf{u} = \mathbf{g}$, this is typically enforced by setting the values of the dofs associated with the basis functions

1.3. Discretization in time of the Navier–Stokes equations

φ_i^h not vanishing on Γ_D , which is trivial to do in the common case of Lagrangian basis functions.

The fact that Eq. (1.4) must hold true for all $\mathbf{v}^h \in \mathcal{V}_0^h$ and $q^h \in \mathcal{Q}^h$ implies that the equations can be tested at the basis functions of \mathcal{V}^h and \mathcal{Q}^h , namely φ_i^h for $i = 1, \dots, N_u^h$ and ψ_i^h for $i = 1, \dots, N_p^h$. We introduce matrices

$$\begin{aligned} M_{ij}^h &= \int_{\Omega} \rho_f \varphi_j^h \cdot \varphi_i^h \in \mathbb{R}^{N_u^h \times N_u^h}, & K_{ij}^h &= a(\varphi_j^h, \varphi_i^h) \in \mathbb{R}^{N_u^h \times N_u^h}, \\ C_{ij}^h(\mathbf{u}^h) &= c(\mathbf{u}^h, \varphi_j^h, \varphi_i^h) \in \mathbb{R}^{N_u^h \times N_u^h}, & D_{ij}^h &= b(\varphi_j^h, \psi_i^h) \in \mathbb{R}^{N_p^h \times N_u^h}, \end{aligned}$$

where M^h and K^h are usually called mass and stiffness matrices, whereas C^h and D^h discretize the convective and the divergence terms, respectively. Furthermore, we define

$$\begin{aligned} \underline{\mathbf{u}}^h &= [u_0, \dots, u_{N_u^h}]^T \in \mathbb{R}^{N_u^h}, \\ \underline{\mathbf{p}}^h &= [p_0, \dots, p_{N_p^h}]^T \in \mathbb{R}^{N_p^h}, \\ \underline{\mathbf{f}}_i^h &= \int_{\Omega} \mathbf{f} \cdot \varphi_i^h + \int_{\Gamma_N} \mathbf{h} \cdot \varphi_i^h \in \mathbb{R}^{N_u^h}. \end{aligned}$$

Eq. (1.4) can now conveniently be expressed in the form of linear system as

$$\begin{bmatrix} M^h \\ \end{bmatrix} \begin{bmatrix} \dot{\underline{\mathbf{u}}}^h \\ \underline{\dot{\mathbf{p}}}^h \end{bmatrix} + \begin{bmatrix} K^h + C^h(\underline{\mathbf{u}}^h) & (D^h)^T \\ \phantom{K^h + C^h(\underline{\mathbf{u}}^h)} & D^h \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^h \\ \underline{\mathbf{p}}^h \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{f}}^h \\ \underline{\mathbf{0}} \end{bmatrix}. \quad (1.5)$$

To simplify the notation, blocks of zeros here and in the rest of the thesis are left empty. In the following, we exploit Eq. (1.3) to indicate the convective term matrix as $C^h(\underline{\mathbf{u}}^h)$, i.e., as a function of the dofs $\underline{\mathbf{u}}^h$ instead of the approximated function \mathbf{u}^h .

For a more comprehensive overview of space discretization methods of the Navier–Stokes equations, we refer to [GR12].

1.3 Discretization in time of the Navier–Stokes equations

The semi-discrete Eq. (1.5) is a Differential-Algebraic system of Equations (DAEs) of the form $H\dot{\mathbf{Y}} = \mathbf{F}(t, \mathbf{Y})$. In the particular case of Eq. (1.5), we identify $\mathbf{Y} = [(\underline{\mathbf{u}}^h)^T, (\underline{\mathbf{p}}^h)^T]^T \in \mathbb{R}^{N_u^h + N_p^h}$ and

$$H = \begin{bmatrix} M^h \\ \end{bmatrix}, \quad \mathbf{F}(t, \mathbf{Y}) = \begin{bmatrix} \underline{\mathbf{f}}^h \\ \underline{\mathbf{0}} \end{bmatrix} - \begin{bmatrix} A^h + C^h(\underline{\mathbf{u}}^h) & (D^h)^T \\ \phantom{A^h + C^h(\underline{\mathbf{u}}^h)} & D^h \end{bmatrix} \mathbf{Y}.$$

A DAE can be, in general, turned into an ODE by differentiating both sides of the equation by the independent variable t . The index of the DAE is the number of times it is necessary to differentiate it in order to get an ODE [AHS11]. The semi-discretized

Navier–Stokes equations are a DAE of index 2.

Let us introduce a series of timesteps t_0, t_1, \dots, t_{N_t} such that $t_0 = 0, t_{N_t} = T$, and $t_{k+1} = t_k + \Delta t$ for every $k = 0, \dots, N_t$; Δt is called timestep size. We denote the value of $\underline{\mathbf{Y}}$ at timestep t_k by $\underline{\mathbf{Y}}(t_k) = \underline{\mathbf{Y}}_k$. Numerical algorithms for the discretization of systems of ODEs or DAEs are classified into multi-step methods, where the solution at t_{k+1} is rewritten as function of the solutions at the previous $m \geq 1$ timesteps $\underline{\mathbf{Y}}_k, \dots, \underline{\mathbf{Y}}_{k-m+1}$, and multi-stage methods (the most notable representative being Runge–Kutta methods), where $\underline{\mathbf{Y}}_{k+1}$ is found as combination of $\underline{\mathbf{Y}}_k$ and s intermediate stages.

In this work, we focus mostly on multi-step methods, and in particular Backward Differentiation Formula (BDF) schemes [QSS10]. Nevertheless, in Section 1.3.2 we also give an example of a multi-stage method, the Rosenbrock schemes, and we discuss the main properties and limitations which lead to the choice of limiting ourselves to BDF schemes.

1.3.1 Backward Differentiation Formula schemes

Given $H\dot{\underline{\mathbf{Y}}} = \underline{\mathbf{F}}(t, \underline{\mathbf{Y}})$ and given $\underline{\mathbf{Y}}_k, \dots, \underline{\mathbf{Y}}_{k-m+1}$, the solution at timestep t_{k+1} by a generic BDF scheme with m steps takes the form

$$H\underline{\mathbf{Y}}_{k+1} = \sum_{j=1}^m \alpha_j H\underline{\mathbf{Y}}_{k-j+1} + \Delta t \beta \underline{\mathbf{F}}(t_{k+1}, \underline{\mathbf{Y}}_{k+1}), \quad (1.6)$$

where α_j for $j = 1, \dots, m$ and β depend on the number of steps m , which also coincides with the order of the method. Specifically, for $m \leq 3$ we have: if $m = 1$, $\alpha_1 = \beta = 1$, if $m = 2$, $\alpha_1 = 4/3$, $\alpha_2 = -1/3$ and $\beta = 2/3$, and if $m = 3$, $\alpha_1 = 11/18$, $\alpha_2 = -9/11$, $\alpha_3 = 2/11$ and $\beta = 6/11$. We refer to a particular BDF scheme with m steps as BDF m ; we remark that BDF1 is equivalent to the well-known Backward Euler scheme. It is important to notice that BDF schemes require a number of start-up solutions equal to $m - 1$.

Eq. (1.6) is in general nonlinear, and therefore its resolution is based on ad-hoc numerical methods; in this thesis, we employ the Newton–Raphson method. Let us rewrite Eq. (1.6) as

$$\underline{\mathbf{R}}(\underline{\mathbf{Y}}_{k+1}) := H\underline{\mathbf{Y}}_{k+1} - \sum_{j=1}^m \alpha_j H\underline{\mathbf{Y}}_{k-j+1} - \Delta t \beta \underline{\mathbf{F}}(t_{k+1}, \underline{\mathbf{Y}}_{k+1}) = \mathbf{0}. \quad (1.7)$$

Vector $\underline{\mathbf{R}}(\underline{\mathbf{Y}}_{k+1})$ is typically referred to as residual of the nonlinear equation. Solution $\underline{\mathbf{Y}}_{k+1}$ of Eq. (1.7) by the Newton–Raphson algorithm is then iteratively found, given an

1.3. Discretization in time of the Navier–Stokes equations

initial guess $\underline{\mathbf{Y}}_{k+1}^{(0)}$ (where the superscript refers to the iteration index), as

$$\underline{\mathbf{Y}}_{k+1}^{(l+1)} = \underline{\mathbf{Y}}_{k+1}^{(l)} - J_R(\underline{\mathbf{Y}}_{k+1}^{(l)})^{-1} \underline{\mathbf{R}}(\underline{\mathbf{Y}}_{k+1}^{(l)}), \quad J_R(\underline{\mathbf{Y}}_{k+1}^{(l)})_{ij} := \left. \frac{\partial \underline{\mathbf{R}}_i}{\partial \underline{\mathbf{Y}}_j} \right|_{\underline{\mathbf{Y}} = \underline{\mathbf{Y}}_{k+1}^{(l)}};$$

here, J_R is the Jacobian matrix of the residual. The Newton–Raphson method is run until a certain stopping criterion—typically based on a user-provided tolerance τ_{NR} —is satisfied. In our numerical simulations, the algorithm stops at the first iteration such that

$$\frac{\left\| \underline{\mathbf{R}}(\underline{\mathbf{Y}}_{k+1}^{(l)}) \right\|_2}{\left\| \underline{\mathbf{R}}(\underline{\mathbf{Y}}_{k+1}^{(0)}) \right\|_2} < \tau_{\text{NR}}$$

The Newton–Raphson procedure is summarized in Algorithm 1.1.

In the case of the Navier–Stokes equations, the residual explicitly takes the form

$$\begin{aligned} \underline{\mathbf{R}}_{\text{NS}}(\underline{\mathbf{u}}_{k+1}^h, \underline{\mathbf{p}}_{k+1}^h) := & \begin{bmatrix} M^h \\ \end{bmatrix} \left(\begin{bmatrix} \underline{\mathbf{u}}_{k+1}^h \\ \underline{\mathbf{p}}_{k+1}^h \end{bmatrix} - \sum_{j=1}^m \alpha_j \begin{bmatrix} \underline{\mathbf{u}}_{k-j+1}^h \\ \underline{\mathbf{p}}_{k-j+1}^h \end{bmatrix} \right) \\ & - \Delta t \beta \left(\begin{bmatrix} \underline{\mathbf{f}}_{k+1}^h \\ \underline{\mathbf{0}} \end{bmatrix} - \begin{bmatrix} A^h + C^h(\underline{\mathbf{u}}_{k+1}^h) & (D^h)^T \\ & D^h \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}_{k+1}^h \\ \underline{\mathbf{p}}_{k+1}^h \end{bmatrix} \right) \end{aligned}$$

and the Jacobian of the residual becomes

$$\begin{aligned} J_{R_{\text{NS}}}(\underline{\mathbf{u}}_{k+1}^h, \underline{\mathbf{p}}_{k+1}^h) := & \begin{bmatrix} M^h \\ \end{bmatrix} + \Delta t \beta \begin{bmatrix} A^h + J_c(\underline{\mathbf{u}}_{k+1}^h) & (D^h)^T \\ & D^h \end{bmatrix}, \\ (J_c(\underline{\mathbf{u}}^h))_{ij} := & \int_{\Omega} \rho_f [(\varphi_j^h \cdot \nabla) \mathbf{u}^h] \cdot \varphi_i^h + \int_{\Omega} \rho_f [(\mathbf{u}^h \cdot \nabla) \varphi_j^h] \cdot \varphi_i^h. \end{aligned}$$

Here, J_c is the Jacobian matrix of the convective term $[(\mathbf{u}^h \cdot \nabla) \mathbf{u}^h] \cdot \varphi_j^h$.

Remark 1.1. The computational cost associated with the resolution of nonlinear system (1.7) is considerably mitigated by evaluating the function $\underline{\mathbf{F}}(\cdot)$, in place of the solution at time t_{k+1} , at a suitable extrapolation of the form $\tilde{\underline{\mathbf{Y}}}_{k+1} = \sum_{j=1}^{\sigma} \kappa_j \underline{\mathbf{Y}}_{k-j+1}$, where σ is the order of the extrapolation. Indeed, under such modification the problem becomes linear. Extrapolations based on Newton–Gregory backward polynomials of order $\sigma = 1$, $\sigma = 2$ and $\sigma = 3$ are obtained by choosing: $\kappa_1 = 1$ for $\sigma = 1$, $\kappa_1 = 2$ and $\kappa_2 = -1$ for $\sigma = 2$, and $\kappa_1 = 3$, $\kappa_2 = -3$, and $\kappa_3 = 1$ for $\sigma = 3$. We remark that, by choosing $\sigma = 1$ in combination with BDF1, we retrieve the Forward Euler method. For more details, we refer to [CK06].

Algorithm 1.1 Newton–Raphson algorithm

Input: Initial guess $\underline{\mathbf{Y}}^{(0)}$, tolerance τ_{NR}
Output: $\underline{\mathbf{Y}}$ such that $\underline{\mathbf{R}}(\underline{\mathbf{Y}}) \approx \mathbf{0}$

- 1: set $e = 1 + \tau_{\text{NR}}$
- 2: set $l = 0$
- 3: compute $\underline{\mathbf{R}}^{(0)} = \underline{\mathbf{R}}(\underline{\mathbf{Y}}^{(0)})$
- 4: **while** $e > \tau_{\text{NR}}$ **do**
- 5: assemble the Jacobian $J_R^{(l)} = J_R(\underline{\mathbf{Y}}^{(l)})$
- 6: find $\delta\underline{\mathbf{Y}}^{(l)}$ such that $J_R^{(l)} \delta\underline{\mathbf{Y}}^{(l)} = \underline{\mathbf{R}}^{(l)}$
- 7: compute $\underline{\mathbf{Y}}^{(l+1)} = \underline{\mathbf{Y}}^{(l)} - \delta\underline{\mathbf{Y}}^{(l)}$
- 8: compute $\underline{\mathbf{R}}^{(l+1)} = \underline{\mathbf{R}}(\underline{\mathbf{Y}}^{(l+1)})$
- 9: set $e = \|\underline{\mathbf{R}}^{(l+1)}\|_2 / \|\underline{\mathbf{R}}^{(0)}\|_2$
- 10: set $l = l + 1$
- 11: set $\underline{\mathbf{Y}} = \underline{\mathbf{Y}}^{(l)}$

1.3.2 Rosenbrock schemes

As already mentioned, the most notable example of multi-stage methods are arguably Runge–Kutta schemes [HW10]. Compared to multi-step methods, Runge–Kutta schemes generally ensure better stability properties and do not require start-up solutions [SK12]. However, a major drawback of implicit Runge–Kutta schemes is that they yield nonlinear equations to be solved at each timestep. Rosenbrock methods [Ros63] offer a valid alternative to Runge–Kutta methods when dealing with nonlinear problems, as at each timestep they require the solution of a few linear systems—whose number depends on the scheme of choice—that can be interpreted as Runge–Kutta systems in which the nonlinear function is linearized with respect to the stages [BBB⁺16]. Moreover, as the matrix of each linear system is the same for all the stages, this can be assembled only once per timestep. Examples of applications of the Rosenbrock schemes to the 2D Navier–Stokes equations discretized in space with the FE method can be found in [JMR06, JR10], while in [LXLX16] and [BBC⁺15, BBC⁺16] they have been used to solve 3D problems with the Discontinuous Galerkin Method in regimes of moderate/high Reynolds numbers and turbulent flows, respectively.

Given a DAE in the form $H\dot{\underline{\mathbf{Y}}} = \underline{\mathbf{F}}(t, \underline{\mathbf{Y}})$ and given $\underline{\mathbf{Y}}_k$, a generic Rosenbrock scheme approximates the solution at time t_{k+1} as

$$\underline{\mathbf{Y}}_{k+1} = \underline{\mathbf{Y}}_k + \Delta t \sum_{i=1}^s b_j \underline{\mathbf{K}}_{i,k}, \quad (1.8)$$

where s is the number of stages and $\underline{\mathbf{K}}_{i,k}$ for $i = 1, \dots, s$ are the stages satisfying the

1.3. Discretization in time of the Navier–Stokes equations

equations

$$\begin{aligned}
 H\mathbf{K}_{i,k} &= \mathbf{F}\left(t_k + a_i\Delta t, \tilde{\mathbf{Y}}_{i,k}\right) + \Delta t J_F(t_k, \mathbf{Y}_k) \sum_{j=1}^i \gamma_{ij} \mathbf{K}_{j,k} + \Delta t \gamma_i \dot{\mathbf{F}}(t_k, \mathbf{Y}_k), \\
 \tilde{\mathbf{Y}}_{i,k} &= \mathbf{Y}_k + \Delta t \sum_{j=1}^{i-1} a_{ij} \mathbf{K}_{j,k}.
 \end{aligned}$$

Matrix $J_F(t_k, \mathbf{Y}_k)$ is the Jacobian of functional $\mathbf{F}(t, \mathbf{Y})$ computed with respect to \mathbf{Y} and evaluated at timestep (t_k, \mathbf{Y}_k) . The set of values a_{ij}, b_i, γ_{ij} with $i, j = 1, \dots, s$ has to be given and defines the specific Rosenbrock scheme, along with the order of convergence and the stability properties of the method. The conditions that must be satisfied by the coefficients to achieve a desired order of convergence can be derived by means of Butcher series, see, e.g., [NW79]. It is worth noting that a_{ij} can be arranged in a low triangular matrix, γ_{ij} can be arranged in a low triangular matrix with positive diagonal values, which we denote $\gamma := \gamma_{ii}$, and a_i and γ_i are computed, for $i = 1, \dots, s$, as

$$a_i = \sum_{j=1}^{i-1} a_{ij}, \quad \gamma_i = \sum_{j=1}^i \gamma_{ij}.$$

An interesting property of Rosenbrock methods is that they allow to build a lower-order approximation in time of the solution \mathbf{Y}_{k+1} by replacing b_i in Eq. (1.8) by a set of different coefficients \tilde{b}_i , for $i = 1, \dots, s$. The lower-order approximation is therefore computed with negligible computational effort and it can be exploited in time adaptive schemes, i.e., schemes in which the timestep size is not kept constant along the computation. Time adaptive schemes are commonly based on an estimator r_k of the difference between the high-order and the low-order solution at time t_k (e.g., a suitable norm of the difference between high- and low-order velocity fields in the case of the Navier–Stokes equations), which can be regarded as a surrogate of the local truncation error. The comparison between r_k and a user-defined target tolerance τ_{TA} indicates if the timestep size should be increased or decreased. Possible choices for the evolution law of the timestep size for a method of order q are the elementary local control algorithm [Söd02]

$$\Delta t_{k+1} = \rho_f \Delta t_k \left(\frac{\tau_{\text{TA}}}{r_k} \right)^{1/q},$$

and the proportional integral (PI) controller [GLS88]

$$\Delta t_{k+1} = \rho_f \frac{\Delta t_k^2}{\Delta t_{k-1}} \left(\frac{\tau_{\text{TA}}}{r_k} \frac{r_{k-1}}{r_k} \right)^{1/q}. \quad (1.9)$$

In the formulas above, $\rho_f \in (0, 1]$ is a safety factor.

The two primary reasons that lead us to focus—in the numerical simulations of the

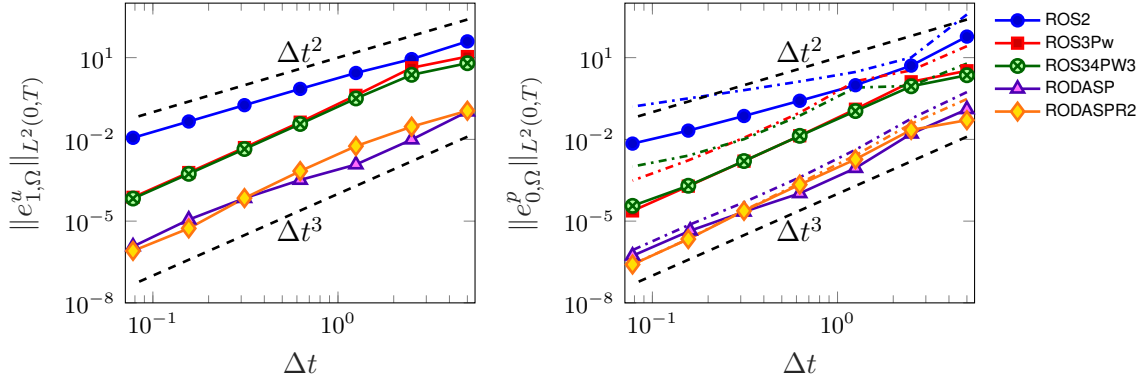


Figure 1.1 – Convergence in time for various Rosenbrock schemes. The solid lines in the convergence plots for the pressure refer to the pressure computed with the correction at the end of the timestep by solving Eq. (1.10), while the dot-dashed lines are obtained without correction.

following chapters—on multi-step methods rather than Rosenbrock schemes are: (i) order-reduction and (ii) the difficulties related to the convergence of the linear system solver in cases of moderate Reynolds numbers. Both these issues are addressed in [DDM⁺19].

Order-reduction. Order-reduction is a well-documented characteristic of these schemes (see, e.g., [LT08, MNL⁺18, HW10]) and it particularly affects the pressure component in the Navier–Stokes equations. For this reason, in [DDM⁺19] we propose to compute an inexpensive correction to the pressure at the end of each timestep by following an approach similar to that of the Pressure Poisson Equation (PPE) [SR11]. In order to derive such correction, we denote $p_c = p + \delta p$ the corrected pressure and we apply the divergence operator to the momentum equation in Eq. (1.1)

$$\nabla \cdot \left(\rho_f \frac{\partial \mathbf{u}}{\partial t} - 2\mu_f \nabla \cdot \varepsilon_f(\mathbf{u}) + \rho_f(\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p_c \right) = \nabla \cdot \mathbf{f}.$$

This yields, by exploiting the incompressibility constraint,

$$-\Delta \delta p = \nabla \cdot [(\mathbf{u} \cdot \nabla) \mathbf{u} + \nabla p - \mathbf{f}] \quad \text{in } \Omega, \quad (1.10)$$

which is solved at each timestep t_k given the solution (\mathbf{u}_k^h, p_k^h) by a Rosenbrock method.

Remark 1.2. The boundary conditions complementing Eq. (1.10) depend on the ones to be imposed on the Navier–Stokes equations. In the numerical results that follow, we consider Dirichlet boundary conditions on $\partial\Omega$, which reflect in $\nabla \delta p \cdot \mathbf{n} = 0$ on $\partial\Omega$. Since these boundary conditions do not allow to determine the pressure correction uniquely, in the implementation we arbitrarily fix one dof of δp to zero. The convergence rates we present are then obtained by setting the mean value of p_c equal to the exact one.

1.3. Discretization in time of the Navier–Stokes equations

Table 1.1 – Number of stages, theoretical and measured orders of the selected Rosenbrock schemes. The numerical rate of convergence are approximated as slopes of the linear regression lines computed from the errors in Fig. 1.1.

method	stages	order	measured orders		
			velocity	pressure	corrected pressure
ROS2	2	2	1.94	1.64	2.10
ROS3Pw	3	3	2.98	2.79	2.97
ROS34PW3	4	3	2.85	2.18	2.79
RODASP	6	4	2.55	3.22	2.95
RODASPR2	6	4	2.94	3.40	3.06

In Fig. 1.1, we show the rate of convergence of the errors

$$\|e_{1,\Omega}^u\|_{L^2(0,T)}^2 := \Delta t \sum_{i=1}^{N_t} \|\mathbf{u}^h(t_i) - \mathbf{u}^{\text{ex}}(t_i)\|_{H^1(\Omega)}^2,$$

$$\|e_{0,\Omega}^p\|_{L^2(0,T)}^2 := \Delta t \sum_{i=1}^{N_t} \|p^h(t_i) - p^{\text{ex}}(t_i)\|_{L^2(\Omega)}^2,$$

obtained by applying the ROS2 scheme [VSBH99], the ROS3Pw scheme [RA05], the ROS34PW3 scheme [JMR06], the RODASP scheme [Ste95] and the RODASPR2 scheme [Ran15] to a test case with known analytical solution $(\mathbf{u}^{\text{ex}}, p^{\text{ex}})$; the coefficients of each scheme are reported in the cited references. We refer the reader to [DDM⁺19] for the details on the specific problem. Moreover, in Table 1.1 we show the theoretical order of convergence of each method and the measured orders of convergence both for the errors in \mathbf{u} and p . The rates are obtained as slopes of the regression lines passing through the points in Fig. 1.1. For ROS2, ROS3Pw, and ROS34PW3 the velocity converges with the expected orders. For RODASP and RODASPR2 the convergence rate is lower than the theoretical one; however, as discussed in [Ran15], these methods are not B_{PR} -consistent of order 4, and the theoretical convergence rate is not guaranteed for stiff problems; we refer the reader to [Ran16] for the definition of B_{PR} -consistency. Order reduction occurs for the pressure with ROS2, RODASP, RODASPR2 and ROS34PW3. We remark that for all the schemes the corrected pressure achieves smaller errors than the uncorrected ones. Moreover, the theoretical orders of convergence are almost completely recovered also for ROS2 and ROS34PW3, whereas for RODASP and RODASPR2 the rates remain lower than optimal because, as already noted, for these schemes order reduction affects also the velocity.

Convergence of linear solver. In [DDM⁺19], we apply the Rosenbrock method to the solution of the 3D benchmark originally proposed by Schäfer and Turek [ST96] of flow past a cylinder; the domain is characterized in Fig. 1.2. The inflow is time dependent

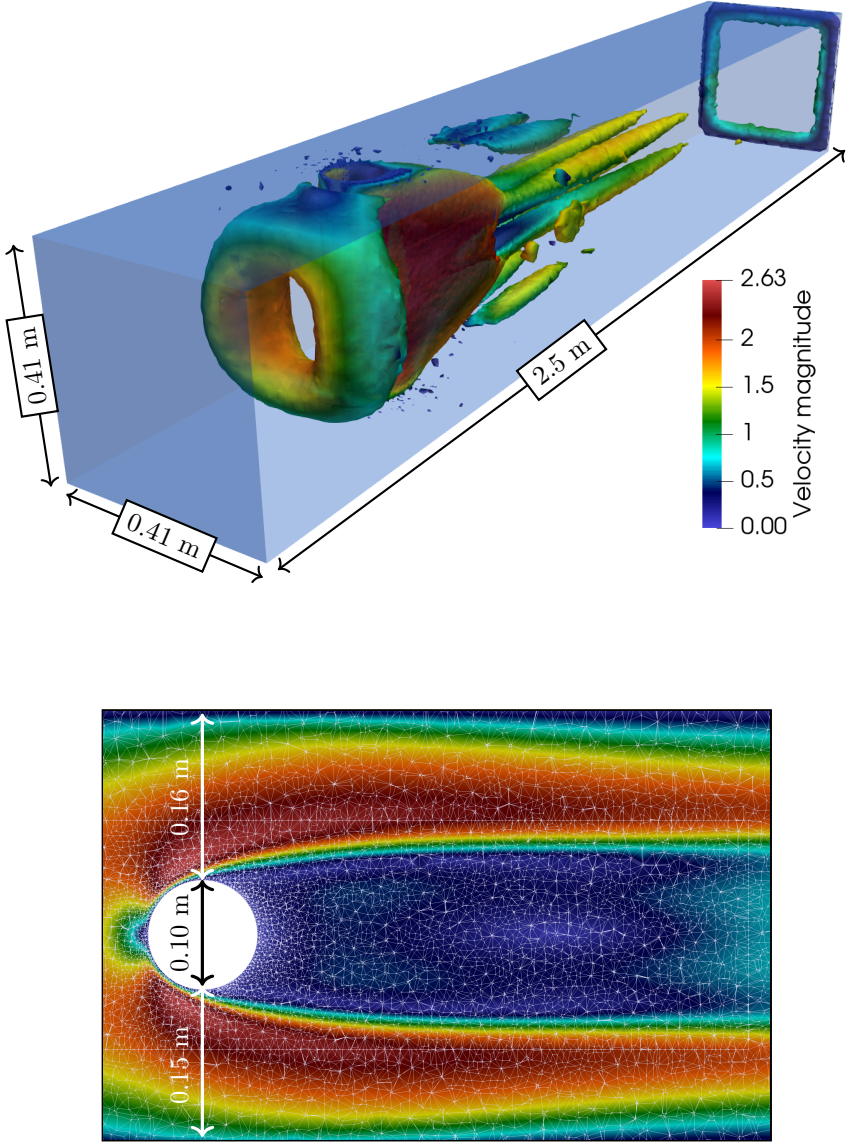


Figure 1.2 – On the top, iso-contour of the Q-criterion with $Q_{\text{iso}} = 6$ of the solution of the flow past a cylinder obtained using the ROS2 scheme with $\Delta t = 5 \cdot 10^{-3}$ s at $t = 4$ s. On the bottom, magnitude of the velocity field of the same solution at $t = 4$ s on a plane perpendicular to the cylinder and passing at half the height of the domain, with detail of the computational mesh.

1.3. Discretization in time of the Navier–Stokes equations

and follows the rule

$$\mathbf{u}_{\text{in}}(0, y, z, t) = 16U_m yz(H - y)(H - z) \sin(\pi t/8)/H^4 [1, 0, 0]^T, \quad (1.11)$$

where $t \in (0, 8)$ s, $U_m = 2.25$ m/s and $H = 0.41$ m is the height and width of the domain. We refer the reader to the original paper [ST96] for details about the geometry of the problem and the fluid parameters. The inflow prescribed by Eq. (1.11) leads to a Reynolds number which varies between 0 and 100. As shown in Fig. 1.2 (top), the solution of the problem presents flow detachment due to the presence of the obstacle and develops complex vorticity patterns behind the cylinder. The same test case has been considered in, e.g., [Joh06] and [BMT12] with similar results. We employ a fine mesh refined around the cylinder (see Fig. 1.2, bottom) and composed of 6'718'220 dofs for the velocity and 287'039 for the pressure, for a total of 7'005'259 dofs. We set homogeneous essential boundary conditions on the lateral walls of the domain and on the cylinder, nonhomogeneous essential boundary conditions on the inlet and homogeneous natural boundary conditions on the outlet.

Fig. 1.3 shows, in red solid lines, the drag and lift coefficients computed with ROS2 and $\Delta t = 5 \cdot 10^{-3}$ s, and, in black dashed lines, the reference solution reported in [Joh06]; this is obtained with a mesh with 6'052'800 and 983'040 dofs for velocity and pressure, respectively, with the Crank-Nicolson scheme, and with $\Delta t = 10^{-2}$ s. We recall that the drag and lift coefficients are computed as

$$c_D = -\frac{1}{q_\infty DH} \oint_{\Gamma_c} [\sigma_f(\mathbf{u}, p)\mathbf{n}] \cdot \mathbf{v}_\infty,$$

$$c_L = \frac{1}{q_\infty DH} \oint_{\Gamma_c} [\sigma_f(\mathbf{u}, p)\mathbf{n}] \cdot \mathbf{n}_\infty,$$

where Γ_c is the boundary of the cylinder, \mathbf{n} is the normal unit vector to Γ_c , $q_\infty = 0.5 U_\infty^2$, $U_\infty = 1$ m/s, and \mathbf{v}_∞ and \mathbf{n}_∞ are unit vectors parallel and perpendicular to the direction of the flow at the inlet, respectively.

The curves for the drag and lift coefficients computed using the two-stages ROS2 scheme are in agreement with the benchmark. The maximum and minimum values for c_D and c_L are: $c_D^{\max} = 3.2942$, $c_D^{\min} = -1.7933 \cdot 10^{-1}$, $c_L^{\max} = 2.7697 \cdot 10^{-3}$, $c_L^{\min} = -1.0967 \cdot 10^{-2}$; these quantities are close to the ones reported in [Joh06], in which $c_D^{\max} = 3.2968$ and $c_L^{\min} = -1.10 \cdot 10^{-2}$, and within the confidence intervals for the exact solution proposed in the original paper [ST96].

Even though the results in Fig. 1.3 are promising, considering schemes with more than two steps for this test case is a challenging task. As discussed in [DDM⁺19], the linear solver based on the GMRES algorithm performs poorly in the temporal regions of the simulation where the Reynolds number reaches its largest values, i.e., at around $t = 4$ s by choosing the magnitude of the imposed inflow profile as characteristic velocity. As

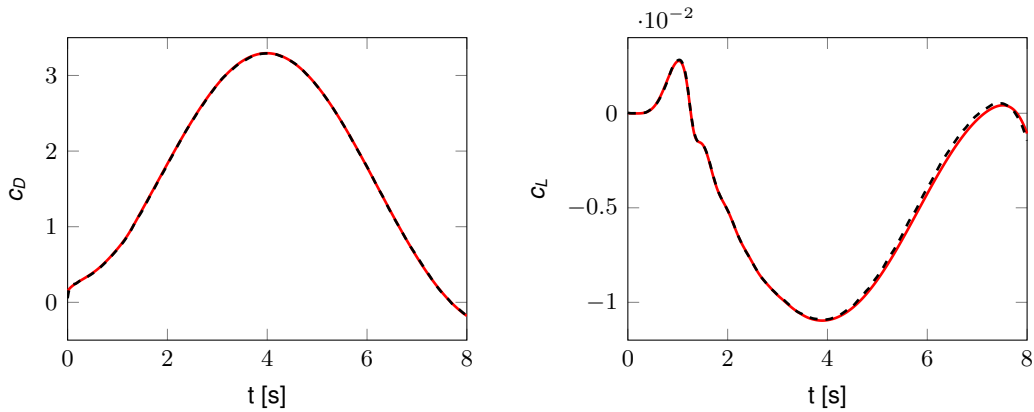


Figure 1.3 – Drag and lift coefficients. The red solid line refers to c_D (left) and c_L (right) obtained with the ROS2 scheme with fixed $\Delta t = 5 \cdot 10^{-3}$ s. The black dashed line refers to the reference solution reported in [Joh06].

an example, in Fig. 1.4 we show the total GMRES iterations in the region $(0, 2)$ s when solving the problem using ROS3Pw, which features three stages, and when applying time adaptivity according to the PI-controller law for the timestep size (see Eq. (1.9)) with the estimator

$$r_k = \left\| \mathbf{u}_{k,\text{high}}^h - \mathbf{u}_{k,\text{low}}^h \right\|_{H^1(\Omega)},$$

where $\mathbf{u}_{k,\text{high}}^h$ and $\mathbf{u}_{k,\text{low}}^h$ are the high-order and the low-order solutions respectively. When the Reynolds number of the flow increases—which leads to the development of complex vorticity patterns downstream of the cylinder—the performance of the linear solver degrades significantly; for larger Δt it stops converging altogether. For this reason, the employment of high-order Rosenbrock schemes for this challenging test case is inefficient. The difficulties related to the convergence of the linear solver prevent us from evaluating the quality of the approximation even for larger tolerances and, consequently, larger timestep sizes. This is particularly unfortunate especially if put in relation with the results in Table 1.2, which show great accuracy in the approximation of the maximum value attained by c_L when employing ROS2 and ROS3Pw, and which therefore illustrate the potential for time adaptivity to lower the computational burden of these simulations greatly.

1.4 Elements of the theory of saddle-point problems

Problem W1.1 features a saddle-point structure that requires particular care from the mathematical perspective. Although the solvability of the Navier–Stokes equations is still an open issue in the general three-dimensional case, the theory of saddle-point problems is fundamental in providing *necessary* conditions for the well-posedness of the discrete

1.4. Elements of the theory of saddle-point problems

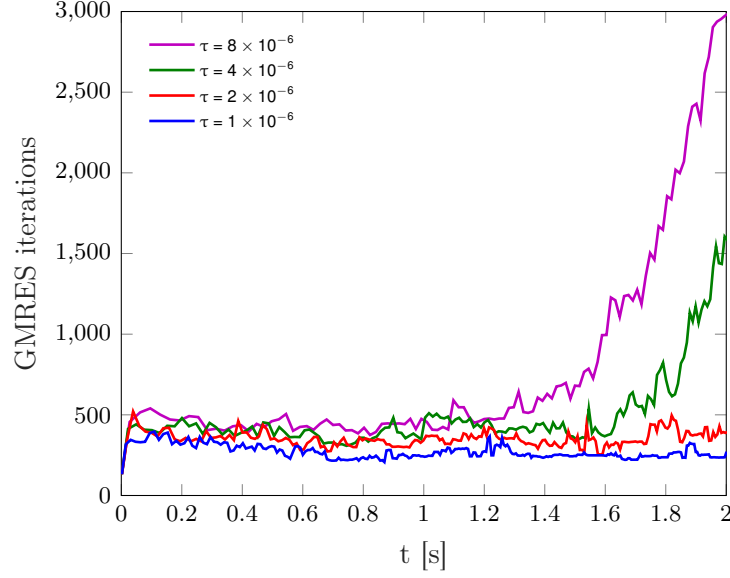


Figure 1.4 – Total GMRES iterations (i.e., sum of the iterations of each stage) when solving the Navier–Stokes equations by using ROS3Pw with adaptive timesteps. The tolerance of the linear solver is set to 10^{-11} with stopping criterion based on the residual norm relative to the initial one.

	τ	timesteps	Δt_n^{\max}	$\overline{\Delta t_n}$	c_L^{\max}
ROS2	$4 \cdot 10^{-3}$	62	$4.46 \cdot 10^{-2}$	$3.26 \cdot 10^{-2}$	$2.75330 \cdot 10^{-3}$
	$2 \cdot 10^{-3}$	87	$3.17 \cdot 10^{-2}$	$2.30 \cdot 10^{-2}$	$2.76385 \cdot 10^{-3}$
	$1 \cdot 10^{-3}$	123	$2.25 \cdot 10^{-2}$	$1.63 \cdot 10^{-2}$	$2.76734 \cdot 10^{-3}$
	$5 \cdot 10^{-4}$	174	$1.60 \cdot 10^{-2}$	$1.15 \cdot 10^{-2}$	$2.76841 \cdot 10^{-3}$
ROS3Pw	$8 \cdot 10^{-6}$	108	$3.46 \cdot 10^{-2}$	$1.86 \cdot 10^{-2}$	$2.76856 \cdot 10^{-3}$
	$4 \cdot 10^{-6}$	150	$2.69 \cdot 10^{-2}$	$1.34 \cdot 10^{-2}$	$2.76950 \cdot 10^{-3}$
	$2 \cdot 10^{-6}$	210	$2.09 \cdot 10^{-2}$	$9.54 \cdot 10^{-3}$	$2.76944 \cdot 10^{-3}$
	$1 \cdot 10^{-6}$	295	$1.63 \cdot 10^{-2}$	$6.79 \cdot 10^{-3}$	$2.77005 \cdot 10^{-3}$

Table 1.2 – Number of timesteps, maximum timestep size Δt_n^{\max} , average timestep size $\overline{\Delta t_n}$ and c_L^{\max} as a function of the tolerance τ of the time adaptivity scheme.

formulation. For this reason, and because other saddle-point problems are treated in the subsequent chapters of this thesis, we address here the main results concerning well-posedness and convergence estimates.

1.4.1 Existence, uniqueness, stability and convergence results

In this section, we provide classical results on some fundamental properties of saddle-point problems. A comprehensive overview from the functional and algebraic point of view is found, e.g., in [Bre74, BB90, BBF13].

We start by considering two generic Hilbert spaces \mathcal{V} and \mathcal{Q} and the two bilinear forms $a(u, v)$ and $b(u, p)$, which are defined for all $u \in \mathcal{V}$, $v \in \mathcal{V}$ and $p \in \mathcal{Q}$. We remark that even though the notation is the same as the one adopted in W1.1, for the time being, we are not yet referring to any specific saddle-point problems, and these spaces and bilinear forms are as general as possible. Let us introduce the concept of kernel of $b(\cdot, \cdot)$, formally

$$\mathcal{K} = \{v \in \mathcal{V} : b(v, q) = 0 \quad \forall q \in \mathcal{Q}\},$$

and let us consider the following set of hypotheses. Firstly, we assume that $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$ are continuous, i.e., there exist two positive constants κ_a and κ_b such that

$$\begin{aligned} a(u, v) &\leq \kappa_a \|u\|_{\mathcal{V}} \|v\|_{\mathcal{V}} \quad \forall u \in \mathcal{V}, \forall v \in \mathcal{V}, \\ b(u, p) &\leq \kappa_b \|u\|_{\mathcal{V}} \|p\|_{\mathcal{Q}} \quad \forall u \in \mathcal{V}, \forall p \in \mathcal{Q}. \end{aligned} \tag{1.12}$$

Secondly, we assume that $a(\cdot, \cdot)$ is coercive on \mathcal{K} , i.e., there exist a positive constant α such that

$$a(u, u) \geq \alpha \|u\|_{\mathcal{V}}^2 \quad \forall u \in \mathcal{K}. \tag{1.13}$$

Lastly, we introduce the Ladyshenskaja–Babuška–Brezzi condition, which is commonly referred to as the inf-sup condition. We assume that there exist a positive constant β such that

$$\inf_{q \in \mathcal{Q} \setminus \{0\}} \sup_{v \in \mathcal{V} \setminus \{0\}} \frac{b(v, q)}{\|v\|_{\mathcal{V}} \|q\|_{\mathcal{Q}}} \geq \beta. \tag{1.14}$$

We are now ready to formulate the following theorem.

Theorem 1.1. [BB90] *Assume conditions (1.12), (1.13) and (1.14). Given two Hilbert spaces \mathcal{V} and \mathcal{Q} and $f \in \mathcal{V}'$ and $g \in \mathcal{Q}'$, the problem*

$$\begin{aligned} a(u, v) + b(v, p) &= \langle f, v \rangle \quad \forall v \in \mathcal{V}, \\ b(u, q) &= \langle g, q \rangle \quad \forall q \in \mathcal{Q}, \end{aligned} \tag{1.15}$$

admits a unique solution $(y, p) \in \mathcal{V} \times \mathcal{Q}$.

1.4. Elements of the theory of saddle-point problems

Remark 1.3. Variables u and p are often called primal and dual variables, the reason being that the second equation in Eq. (1.15) usually derives from the imposition of a constraint on u through the Lagrange multiplier p . In the case of the incompressible Navier–Stokes equations, the continuity equation constrains the velocity to be divergence-free.

As we shall see in the following lemma, the continuity, coercivity and inf-sup constants play an important role not only in the well-posedness of the saddle-point problem but also in its stability.

Lemma 1.1 ([BBF13]). *Under the assumptions of Theorem 1.1, the following stability estimates hold*

$$\begin{aligned} \|u\|_{\mathcal{V}} &\leq \frac{1}{\alpha} \|f\|_{\mathcal{V}'} + \frac{1}{\beta} \left(1 + \frac{\kappa_a}{\alpha}\right) \|g\|_{\mathcal{Q}'}, \\ \|p\|_{\mathcal{Q}} &\leq \frac{1}{\beta} \left(1 + \frac{\kappa_a}{\alpha}\right) \|f\|_{\mathcal{V}'} + \frac{\kappa_a}{\beta^2} \left(1 + \frac{\kappa_a}{\alpha}\right) \|g\|_{\mathcal{Q}'}. \end{aligned}$$

Proof. Let us consider $u \in \mathcal{V}$ and its decomposition $u = u_0 + u_g$, such that $u_0 \in \mathcal{K}$ and $u_g \in \mathcal{K}^\perp$. Moreover, let us note that, for all $q \in \mathcal{Q}$, $b(u, q) = b(u_0 + u_g, q) = b(u_g, q) = \langle g, q \rangle$; therefore, owing to the inf-sup condition,

$$\|g\|_{\mathcal{Q}'} \|q\|_{\mathcal{Q}} \geq \langle g, q \rangle = b(u_g, q) \geq \beta \|u_g\|_{\mathcal{V}} \|q\|_{\mathcal{Q}}$$

and equivalently $\|u_g\|_{\mathcal{V}} \leq 1/\beta \|g\|_{\mathcal{Q}'}$. By evaluating the first equation in (1.15) at $u_0 \in \mathcal{K}$, we obtain

$$a(u_0 + u_g, u_0) + b(u_0, p) = a(u_0, u_0) + a(u_g, u_0) = \langle f, u_0 \rangle.$$

Hence, by coercivity,

$$\alpha \|u_0\|_{\mathcal{V}}^2 \leq \|f\|_{\mathcal{V}'} \|u_0\|_{\mathcal{V}} + \kappa_a \|u_g\|_{\mathcal{V}} \|u_0\|_{\mathcal{V}}$$

and the first condition in the lemma follows

$$\|u\|_{\mathcal{V}} = \|u_0 + u_g\|_{\mathcal{V}} \leq \|u_0\|_{\mathcal{V}} + \|u_g\|_{\mathcal{V}} \leq \frac{1}{\alpha} \|f\|_{\mathcal{V}'} + \frac{1}{\beta} \left(1 + \frac{\kappa_a}{\alpha}\right) \|g\|_{\mathcal{Q}'}. \quad (1.16)$$

The second condition is a direct consequence of the inf-sup condition. We have

$$\beta \|p\|_{\mathcal{Q}} \leq \sup_{v \in \mathcal{V}} \frac{b(v, p)}{\|v\|_{\mathcal{V}}} = \sup_{v \in \mathcal{V}} \frac{\langle f, v \rangle - a(u, v)}{\|v\|_{\mathcal{V}}} \leq \|f\|_{\mathcal{V}'} + \kappa_a \|u\|_{\mathcal{V}},$$

Thus, by using Eq. (1.16),

$$\|p\|_{\mathcal{Q}} \leq \frac{1}{\beta} \|f\|_{\mathcal{V}'} + \frac{\kappa_a}{\beta} \|u\|_{\mathcal{V}} \leq \frac{1}{\beta} \left(1 + \frac{\kappa_a}{\alpha}\right) \|f\|_{\mathcal{V}'} + \frac{\kappa_a}{\beta^2} \left(1 + \frac{\kappa_a}{\alpha}\right) \|g\|_{\mathcal{Q}'}. \quad \square$$

Chapter 1. Mathematical models for hemodynamics

Let $\mathcal{V}^h \subset \mathcal{V}$ and $\mathcal{Q}^h \subset \mathcal{Q}$ be finite dimensional subspaces of \mathcal{V} and \mathcal{Q} , respectively (to fix our ideas, \mathcal{V}^h and \mathcal{Q}^h are FE spaces). The well-posedness of the saddle-point problem

$$\begin{aligned} a(u^h, v^h) + b(v^h, p^h) &= \langle f^h, v^h \rangle & \forall v^h \in \mathcal{V}^h, \\ b(u^h, q^h) &= \langle g^h, q^h \rangle & \forall q^h \in \mathcal{Q}^h \end{aligned} \quad (1.17)$$

is subject to the same conditions of Theorem 1.1. In other words, the existence and uniqueness of the solution in the continuous setting is in general not sufficient to guarantee that these properties are passed on to the discrete counterparts. As we shall see in Section 1.4.3, in the FE context this reflects in the need to appropriately choose the basis functions or to consider ad-hoc stabilization strategies.

The hypotheses of Theorem 1.1 being satisfied in the discrete setting, it is of crucial importance to take into consideration estimates for the rate of convergence of the discrete solutions to the continuous one. We define the spaces

$$\begin{aligned} \mathcal{K}^h &= \{v^h \in \mathcal{V}^h : b(v^h, q^h) = 0 \quad \forall q^h \in \mathcal{Q}^h\}, \\ \mathcal{V}_g^h &= \{v^h \in \mathcal{V}^h : b(v^h, q^h) = \langle g, q^h \rangle \quad \forall q^h \in \mathcal{Q}^h\}. \end{aligned}$$

We remark that $\mathcal{K}^h \not\subset \mathcal{K}$, because if $v \in \mathcal{K}^h$ there could be $q \in \mathcal{Q} \setminus \mathcal{Q}^h$ such that $b(v^h, q) \neq 0$. The following theorem provides a sharp bound for the approximation error. The proof can be found, e.g., in [Qua14, Theorem 16.6] and in [GR12].

Theorem 1.2. *Let the assumptions of Theorem 1.1 be satisfied in the continuous setting. Moreover, there exist $\alpha^h > 0$ such that $a(u^h, u^h) \geq \alpha^h \|u^h\|_{\mathcal{V}}^2$ for all $u^h \in \mathcal{K}^h$ and β^h such that*

$$\inf_{q^h \in \mathcal{Q}^h \setminus \{0\}} \sup_{v^h \in \mathcal{V}^h \setminus \{0\}} \frac{b(v^h, q^h)}{\|v^h\|_{\mathcal{V}^h} \|q^h\|_{\mathcal{Q}^h}} \geq \beta^h.$$

Then the solution (u, p) of Eq. (1.15) and the solution (u^h, p^h) of Eq. (1.17) satisfy the following error estimates

$$\begin{aligned} \|u - u^h\|_{\mathcal{V}} &\leq \left(1 + \frac{\kappa_a}{\alpha^h}\right) \inf_{v^h \in \mathcal{V}_g^h} \|u - v^h\|_{\mathcal{V}} + \frac{\kappa_b}{\alpha^h} \inf_{q^h \in \mathcal{Q}^h} \|p - q^h\|_{\mathcal{Q}} \\ \|p - p^h\|_{\mathcal{Q}} &\leq \frac{\kappa_a}{\beta^h} \left(1 + \frac{\kappa_a}{\alpha^h}\right) \inf_{v^h \in \mathcal{V}_g^h} \|u - v^h\|_{\mathcal{V}} + \left(1 + \frac{\kappa_b}{\beta^h} + \frac{\kappa_a \kappa_b}{\alpha^h \beta^h}\right) \inf_{q^h \in \mathcal{Q}^h} \|p - q^h\|_{\mathcal{Q}^h}. \end{aligned}$$

Moreover, the following error estimate holds

$$\inf_{v^h \in \mathcal{V}_g^h} \|u - v^h\|_{\mathcal{V}} \leq \left(1 + \frac{\kappa_b}{\beta^h}\right) \inf_{v^h \in \mathcal{V}^h} \|u - v^h\|_{\mathcal{V}}.$$

Theorem 1.2 shows that the rate of convergence of both the primal and dual variables are dependent on the continuity, coercivity and inf-sup constants. In particular, there exists an inverse relationship between the approximation errors and α^h and β^h . While the coercivity constant is typically easy to control, the inf-sup constant is more delicate and, therefore, in the next section we present a numerical method to compute it.

1.4.2 Numerical estimation of the inf-sup constant

Let us suppose that $X_{\mathcal{V}}^h$ and $X_{\mathcal{Q}}^h$ are norm matrices such that $\|v^h\|_{\mathcal{V}}^2 = (\underline{\mathbf{v}}^h)^T X_{\mathcal{V}}^h \underline{\mathbf{v}}^h$ and $\|p^h\|_{\mathcal{Q}}^2 = (\underline{\mathbf{q}}^h)^T X_{\mathcal{Q}}^h \underline{\mathbf{q}}^h$, and B^h is a matrix such that $b(v^h, q^h) = (\underline{\mathbf{q}}^h)^T B^h \underline{\mathbf{v}}^h$, for every $v^h \in \mathcal{V}^h$ and every $p^h \in \mathcal{Q}^h$. In the previous expressions, we denoted $\underline{\mathbf{v}}^h$ and $\underline{\mathbf{q}}^h$ the vectors of dofs of v^h and q^h . Then, we have

$$\begin{aligned} \tilde{\beta}^h &= \inf_{q^h \in \mathcal{Q}^h \setminus \{0\}} \sup_{v^h \in \mathcal{V}^h \setminus \{0\}} \frac{b(v^h, q^h)}{\|v^h\|_{\mathcal{V}} \|q^h\|_{\mathcal{Q}^h}} = \inf_{\underline{\mathbf{q}}^h \neq \mathbf{0}} \sup_{\underline{\mathbf{v}}^h \neq \mathbf{0}} \frac{(\underline{\mathbf{q}}^h)^T B^h \underline{\mathbf{v}}^h}{\sqrt{(\underline{\mathbf{v}}^h)^T X_{\mathcal{V}}^h \underline{\mathbf{v}}^h} \sqrt{(\underline{\mathbf{q}}^h)^T X_{\mathcal{Q}}^h \underline{\mathbf{q}}^h}} \\ &= \inf_{\underline{\mathbf{q}}^h \neq \mathbf{0}} \frac{1}{\sqrt{(\underline{\mathbf{q}}^h)^T X_{\mathcal{Q}}^h \underline{\mathbf{q}}^h}} \sup_{\substack{\underline{\mathbf{w}}^h \neq \mathbf{0} \\ \|\underline{\mathbf{w}}^h\|=1}} (\underline{\mathbf{w}}^h)^T Z^h \underline{\mathbf{q}}^h = \inf_{\underline{\mathbf{q}}^h \neq \mathbf{0}} \frac{\sqrt{(Z^h \underline{\mathbf{q}}^h)^T Z^h \underline{\mathbf{q}}^h}}{\sqrt{(\underline{\mathbf{q}}^h)^T X_{\mathcal{Q}}^h \underline{\mathbf{q}}^h}}, \end{aligned} \quad (1.18)$$

where $Z^h = (X_{\mathcal{V}}^h)^{-1/2} (B^h)^T$. We remark that the last equivalence in Eq. (1.18) comes from the fact that the unitary vector $\underline{\mathbf{w}}^h$ maximizing its scalar product with $\underline{\mathbf{y}}^h = Z^h \underline{\mathbf{q}}^h$ is $\underline{\mathbf{w}}^h = \underline{\mathbf{y}}^h / \|\underline{\mathbf{y}}^h\|$ by the Cauchy–Schwarz inequality. Introducing now the following generalized eigenvalue problem

$$\begin{bmatrix} X_{\mathcal{V}}^h & (B^h)^T \\ B^h & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{v}}^h \\ \underline{\mathbf{q}}^h \end{bmatrix} = -\sigma^h \begin{bmatrix} \underline{\mathbf{v}}^h \\ \underline{\mathbf{q}}^h \end{bmatrix}, \quad (1.19)$$

and recognizing that we have

$$B^h (X_{\mathcal{V}}^h)^{-1} (B^h)^T \underline{\mathbf{q}}^h = \sigma^h X_{\mathcal{Q}}^h \underline{\mathbf{q}}^h \quad \Rightarrow \quad \sigma^h = \frac{(Z^h \underline{\mathbf{q}}^h)^T Z^h \underline{\mathbf{q}}^h}{(\underline{\mathbf{q}}^h)^T X_{\mathcal{Q}}^h \underline{\mathbf{q}}^h},$$

we conclude that $\tilde{\beta}^h$ can be computed as the square root of the minimum eigenvalue of Eq. (1.19), i.e., $\tilde{\beta}^h = \sqrt{\sigma_{\min}^h}$.

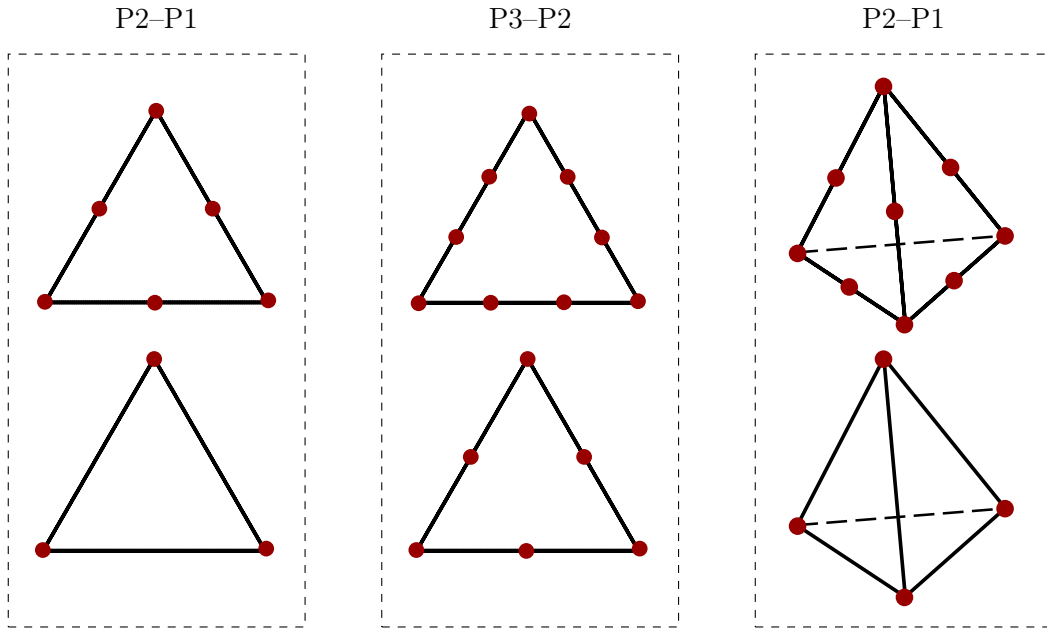


Figure 1.5 – Some generalized Taylor–Hood elements in two and three dimensions. The velocity and pressure elements are on the top and on the bottom, respectively. For each element, the position of the mesh node is marked with red dots.

1.4.3 The case of the Stokes equations

Because of the difficulties related to the treatment of the unsteady nonlinear Navier–Stokes equations, we consider for simplicity the steady Stokes equations, namely

$$\begin{aligned} -\nabla \cdot \sigma_f(\mathbf{u}, p) &= \mathbf{f} && \text{in } \Omega, \\ \nabla \cdot \mathbf{u} &= 0 && \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_D, \\ \sigma_f(\mathbf{u}, p)\mathbf{n} &= \mathbf{h} && \text{on } \Gamma_N. \end{aligned}$$

We also consider $\mathbf{g} = \mathbf{0}$ and $\mathbf{h} = \mathbf{0}$. By employing the same forms appearing in W1.1, it is trivial to derive the weak formulation of the Stokes equations.

(W1.2) *given \mathbf{f} regular enough, find $(\mathbf{u}, p) \in \mathcal{V}_0 \times \mathcal{Q}$, such that,*

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) &= \langle \mathbf{f}, \mathbf{v} \rangle && \forall \mathbf{v} \in \mathcal{V}_0, \\ b(\mathbf{u}, q) &= 0 && \forall q \in \mathcal{Q}. \end{aligned}$$

This weak formulation is clearly a particular instance of the general saddle-point problem (1.15). One of the most popular approaches to ensure the well-posedness of the Stokes equations in the discrete setting is by resorting to the generalized Taylor–Hood $P(k)$ – $P(k-$

1.5. Stabilization of the Navier–Stokes equations by Variational Multiscale modeling

1) elements [TH73] with continuous pressure. We remark that, with the notation Pk , we denote piecewise polynomial basis functions (typically, Lagrangian basis functions) of degree k . In Fig. 1.5 we show some popular generalized Taylor–Hoods elements on triangles and tetrahedra; these are also the choices that are employed in the numerical simulations presented in this thesis. Proofs of convergence for these type of elements can be found, e.g., in [BP79, Ver84, Bof94, Bof97].

As we shall see in the next section, another viable possibility in the context of the (Navier)–Stokes equation is to consider stabilized formulations.

1.5 Stabilization of the Navier–Stokes equations by Variational Multiscale modeling

We present here an example of a stabilized formulation of the Navier–Stokes equations. In particular, we consider the Variational Multiscale (VMS) approach [HSF18, BCC⁺07], which is employed in the numerical results presented in Section 2.7 and in the reference solution in Section 5.3; a simplified version of the stabilization (Streamline–Upwind Petrov–Galerkin, SUPG) is employed in Section 5.4.

The VMS stabilization for the Navier–Stokes equations is based on decomposing the solution into a coarse and a fine component. The former is discretized by means of the FE method, while the latter needs to be modeled. Specifically, let us consider the following decompositions

$$\begin{aligned}\mathcal{V} &= \mathcal{V}^h \oplus \mathcal{V}', \\ \mathcal{Q} &= \mathcal{Q}^h \oplus \mathcal{Q}',\end{aligned}$$

where \mathcal{V} and \mathcal{Q} are the spaces for velocity and pressure appearing in W1.1. As already anticipated, the continuous spaces are imagined as the direct sum of the FE spaces \mathcal{V}^h and \mathcal{Q}^h and additional spaces \mathcal{V}' and \mathcal{Q}' representing the fine scales of velocity and pressure. For these functions, we follow the choice adopted in [BCC⁺07]. We introduce the residuals in strong form of the momentum and continuity equations evaluated at the FE functions, i.e.,

$$\begin{aligned}\mathbf{r}_M(\mathbf{u}^h, p^h) &= \rho_f \frac{\partial \mathbf{u}^h}{\partial t} + \rho_f (\mathbf{u}^h \cdot \nabla) \mathbf{u}^h - \nabla \cdot \sigma_f(\mathbf{u}^h, p^h) - \mathbf{f}^h, \\ r_C(\mathbf{u}^h) &= \nabla \cdot \mathbf{u}^h.\end{aligned}$$

Then, the fine components of velocity and pressure are modeled as

$$\mathbf{u}'(\mathbf{u}^h) \approx -\tau_M(\mathbf{u}^h) \mathbf{r}_M(\mathbf{u}^h, p^h), \quad p'(\mathbf{u}^h) \approx -\tau_C(\mathbf{u}^h) r_C(\mathbf{u}^h), \quad (1.20)$$

where τ_M and τ_C are element dependent quantities defined as [For16]

$$\tau_M(\mathbf{u}^h) = \left(\frac{\sigma^2 \rho_f^2}{\Delta t^2} + \frac{\rho_f^2}{h_K^2} |\mathbf{u}^h|^2 + \frac{\mu_f^2}{h_K^4} C_r \right)^{-1/2},$$

$$\tau_C(\mathbf{u}^h) = \frac{h_K^2}{\tau_M(\mathbf{u}^h)}.$$

Here, h_K is the element size of element $K \in \mathcal{T}^h$ (this being a valid triangulation of Ω), σ is the order of the time discretization, and $C_r = 60 \cdot 2^{q_u - 2}$ (q_u being the polynomial order of the velocity).

Let us now introduce the following forms

$$\begin{aligned} a_{NS}(\{\mathbf{u}^h, p^h\}, \{\mathbf{v}^h, q^h\}) &= \langle \rho_f \frac{\partial \mathbf{u}^h}{\partial t}, \mathbf{v}^h \rangle + \langle (\mathbf{u}^h \cdot \nabla) \mathbf{u}^h, \mathbf{v}^h \rangle \\ &\quad + \langle \sigma_f(\mathbf{u}^h, p^h), \nabla \mathbf{v}^h \rangle + \langle \nabla \cdot \mathbf{u}^h, q^h \rangle, \\ a_{SUPG}(\{\mathbf{u}^h, p^h\}, \{\mathbf{v}^h, q^h\}) &= -\langle \mathbf{u}'(\mathbf{u}^h), \rho_f \mathbf{u}^h \cdot \nabla \mathbf{v}^h + \nabla q^h \rangle - \langle p'(\mathbf{u}^h), \nabla \cdot \mathbf{v}^h \rangle, \\ a_{VMS}(\{\mathbf{u}^h, p^h\}, \{\mathbf{v}^h, q^h\}) &= -\langle \mathbf{u}'(\mathbf{u}^h), \rho_f \mathbf{u}^h \cdot (\nabla \mathbf{u}^h)^T \rangle - \langle \mathbf{u}'(\mathbf{u}^h) \otimes \mathbf{u}'(\mathbf{u}^h), \nabla \mathbf{v}^h \rangle. \end{aligned}$$

The weak formulation of the Navier–Stokes equations in VMS formulation reads:

(W1.3) given \mathbf{f} , \mathbf{g} , \mathbf{h} regular enough, find $U^h = (\mathbf{u}^h, p^h) \in \mathcal{V}^h \times \mathcal{Q}^h$, such that, for every $t \in (0, T)$ and for all $V^h = \{\mathbf{v}^h, q^h\} \in \mathcal{V}_0^h \times \mathcal{Q}^h$,

$$a_{NS}(U^h, V^h) + a_{SUPG}(U^h, V^h) + a_{VMS}(U^h, V^h) = \langle \mathbf{f}, \mathbf{v} \rangle + \langle \mathbf{h}, \mathbf{v} \rangle_{\Gamma_N},$$

and such that $\mathbf{u}^h|_{t=0} = \mathbf{u}_0^h$ and $\gamma_{\Gamma_D} \mathbf{u}^h = \mathbf{g}$.

Remark 1.4. Form a_{SUPG} is the typical stabilization term of the Streamline–Upwind Petrov–Galerkin (SUPG) method, whereas a_{VMS} includes additional quantities which are particular to the VMS method. In this thesis, we adopt either the SUPG stabilization—hence, without accounting for the bilinear form a_{VMS} , or a VMS-SUPG type stabilization, which instead does not include the last term in a_{VMS} (which models the Reynolds cross-stresses).

Remark 1.5. From the practical perspective, SUPG and VMS-SUPG methods offer the great advantage that they allow for stable discretizations of the Navier–Stokes equations even when employing equal order polynomials, which would normally lead to the inf-sup condition not being satisfied. This can offer great efficiency improvements because employing, for example, P1-P1 instead of P2-P1 elements considerably reduces the total number of dofs, especially when considering fine meshes.

We highlight that the VMS stabilization is consistent because every term that is not found in the usual formulation of the Navier–Stokes equations depends on the fine-scale velocity and pressure, which are functions of the residuals of the momentum and continuity

equations, as shown in Eq.(1.20). Hence, the exact solution of the Navier–Stokes equations trivially satisfies the additional terms required in the VMS stabilization.

For the full discretization of W1.3, we refer to [FD15, For16].

1.6 Fluid-structure interaction with deformable domains

In cardiovascular simulations, it is often essential to take into account the physical action of the arterial wall. As a matter of fact, the vessels are compliant to the blood flow through them, and their mechanical response influences, in turn, the blood dynamics. From the mathematical perspective, the coupling of the blood dynamics with the arterial wall response is a multiphysics problem, and it is beneficial to introduce the fluid domain Ω_f , the structure domain Ω_s , and the interface between the two Γ_{fs} . However, since the arterial wall forces also cause its deformation, these domains are moving in time and should be treated as such for the most accurate results. In this section, we introduce a model of fluid-structure interaction in which the two types of physics at play are modeled and in which the deformation of the domains is carried out through an ALE map. This formulation is also adopted in [Cro11, CDFQ11, For16]. The relaxation of some of the hypotheses which this model is built upon leads to a reduced model in which the structure domain is collapsed to a membrane and in which the domains are kept fixed in time. We refer the reader to [FVCJ⁺06, FBTH09, Col14, CDQ14] for more information regarding these simplified fluid-structure interaction models.

1.6.1 Strong formulation

In order to take into account the deformation of the fluid and structure domains, we first introduce the corresponding reference configurations $\widehat{\Omega}_f$ and $\widehat{\Omega}_s$ and the reference interface $\widehat{\Gamma}_{fs}$. We remark that any symbol written with a hat notation in this section refers to the reference configuration. The deformed fluid and structure domains and the relative interface are then a function of time, i.e., $\Omega_f(t)$, $\Omega_s(t)$ and $\Gamma_{fs}(t)$, respectively (although the explicit dependence on time is omitted hereon for simplicity). Moreover, we use the subscripts f and s to denote quantities related to the fluid or structure problems, respectively. Let us introduce the structure displacement $\widehat{\mathbf{d}}_s$ and its harmonic extension to the fluid domain $\widehat{\mathbf{d}}_f$, which satisfies

$$\begin{aligned} -\Delta \widehat{\mathbf{d}}_f &= \mathbf{0} && \text{in } \widehat{\Omega}_f, \\ \widehat{\mathbf{d}}_f &= \widehat{\mathbf{d}}_s && \text{on } \widehat{\Gamma}_{fs}. \end{aligned}$$

Then, for any point $\widehat{\mathbf{x}} \in \widehat{\Omega}_f$, we define the ALE map $\mathcal{A}_t(\widehat{\mathbf{x}}) = \widehat{\mathbf{x}} + \widehat{\mathbf{d}}_f(\widehat{\mathbf{x}}, t)$, which determines the geometric deformation of the fluid domain as shown in Fig. 1.6.

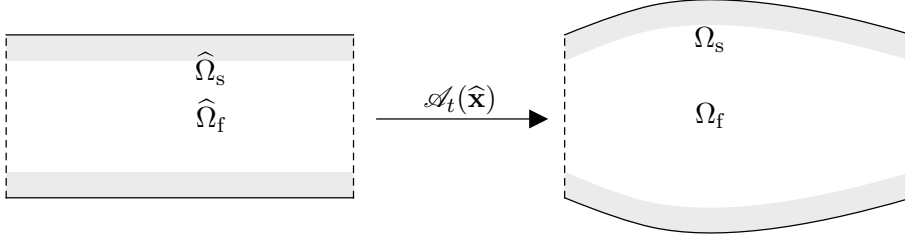


Figure 1.6 – Scheme of the ALE mapping.

The Navier–Stokes equations in ALE formulation read

$$\begin{aligned}
 \rho_f \frac{\partial \mathbf{u}_f}{\partial t} \Big|_{\hat{\mathbf{x}}} + \rho_f [(\mathbf{u}_f - \mathbf{w}_f) \cdot \nabla] \mathbf{u}_f - \nabla \cdot \sigma_f(\mathbf{u}_f, p_f) &= \mathbf{f}_f & \text{in } \Omega_f, \\
 \nabla \cdot \mathbf{u}_f &= 0 & \text{in } \Omega_f, \\
 \mathbf{u}_f &= \mathbf{g}_f & \text{on } \Gamma_{f,D}, \\
 \sigma_f \mathbf{n}_f &= \mathbf{h}_f & \text{on } \Gamma_{f,N}, \\
 \mathbf{u}_f \circ \mathcal{A}_t &= \frac{\partial \hat{\mathbf{d}}_s}{\partial t} & \text{on } \hat{\Gamma}_{fs},
 \end{aligned} \tag{1.21}$$

where \mathbf{n}_f is the normal unit vector to the fluid domain, and

$$\frac{\partial}{\partial t} \Big|_{\hat{\mathbf{x}}} = \frac{\partial}{\partial t} + \mathbf{w}_f \cdot \nabla, \quad \mathbf{w}_f(\mathbf{x}) = \frac{\partial \mathcal{A}_t(\mathbf{x})}{\partial t}.$$

The initial condition of the velocity has to be prescribed, i.e., $\mathbf{u}_f = \mathbf{u}_{f,0}$ at $t = 0$. The last condition (*kinematic condition*) in Eq. (1.21) ensures the adherence of the fluid to the structure; owing to the definition of \mathbf{w}_f , it is possible to rewrite it as $\mathbf{u}_f \circ \mathcal{A}_t = \mathbf{w}_f$.

In this thesis we consider a linear elasticity model for the structure. The equations, in the Lagrangian frame of reference, read

$$\begin{aligned}
 \hat{\rho}_s \frac{\partial^2 \hat{\mathbf{d}}_s}{\partial t^2} - \nabla_{\hat{\mathbf{x}}} \cdot \Pi(\hat{\mathbf{d}}_s) &= \hat{\mathbf{f}}_s & \text{in } \hat{\Omega}_s, \\
 \hat{\mathbf{d}}_s &= \hat{\mathbf{g}}_s & \text{on } \hat{\Gamma}_{s,D}, \\
 \Pi(\hat{\mathbf{d}}_s) \hat{\mathbf{n}}_s &= \hat{\mathbf{h}}_s & \text{on } \hat{\Gamma}_{s,N}, \\
 \Pi(\hat{\mathbf{d}}_s) \hat{\mathbf{n}}_s + \hat{\sigma}_f \hat{\mathbf{n}}_f &= \mathbf{0} & \text{on } \hat{\Gamma}_{fs},
 \end{aligned} \tag{1.22}$$

where $\hat{\mathbf{f}}_s$ is a forcing term, $\hat{\mathbf{g}}_s$ and $\hat{\mathbf{h}}_s$ are the Dirichlet and Neumann data, $\hat{\mathbf{n}}_f$ and $\hat{\mathbf{n}}_s$ are normal unit vectors to the fluid and structure domains in the reference configurations

1.6. Fluid-structure interaction with deformable domains

respectively, $\widehat{\sigma}_f = |I + \nabla_{\widehat{\mathbf{x}}}\widehat{\mathbf{d}}_f|(I + \nabla_{\widehat{\mathbf{x}}}\widehat{\mathbf{d}}_f)^{-T} \sigma_f$, $\widehat{\rho}_s$ is the density of the structure, and

$$\Pi(\widehat{\mathbf{d}}_s) = \lambda_s \text{Tr} \left(\frac{\nabla \widehat{\mathbf{d}}_s + \nabla \widehat{\mathbf{d}}_s^T}{2} \right) I + \mu_s (\nabla \widehat{\mathbf{d}}_s + \nabla \widehat{\mathbf{d}}_s^T) \quad (1.23)$$

is the first Piola–Kirchhoff stress tensor. In this constitutive law, the coefficients λ_s and μ_s may be written in terms of the Lamé coefficients (i.e., the Young modulus E_s and the Poisson ratio ν_s) as

$$\lambda_s = \frac{E_s \nu_s}{(1 - 2\nu_s)(1 + \nu_s)}, \quad \mu_s = \frac{E_s}{2(1 + \nu_s)}.$$

Due to the presence of the second derivative in time in the first equation in (1.22), in the case of the linear elasticity problem the initial data for the displacement and its derivative must be prescribed. Hence, $\widehat{\mathbf{d}}_s = \widehat{\mathbf{d}}_{s,0}$ and $\partial \widehat{\mathbf{d}}_s / \partial t = \widehat{\mathbf{d}}_{s,0}^{\text{der}}$ at $t = 0$. The last condition in Eq. (1.22) (*dynamic condition*) imposes Newton’s third law of motion, i.e., the equivalence of the stresses on the fluid and structure sides of the interface $\widehat{\Gamma}$.

1.6.2 Space discretization

The weak formulation for this fluid-structure interaction problem is derived in [Nob01, LTM01], whereas for the space discretization we refer, e.g., to [CDF17, For16]. Since in Section 2.7 we employ the coupling method introduced in Chapter 2 in a fluid-structure interaction benchmark, we briefly touch on the main ingredients of the discretized problem.

Specifically, the semi-discrete form of the fluid-structure interaction problem reads

$$\mathcal{M}_s \ddot{\mathbf{W}}^h + \mathcal{M}_f \dot{\mathbf{W}}^h + \mathcal{L}(\mathbf{W}^h) \mathbf{W}^h = \mathbf{F}^h.$$

Vector $\mathbf{W} = \text{vec}(\mathbf{u}_f^h, \mathbf{p}_f^h, \widehat{\mathbf{d}}_s^h, \widehat{\mathbf{d}}_f^h, \boldsymbol{\lambda}^h)$ is the column vector collecting the dofs of all the variables involved (the vec function performs the vectorization of its arguments). In particular, we have: the vectors of dofs of fluid velocity \mathbf{u}_f^h and pressure \mathbf{p}_f^h , the vectors of dofs in the reference domain for fluid $\widehat{\mathbf{d}}_f^h$ and structure displacement $\widehat{\mathbf{d}}_s^h$, and a vector of dofs for a Lagrange multiplier, $\boldsymbol{\lambda}^h$, which is introduced to perform the coupling. Chapter 2 is devoted to the study of a nonconforming method for the discretization of the space corresponding to such Lagrange multiplier. Furthermore, $\mathbf{F} = \text{vec}(\mathbf{f}_f^h, \mathbf{0}, \widehat{\mathbf{f}}_s^h, \mathbf{0}, \mathbf{0})$ (where \mathbf{f}_f^h and $\widehat{\mathbf{f}}_s^h$ encode the projection of forcing terms and Neumann terms of fluid and structure

onto the FE space),

$$\mathcal{M}_s = \begin{bmatrix} & \\ & M_s^h \\ & \\ & \end{bmatrix}, \quad \mathcal{M}_f = \begin{bmatrix} M_f^h \\ & \\ & \\ & \end{bmatrix},$$

and

$$\mathcal{L}(\mathbf{W}^h) = \begin{bmatrix} K^h + C^h(\mathbf{u}_f^h) & (D^h)^T & & (\widehat{B}_f^h)^T \\ & D^h & & \\ & & \widehat{E}^h & (\widehat{B}_s^h)^T \\ & & & \widehat{G}^h \\ \widehat{B}_f^h & & \widehat{B}_s^h & \end{bmatrix}. \quad (1.24)$$

Matrix in Eq. (1.24) features the same matrices which appear in the semi-discrete formulation of the Navier–Stokes equations (1.5), along with the matrices discretizing the linear elasticity problem \widehat{E}^h and the geometry problem for the motion of the fluid domain \widehat{G}^h (that is, the stiffness matrix computed on the FE space used for the discretization of $\widehat{\mathbf{d}}_f$), and the coupling matrices \widehat{B}_f^h and \widehat{B}_s^h . These are computed as

$$(\widehat{B}_f^h)_{ij} = \int_{\widehat{\Gamma}_{fs}} \widehat{\boldsymbol{\eta}}_i^h \cdot (\boldsymbol{\varphi}_{f,j}^h \circ \mathcal{A}_t), \quad (\widehat{B}_s^h)_{ij} = \int_{\widehat{\Gamma}_{fs}} \widehat{\boldsymbol{\eta}}_i^h \cdot \widehat{\boldsymbol{\varphi}}_{s,j}^h;$$

$\boldsymbol{\varphi}_{f,i}^h$ and $\widehat{\boldsymbol{\varphi}}_{s,i}^h$ are FE basis functions discretizing the spaces for the velocity and structure displacement, respectively, whereas $\widehat{\boldsymbol{\eta}}_i^h$ are, for the time being, generic basis functions based on the triangulations of either $\widehat{\Omega}_f$ or $\widehat{\Omega}_s$ spanning the space of the discrete Lagrange multiplier. We consider a specific choice for these basis functions, which is also independent of the FE grids, in Section 2.7.

The discretization in time requires the treatment of the second-order time derivative. This can be performed by the Newmark method [New59, Hug12]. The first-order derivative can be discretized by employing, for example, a BDF scheme as done in Section 1.3 for the Navier–Stokes equations. The complete discrete formulation of the fluid-structure problem with this approach is derived in [For16].

1.7 Concluding remarks

This chapter presented mathematical models that are extensively used in the cardiovascular context and specifically throughout this thesis. We introduced the strong formulation of the Navier–Stokes equations and addressed its discretization in space by the FE method and its discretization in time by BDF and Rosenbrock schemes. We motivated the choice

to focus on BDF in the remainder of the thesis by presenting results obtained during a study of Rosenbrock schemes that highlighted problematic aspects of these methods. These are the order reduction affecting (in particular) the pressure and the poor performance of the linear solver, which failed to converge even for moderate Reynolds numbers. Then, we presented basic concepts of the theory of saddle-point problems—of which the Navier–Stokes equations are a representative member—and discussed alternatives to satisfy the inf-sup condition in fluid problems: the use of suitable stable elements (i.e., the generalized Taylor–Hood elements) and ad-hoc stabilization techniques such as the Variational Multiscale one. Lastly, we focused on the task of coupling fluid and structure in an ALE framework. We first presented the strong formulation of the multiphysics problem and then briefly discussed the structure of the discretized problem.

2 Non-conforming coupling of PDEs

The goal of this thesis is the development of a ROM for the approximation of the blood dynamics in vessels subdivided into several geometrical building blocks. As explained in further details in Chapter 4, in each building block, the PDEs are discretized by the FE or RB methods. Although the approximation spaces in a given subdomain are independent of the neighboring ones, the local solutions need to satisfy particular coupling conditions in order for the global approximation to be an accurate approximation of blood flow in the whole vessel.

This chapter focuses on the definition of such coupling conditions and on a specific and original discretization algorithm for the numerical discretization of the resulting global problem. In Section 2.1 we introduce some fundamental concepts characterizing Domain Decomposition methods. These approaches are beneficial to the design of parallel preconditioners, and for this reason, the term is nowadays often associated with this particular branch of computational science; however, we here present this family of methods in their original form, namely as an ensemble of strategies for the solution of problems defined on separate regions of the domain. As we shall see throughout this introduction, the global problem is—from the theoretical point of view—a mixed problem featuring the primal variable (e.g., the velocity in the Navier–Stokes equations) and a dual variable; the latter is a Lagrange multiplier ensuring that the (weak) continuity of the primal variable and that the (strong) continuity of the normal stresses over the separating interface are satisfied. Numerical methods focusing on problems with this structure are sometimes referred to as primal hybrid methods. In Section 2.2, we present the principles of this class of methods on an elliptic problem defined over a domain partitioned into two regions. Section 2.3 focuses on the development of a novel approach for the discretization of the weak formulation derived in Section 2.2. In Section 2.4, we briefly present other nonconforming methods, namely the mortar method [Ber89], INTERNODES [DFGQ16, GQ16], and the three-field method [BM94], and explicitly highlight the similarities and peculiarities with respect to our approach. In Section 2.5 we contextualize the stability of the discrete problem—which features a saddle point

structure—in light of the concepts presented in Section 1.4 and in particular in relationship with the inf-sup condition. In Section 2.6, the method is applied to two-dimensional and three-dimensional benchmark problems: we consider the Poisson problem approximated with the FE method on two subdomains (Section 2.6.2) and the Navier–Stokes equations on three subdomains (Section 2.6.3), numerically solved by employing the FE method (Section 2.6.3) and the FE method coupled with IGA (Section 2.6.3). Moreover, in Section 2.7 we draw a parallel between these mixed methods and the fluid-structure interaction problem introduced in Section 1.6. We also apply the same discretization approach of the previous sections to a classic fluid-structure interaction benchmark.

The theory and the discussion carried out in this chapter, with the exception of the original results presented in Section 2.7, have been published in [DIP19].

2.1 An introduction to Domain Decomposition methods

In numerical analysis, Domain Decomposition methods are techniques for splitting PDEs into smaller and coupled problems defined over subsets of the original domain. The splitting may be motivated by physical reasons, for instance when the subdomains are characterized by different governing equations (e.g., in fluid-structure-interaction problems [DDQ06]) or by discretization needs, should it be required to employ specific methods—e.g., the FE method or isogeometric analysis (IGA)—or specific polynomial degrees in certain regions of the domain [TW05]. Moreover, Domain Decomposition methods allow the mapping of the subproblems to separate cores [IVA93] and the computation of scalable and efficient parallel preconditioners. These techniques have become so important to the solution of large-scale problems on multiprocessors or clusters that, nowadays, the term *Domain Decomposition* is most commonly used in the context of high-performance computing.

Domain decomposition methods are typically based either on iterative or direct procedures [BHS03]. In the first class of techniques, the continuity on the interfaces of the solution, of its normal derivatives, or combinations of the two are strongly imposed. Typically, these methods require solving the problems defined on the subdomains separately multiple times while imposing artificial boundary conditions based on the solutions at the previous iteration. The type of boundary conditions employed in every subdomain is a peculiarity of each algorithm, so that the literature on the topic commonly refers to the Dirichlet-Dirichlet algorithm, the Dirichlet-Neumann algorithm, and so forth; see e.g., [TW05] for details. These strategies allow reducing the size of the linear systems to be solved and, most importantly, to compute the solution in each subdomain in parallel.

In this chapter, we present an approach belonging to the class of direct procedures in which the continuity conditions (often called *transmission conditions*) are weakly imposed through the use of suitable Lagrange multipliers. Our method is applied to PDEs written

in primal hybrid formulation, and for this reason, it shares some of the features of the well-known mortar method [Ber89, BMR05]. This was originally proposed to solve PDEs by combining spectral elements and finite elements or by combining FE spaces with different polynomial degrees in nonoverlapping portions of the domain [QV99]. Since then, the mortar method has become the nonconforming method of choice in many areas of computational science and engineering, for example in contact mechanics [PL04], solid mechanics [Pus04], fluid mechanics [EPGW14] and fluid-structure interaction problems [KPKW11]; see also [PW14, HGC⁺14, BCS03]. The implementation of the mortar method is not straightforward, as the algorithm is based on L^2 -projections of the traces of functional spaces defined on a group subdomains—the masters—onto the interfaces of the adjacent ones—the slaves. INTERNODES [DFGQ16], a recently developed method for the treatment of nonconforming meshes, overcomes this issue by treating the transmission conditions with the interpolation of basis functions of the master domains onto the interfaces of the slaves.

As in the mortar method, our approach is based on the idea that the global problem can be subdivided into a set of smaller problems coupled with weak conditions relying on basis functions defined on the interfaces. In the mortar method, such basis functions are obtained from the trace space of the adjacent slave domains. This choice is convenient from the analysis standpoint but makes the implementation of the method cumbersome. Another drawback is that the final solution is dependent on the choice of master and slave domains. The originality of our method is to consider basis functions on the interfaces that are completely independent of the discretization of the neighboring domains: in this chapter, we employ spectral basis functions (specifically, Fourier basis functions, but other possibilities are considered in the rest of this thesis). The advantage of the approach we present is that the resulting global solution is not dependent on the classification of master and slave domains, which, in fact, is not required. Moreover, the accuracy of the coupling of solutions at the interfaces is easily tuned by varying the number of basis functions on the common boundary. Our approach can be interpreted as a specialization of the three-field method [BM94], where the space of the three Lagrange multipliers used to impose the continuity of the solution weakly is (a priori) independent of the spaces defined on the adjacent domains. As the functional spaces in the subdomains are mutually independent, our choice of basis functions is well-suited for the coupling of solutions obtained on nonconforming meshes (at the interfaces), with FE spaces with different polynomial degrees, or with different numerical methods, e.g., FE method, spectral element method, or IGA. In this chapter, we specifically consider the coupling of FE and IGA; the coupling of RB solutions is treated in the subsequent parts of the thesis.

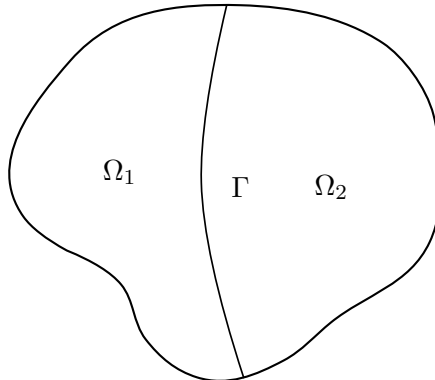


Figure 2.1 – Example of partitioned domain $\Omega = \Omega_1 \cup \Omega_2$.

2.2 Theory of primal hybrid methods

In this section, we recall the theory of primal hybrid methods for the solution of PDEs. These approaches are based on the *primal hybrid principle* [RT77], according to which the continuity across subdomains is weakened by means of Lagrange multipliers. We refer the reader to [BBF13, AP16, Bel99] for the theory of primal hybrid methods. Here, we recall the main ideas by following the presentation in [BDW99]. We also restrict ourselves to only two partitions of the domain; however—as we shall see in Section 2.3.1 and in Chapter 4—the discussion can be easily extended to the case of multiple subdomains.

We are interested in solving a generic PDE described by a second order elliptic operator on an open and bounded domain Ω with homogeneous Dirichlet boundary conditions on $\partial\Omega$. Specifically, we assume that $a(\varphi, \psi)$ for $\varphi, \psi \in H^1(\Omega)$ is the bilinear form corresponding to the elliptic operator and f is a given forcing term; we consider problems whose weak formulation can be written as:

(W2.1) *given $f \in H^{-1}(\Omega)$, find $u \in H_0^1(\Omega)$, such that*

$$a(u, v) = \langle f, v \rangle \quad \forall v \in H_0^1(\Omega). \quad (2.1)$$

In the sequel, we will use the Poisson problem

$$-\Delta u = f \quad \text{in } \Omega \quad (2.2)$$

with homogeneous Dirichlet boundary conditions (i.e., $u = 0$ on $\partial\Omega$) as representative of this class of problems. In this specific case, $a(\varphi, \psi) = \langle \nabla\varphi, \nabla\psi \rangle$.

Let us assume that the domain Ω can be partitioned into two nonoverlapping open and bounded domains, such that $\Omega = \Omega_1 \cup \Omega_2$ and $\Omega_1 \cap \Omega_2 = \emptyset$; as shown in Fig. 2.1, we denote Γ the interface between the two domains, i.e., $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$. Rather than the global problem W2.1, our goal is to solve two local and coupled problems defined on

the partitions Ω_i , such that the global solution can be constructed by combining the solutions of the local problems. To this end, let us introduce for $i = 1, 2$ the functional spaces $\mathcal{X}_i = H^1_{\partial\Omega \cap \partial\Omega_i}(\Omega_i)$ and

$$\mathcal{X} := \{\varphi \in L^2(\Omega) : \varphi|_{\Omega_i} \in \mathcal{X}_i \text{ for } i = 1, 2\},$$

which is a Hilbert space when endowed with the (broken) norm

$$\|\varphi\|_{\mathcal{X}}^2 := \sum_{i=1}^2 \|\varphi|_{\Omega_i}\|_{H^1(\Omega_i)}^2.$$

Let us also recall the following definition

$$H(\operatorname{div}; \Omega) := \{\phi \in [L^2(\Omega)]^d : \nabla \cdot \phi \in L^2(\Omega)\},$$

where d is the dimensionality of the problem.

Space $H_0^1(\Omega)$ is characterized as a subspace of \mathcal{X} under suitable conditions [BDW99], which we state in the following Proposition. We remark that an analogous result is presented and proved in [BBF13].

Proposition 2.1. *Let*

$$\tilde{\mathcal{V}} := \{\varphi \in \mathcal{X} : \sum_{i=1}^2 \langle \phi \cdot \mathbf{n}_i, \varphi \rangle_{\partial\Omega_i} = 0 \quad \forall \phi \in H(\operatorname{div}; \Omega)\},$$

where \mathbf{n}_i is the outward unit vector normal to $\partial\Omega_i$. Then, $H_0^1(\Omega) \equiv \tilde{\mathcal{V}}$.

Remark 2.1. Proposition 2.1 must be modified if $\partial\Omega = \Gamma_D \cup \Gamma_N$ and if the PDE is equipped with homogeneous Dirichlet conditions on Γ_D and Neumann conditions on Γ_N , as discussed in [BBF13]. In this case, we introduce

$$H_{\Gamma_N}(\operatorname{div}; \Omega) := \{\phi \in H(\operatorname{div}; \Omega) : \langle \phi \cdot \mathbf{n}, v \rangle_{\partial\Omega} = 0 \quad \forall v \in H_{\Gamma_D}^1(\Omega)\}.$$

Then

$$\overset{\circ}{\mathcal{V}} := \{\varphi \in \mathcal{X} : \sum_{i=1}^2 \langle \phi \cdot \mathbf{n}_i, \varphi \rangle_{\partial\Omega_i} = 0 \quad \forall \phi \in H_{\Gamma_N}(\operatorname{div}; \Omega)\}$$

is a characterization of $H_{\Gamma_D}^1(\Omega)$.

Condition $\sum_{i=1}^2 \langle \phi \cdot \mathbf{n}_i, \varphi \rangle_{\partial\Omega_i} = 0$ for each $\phi \in H(\operatorname{div}; \Omega)$ is global, in the sense that it involves the trace of $\varphi|_{\Omega_i}$ on the whole $\partial\Omega_i$, even though it essentially constrains the restrictions of φ to Ω_1 and Ω_2 to have the same trace at the common interface Γ . Unfortunately, splitting the dualities into two parts corresponding to Γ and $\partial\Omega \setminus \Gamma$ is

Chapter 2. Non-conforming coupling of PDEs

not allowed, as the restrictions of the traces to portions of $\partial\Omega_i$ can lead to unbounded dualities. To overcome this issue, we introduce

$$H_{00}^{1/2}(\Gamma_i) := \{\eta \in H^{1/2}(\Gamma_i) : E_i^0 \eta \in H^{1/2}(\partial\Omega_i)\},$$

with norm

$$\|\eta\|_{H_{00}^{1/2}(\Gamma_i)} := \|E_i^0 \eta\|_{H^{1/2}(\partial\Omega_i)},$$

where $E_i^0 \eta$ is the trivial extension by zero of η to the whole boundary of $\partial\Omega_i$ and $\Gamma_i = \partial\Omega_i \setminus \partial\Omega$. In the following, we will consider $H_{00}^{1/2}(\Gamma) := H_{00}^{1/2}(\Gamma_1) \cap H_{00}^{1/2}(\Gamma_2)$. Let us define the spaces

$$\mathcal{X}_{00} := \{\varphi \in \mathcal{X} : [\varphi]_{\Gamma} \in H_{00}^{1/2}(\Gamma)\},$$

where $[\varphi]_{\Gamma}$ denotes by our convention the difference of the traces of $\varphi|_{\Omega_2}$ and $\varphi|_{\Omega_1}$ on Γ , and

$$\Lambda := H_{00}^{-1/2}(\Gamma), \tag{2.3}$$

with norm

$$\|\eta\|_{\Lambda} := \|\eta\|_{H_{00}^{-1/2}(\Gamma_1)} + \|\eta\|_{H_{00}^{-1/2}(\Gamma_2)}.$$

Furthermore, we introduce the bilinear form

$$b(\varphi, \xi) := \langle \xi, [\varphi]_{\Gamma} \rangle_{\Lambda}$$

for $\varphi \in \mathcal{X}_{00}$ and $\xi \in \Lambda$. It can be easily verified [BDW99] that another characterization of $H_0^1(\Omega)$ analogous to that in Proposition 2.1 is given by

$$H_0^1(\Omega) \equiv \mathcal{V} := \{\varphi \in \mathcal{X}_{00} : b(\varphi, \xi) = 0 \quad \forall \xi \in \Lambda\}. \tag{2.4}$$

In the sequel, we will use the letter \mathcal{V} to refer to $H_0^1(\Omega)$.

We are now ready to state the primal hybrid formulation of the original weak formulation W2.1. We remark that, whenever applied to functions of \mathcal{X} , the bilinear form $a(\cdot, \cdot)$ is to be intended as the sum of the bilinear forms restricted to the two subdomains.

(W2.2) *given $f \in H^{-1}(\Omega)$, find $u \in \mathcal{X}_{00}$ and $\lambda \in \Lambda$ such that*

$$\begin{aligned} a(u, v) + b(v, \lambda) &= \langle f, v \rangle \quad \forall v \in \mathcal{X}_{00}, \\ b(u, \eta) &= 0 \quad \forall \eta \in \Lambda. \end{aligned} \tag{2.5}$$

2.3. Discretization of the primal hybrid formulation

Proposition 2.2. *If $u \in \mathcal{V}$ is a solution of W2.1 and there exists $\lambda \in \Lambda$ such that*

$$b(v, \lambda) = \langle f, v \rangle - a(u, v) \quad \forall v \in \mathcal{X}_{00}, \quad (2.6)$$

then $(u, \lambda) \in \mathcal{X}_{00} \times \Lambda$ is a solution of W2.2. On the other hand, if $(u, \lambda) \in \mathcal{X}_{00} \times \Lambda$ is a solution of W2.2, then $u \in \mathcal{V}$ and u is a solution of W2.1.

Proof. Let $u \in \mathcal{V}$ be a solution of W2.1, then $u \in \mathcal{X}_{00}$ and the second condition in Eq. (2.5) is satisfied because of definition (2.4). The first condition in Eq. (2.5) is satisfied when choosing $\lambda \in \Lambda$ such that Eq. (2.6) is verified. Conversely, if $(u, \lambda) \in \mathcal{X}_{00} \times \Lambda$ is a solution for W2.2, then $u \in \mathcal{V}$ because of the second condition in Eq. (2.5). Moreover, for each $v \in \mathcal{V}$, $b(v, \xi) = 0$ for all $\xi \in \Lambda$ and, in particular, for $\xi = \lambda$, and the first condition in Eq. (2.5) becomes Eq. (2.1). \square

Remark 2.2. If we consider the Poisson equation (2.2), then Eq. (2.6) is verified by taking $\phi = -\nabla u$ and by choosing $\lambda \in \Lambda$ such that $\phi \cdot \mathbf{n}_1 = \lambda$, \mathbf{n}_1 being the outward unit vector normal to $\partial\Omega_1$. Indeed, by using integration by parts we find, for all $v \in \mathcal{X}_{00}$,

$$\begin{aligned} b(v, \lambda) &= \langle \phi \cdot \mathbf{n}_1, [v]_\Gamma \rangle_\Lambda = - \sum_{i=1}^2 \int_\Gamma \nabla u \cdot \mathbf{n}_i v \\ &= \sum_{i=1}^2 \left(\int_{\Omega_i} f v - \int_{\Omega_i} \nabla u \cdot \nabla v \right) = \langle f, v \rangle - a(u, v), \end{aligned}$$

where we used the fact that $\mathbf{n}_1 = -\mathbf{n}_2$. Note that, if we defined the jump across the interface of a function $\varphi \in \mathcal{X}_{00}$ as the difference of the traces on Γ of $\varphi|_{\Omega_1}$ and $\varphi|_{\Omega_2}$, then $\phi \cdot \mathbf{n}_2 = \lambda$. Hence, the Lagrange multiplier in Eq. (2.5) plays the role of the normal derivative of u at the interface Γ [Woh00], with the direction of the normal at the interface being determined by the definition of the jump.

2.3 Discretization of the primal hybrid formulation

We now consider the discretization of the weak formulation W2.2. We take two arbitrary finite dimensional functional spaces $\mathcal{X}_1^h \subset \mathcal{X}_1$ and $\mathcal{X}_2^h \subset \mathcal{X}_2$ spanned by two sets of basis functions $\varphi_{1,i}^h \in \mathcal{X}_1^h$ (with $i = 1, \dots, N_{u,1}^h$) and $\varphi_{2,i}^h \in \mathcal{X}_2^h$ (with $i = 1, \dots, N_{u,2}^h$) respectively. We assume that functions in \mathcal{X}_1^h and \mathcal{X}_2^h can be trivially extended by zero in the other domain and that such extension belong to \mathcal{X}_{00} . The discrete version of the global space \mathcal{X}_{00} is consequently obtained by considering the space $\mathcal{X}^h \subset \mathcal{X}_{00}$ of dimension $\dim(\mathcal{X}^h) = N_u^h = N_{u,1}^h + N_{u,2}^h$ and spanned by the basis functions

$$\{\varphi_i^h\}_{i=1}^{N_u^h} = \{\varphi_{i,1}^h\}_{i=1}^{N_{u,1}^h} \cup \{\varphi_{i,2}^h\}_{i=1}^{N_{u,2}^h}.$$

Chapter 2. Non-conforming coupling of PDEs

The solution can be then approximated as $u \approx u^h = \sum_{i=1}^{N_u^h} u_i^h \varphi_i^h$. In the numerical applications in Section 2.6, we will consider standard FE Lagrangian basis functions and B-Splines basis functions for the discretization of the spaces defined above; in this section, we consider—to fix our ideas—the case in which the FE method is employed. The FE basis functions are built over suitable triangulations \mathcal{T}_1^h and \mathcal{T}_2^h of Ω_1 and Ω_2 respectively; we will always assume that such triangulations meet standard regularity requirements [QV08], but we do not require the conformity of the global mesh $\mathcal{T}^h = \mathcal{T}_1^h \cup \mathcal{T}_2^h$. We define conforming meshes those meshes for which the intersection of two elements is either null, a vertex or a whole edge; in nonconforming meshes, on the contrary, two elements can also share portions of their edges. The discretization parameter h is generic and defines a family of discretized spaces; when using finite elements, for example, h refers to the maximum edge length of an element (often called mesh size) in the triangulations of Ω_1 and Ω_2 . More generally, h could be also considered a characteristic of the single subdomain, since—as we already mentioned—the discretizations in Ω_1 and Ω_2 are independent one of the other and could be obtained by different discretization methods.

As proposed in [DIP19], we discretize Λ as Λ^δ by using a set of spectral basis functions $\xi_i^\delta \in \Lambda$, such that $\lambda \in \Lambda$ is approximated as $\lambda \approx \lambda^\delta = \sum_{i=1}^{N_\lambda^\delta} \lambda_i^\delta \xi_i^\delta$. We remark that we characterize the refinement levels for \mathcal{X}_1^h , \mathcal{X}_2^h and Λ^δ with different discretization parameters h and δ : this is to indicate that the discretization of Λ is indeed independent of the discretization on Ω_1 and Ω_2 . For instance, in the two-dimensional case, a suitable choice would consist of choosing as ξ_i^δ the basis functions associated with the low-frequencies of the Fourier basis defined on the common interface Γ , and the accuracy of the discretization of Λ^δ can be increased independently of h by adding Fourier basis functions to the set ξ_i^δ . In the numerical simulations of Section 2.6 we will follow this approach. Alternatives for the discretization of the Lagrange multiplier space include other spectral basis functions, such as, e.g., Legendre or Chebyshev polynomials. The three-dimensional case is treated in Section 2.6.2, Section 2.7, and more extensively in Chapter 4.

The discrete space for the approximation of \mathcal{V} is then defined as

$$\mathcal{V}^{h\delta} := \{\varphi^h \in \mathcal{X}^h : b(\varphi^h, \xi^\delta) = 0 \quad \forall \xi^\delta \in \Lambda^\delta\},$$

Remark 2.3. $\mathcal{V}^{h\delta}$ is not a subspace of \mathcal{V} . As a matter of fact, if Λ^δ is not equal to Λ , then there may exist $\xi \in \Lambda$, $\xi \notin \Lambda^\delta$ such that $b(\varphi^h, \xi) \neq 0$ for some $\varphi^h \in \mathcal{V}^{h\delta}$, and therefore $\varphi^h \notin \mathcal{V}$. If we replaced \mathcal{V} by $\mathcal{V}^{h\delta}$ in W2.1, we would obtain a *nonconforming method*, i.e., a numerical method in which the discretized search space is not contained into the continuous search space. The generalized version of C ea’s lemma [C ea64] for this family of methods is Strang’s second lemma [Cia02], which states that the solution u^h of the

2.3. Discretization of the primal hybrid formulation

discretized version of W2.1 satisfies

$$\|u - u^h\|_{\mathcal{V}^{h\delta}} \leq C \left(\inf_{v^h \in \mathcal{V}^{h\delta}} \|u - v^h\|_{\mathcal{V}^{h\delta}} + \sup_{w^h \in \mathcal{V}^{h\delta}} \frac{|a(u, w^h) - \langle f, w^h \rangle|}{\|w^h\|_{\mathcal{V}^{h\delta}}} \right), \quad (2.7)$$

where $C > 0$ and $\|\cdot\|_{\mathcal{V}^{h\delta}}$ is a norm for $\mathcal{V}^{h\delta}$. Note that the consistency error—i.e., the second term of the right-hand side in Eq. (2.7)—is identically zero for each $w^h \in \mathcal{V}^{h\delta}$ if $\mathcal{V}^{h\delta} \subset \mathcal{V}$ because u is a solution of W2.1.

The discretization of W2.2 is simply obtained by replacing the continuous functional spaces by their discrete counterparts, namely:

(W2.3) given $f \in H^{-1}(\Omega)$, find $u^h \in \mathcal{X}^h$ and $\lambda^\delta \in \Lambda^\delta$ such that

$$\begin{aligned} a(u^h, v^h) + b(v^h, \lambda^\delta) &= \langle f, v^h \rangle \quad \forall v^h \in \mathcal{X}^h, \\ b(u^h, \eta^\delta) &= 0 \quad \forall \eta^\delta \in \Lambda^\delta. \end{aligned} \quad (2.8)$$

By expanding u^h and λ^δ on their respective bases, Eq. (2.8) can be rewritten in system form as

$$\begin{bmatrix} A^h & (B^{h\delta})^T \\ B^{h\delta} & \end{bmatrix} \begin{bmatrix} \mathbf{u}^h \\ \boldsymbol{\lambda}^\delta \end{bmatrix} = \begin{bmatrix} \mathbf{f}^h \\ \mathbf{0} \end{bmatrix}, \quad (2.9)$$

where $A_{ij}^h = a(\varphi_j^h, \varphi_i^h)$, $B_{ij}^{h\delta} = b(\varphi_j^h, \xi_i^\delta)$, $\mathbf{u}^h = [u_1^h, \dots, u_{N_u}^h]$, $\boldsymbol{\lambda}^\delta = [\lambda_1^\delta, \dots, \lambda_{N_\lambda}^\delta]$ and $\mathbf{f}_i^h = \langle f, \varphi_i^h \rangle$. By arranging the basis functions φ_i^h and the dofs such that all the basis functions corresponding to Ω_1 come before those of Ω_2 , system (2.9) can be written as

$$\begin{bmatrix} A_1^h & & - (B_1^h)^T \\ & A_2^h & (B_2^{h\delta})^T \\ -B_1^{h\delta} & B_2^{h\delta} & \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^h \\ \mathbf{u}_2^h \\ \boldsymbol{\lambda}^\delta \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1^h \\ \mathbf{f}_2^h \\ \mathbf{0} \end{bmatrix},$$

where $(B_1^{h\delta})_{ij} = \int_\Gamma \varphi_{j,1}^h \xi_i^\delta$ and $(B_2^{h\delta})_{ij} = \int_\Gamma \varphi_{j,2}^h \xi_i^\delta$ are coupling matrices. Clearly, $B_1^{h\delta}$ and $B_2^{h\delta}$ are likely to be sparse if the basis functions internal to the subdomains have compact support, as the integrals are equal to zero if $\varphi_{j,1}^h$ and $\varphi_{j,2}^h$ vanish on Γ .

In this thesis, the computation of the coupling matrices is performed by numerically estimating the integrals by Gauss quadrature rules. Let us consider, for $z = 1$ or $z = 2$, the case of $B_z^{h\delta}$ in the two-dimensional case. The triangulation \mathcal{T}_z^h induces on Γ a partition into $N_{\Gamma,z}$ elements, i.e., $\Gamma = \bigcup_{i=1}^{N_{\Gamma,z}} K_{i,z}$. Given a Gauss quadrature rule of order

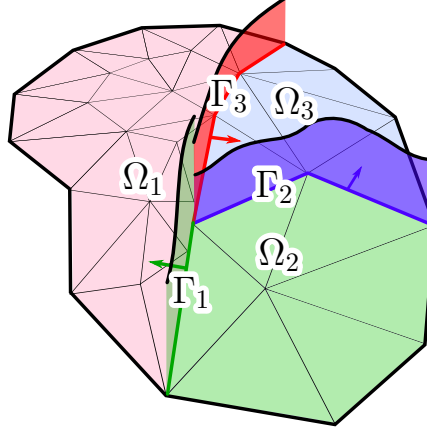


Figure 2.2 – Example of a three-way partition of Ω with three interfaces; each interface corresponds to a Lagrange multiplier space.

$2q - 1$, the approximation of each term of $B_z^{h\delta}$ is computed as

$$\begin{aligned} \left(B_z^{h\delta}\right)_{mn} &= \int_{\Gamma} \varphi_{n,z}^h \xi_m^\delta = \sum_{i=1}^{N_{\Gamma,z}} \int_{K_{i,z}} \varphi_{n,z}^h \xi_m^\delta \\ &\approx \sum_{i=1}^{N_{\Gamma,z}} \sum_{j=1}^q |\det(J_{i,z}(\mathbf{x}_j^{\text{gq}}))| \varphi_{n,z}^h(\phi_{i,z}(\mathbf{x}_j^{\text{gq}})) \xi_m^\delta(\phi_{i,z}(\mathbf{x}_j^{\text{gq}})) \omega_j, \end{aligned} \quad (2.10)$$

where $\det(J_{i,z}(\mathbf{x}_j^{\text{gq}}))$ is the determinant of the Jacobian of the map $\phi_{i,z} : K_{i,z} \rightarrow (-1, 1)$ from $K_{i,z}$ to the reference interval $(-1, 1)$, \mathbf{x}_j^{gq} is the j^{th} Gauss quadrature node in $(-1, 1)$ and ω_j is the associated weight. As it is evident from Eq. (2.10), being able to evaluate the product $\varphi_{n,z}^h \xi_m^\delta$ at each quadrature node is sufficient to compute the approximation of $B_z^{h\delta}$. In particular, it is not necessary to interpolate functions between domains. The case of geometric nonconformity, which occurs whenever the interfaces are not geometrically the same, still needs to be addressed.

2.3.1 Generalization to multiple subdomains

In previous sections we decided to limit ourselves to the case where the domain of the PDE is partitioned into two subdomains. This choice is motivated mainly by the fact that considering the generic case of multiple subdomains leads necessarily to complexity in the notation. We refer the reader to the already mentioned references [BDW99, Bel99] for examples of how the functional spaces we considered in Section 2.2 could be adapted to the case of multiple subdomains. One aspect that differentiates our approach from other methods (such as the mortar method) is that, in the discretization process, our

2.4. Relationship with other nonconforming methods

method requires defining a set of basis functions for the Lagrange multiplier space of each interface. These bases can be chosen independently, one from the other. In particular, in our numerical results, we were able to recover optimal convergence without imposing any requirements on the values of the basis functions at the cross points.

As an example, Fig. 2.2 shows a three-way partition of Ω into three domains with three interfaces. Each of the interfaces requires the definition of a corresponding space for the Lagrange multipliers. After the discretization, the algebraic system can be written as in Eq. (2.9), with

$$A^h = \begin{bmatrix} A_1^h & & \\ & A_2^h & \\ & & A_3^h \end{bmatrix}, \quad B^h = \begin{bmatrix} B_{11}^{h\delta} & -B_{12}^{h\delta} & \\ & -B_{22}^{h\delta} & B_{23}^{h\delta} \\ -B_{31}^{h\delta} & & B_{33}^{h\delta} \end{bmatrix}. \quad (2.11)$$

Matrices $(B_{ij}^{h\delta})_{mn} = \int_{\Gamma_i} \varphi_{n,j}^h \xi_{m,i}^\delta$ discretize the coupling between the j^{th} domain and the i^{th} interface. We remark that the signs of the coupling matrices are determined by the definition of the normals at each interface. It is worth noting that, besides the choice of the orientation of the normals at the interfaces, there is no hierarchy among the subdomains.

In Section 2.6.3, we consider numerical results obtained on a domain whose partition is topologically equivalent to that presented in Fig. 2.2.

2.4 Relationship with other nonconforming methods

2.4.1 Relationship with the mortar method

The mortar method can be derived from the same problem written in primal hybrid formulation W2.2 we considered in Section 2.2 [Woh00]. Its discretized weak formulation could be rewritten in the form of a saddle-point problem similar to Eq.(2.5) in which the space of Lagrange multiplier Λ_M^h depends on the discretization of either Ω_1 or Ω_2 ; see [SS98, Ses03]. In particular, the classic mortar method requires assigning to Ω_1 or to Ω_2 the role of master and slave domains. The basis functions of Λ_M^h are chosen as the traces of the basis functions defined over the triangulation of the slave domain that do not vanish on Γ ; the polynomial order of the basis functions is usually decreased by one over the first and last elements of the partition of Γ .

With respect to the mortar method, the main advantages of an approach based on the use of spectral basis functions are the following:

1. the accuracy of the coupling can be increased or decreased by varying the number of basis functions for space Λ^δ independently of the discretization in the subdomains;

2. the solution is independent of any classification of the subdomains into master and slaves, which is not required by the method;
3. the computation of the coupling matrices does not require projections between meshes, which makes the implementation of the method easier.

One drawback of our method is that the Lagrange multiplier space has to be rich enough to provide the necessary accuracy but coarse enough to satisfy the inf-sup condition, as described in Section 2.5.

2.4.2 Relationship with INTERNODES

The INTERNODES (INTERpolation for NONconforming DEcompositionS) method [DFGQ16] is based on an interpolation approach, rather than the L^2 -projection approach which characterizes the mortar method. Given each interface, the two adjacent subdomains are given the role of master and slave domains. Similarly to the mortar method, the traces of the (FE or spectral element) basis functions defined over the meshes of the master and slave domains are used to enforce the continuity of the solution and the normal stresses. More precisely, two interpolation operators—or intergrid operators—are defined: the interpolation operator from the master to the slave domain is used to ensure the continuity of the solution, while the interpolation operator from the slave to the master domain enforces the continuity of the normal fluxes. INTERNODES has been proven to retain the optimal convergence properties of the mortar method. For more information about the method and its analysis, we refer the reader to [GQ16].

Being INTERNODES closely related to the mortar method, compared to the former, our approach offers the same advantages we presented in Section 2.4.1 except for 3. Indeed, INTERNODES has the big advantage of being simple to implement and allowing for nonmatching interfaces, namely interfaces which could either feature (small) overlaps or holes between them; we refer to this situation as geometric nonconformity. We believe that our method is as simple to implement as INTERNODES and that it can be extended to nonconforming geometries with the help of localized Rescaled Radial Basis Interpolation [For16, DFQ14].

2.4.3 Relationship with the three-field method

The three-field method was originally proposed in [BM94] and analyzed in [BM01]. Compared to the mortar method, it has had significantly less impact on the Domain Decomposition community.

The multidomain extension of the weak formulation W2.1 by the three-field method reads [QV99]:

2.4. Relationship with other nonconforming methods

(W2.4) for $i = 1, 2$, find $u_i \in \mathcal{X}_i$, $\sigma_i \in H^{-1/2}(\Gamma)$ and $\psi \in H^{1/2}(\Gamma)$ such that

$$\begin{aligned}
 a(u_1, v_1) - \langle \sigma_1, v_1 \rangle_{H^{-1/2}(\Gamma)} &= \langle f, v_1 \rangle & \forall v_1 \in \mathcal{X}_1, \\
 \langle \eta_1, \psi - u_1 \rangle_{H^{-1/2}(\Gamma)} &= 0 & \forall \eta_1 \in H^{-1/2}(\Gamma), \\
 \langle \sigma_1 + \sigma_2, \rho \rangle_{H^{-1/2}(\Gamma)} &= 0 & \forall \rho \in H^{1/2}(\Gamma), \\
 \langle \eta_2, \psi - u_2 \rangle_{H^{-1/2}(\Gamma)} &= 0 & \forall \eta_2 \in H^{-1/2}(\Gamma), \\
 a(u_2, v_2) - \langle \sigma_2, v_2 \rangle_{H^{-1/2}(\Gamma)} &= \langle f, v_2 \rangle & \forall v_2 \in \mathcal{X}_2.
 \end{aligned} \tag{2.12}$$

It can be proven (see [QV99, Proposition 1.7.1]) that if u is the solution of W2.1 and u_i , σ_i , ψ are solutions of W2.4, then $u_i = u|_{\Omega_i}$, $\sigma_i = (\nabla_L u \cdot \mathbf{n}_i)_\Gamma$ (where $\nabla_L u \cdot \mathbf{n}_i$ indicates the conormal derivative of u with respect to the normal vector \mathbf{n}_i), and $\psi = u|_\Gamma$.

The weak formulation W2.2 we derived in Section 2.2 can be interpreted as a particular case of W2.4. Indeed, let us firstly restrict the space $H^{1/2}(\Gamma)$ to its embedded subset $H_{00}^{1/2}(\Gamma)$ and let us consider the particular case in which $\lambda = \sigma_1 = -\sigma_2 \in \Lambda = H_{00}^{-1/2}(\Gamma)$: then, the third equation in Eq. (2.12) is automatically satisfied for all choices of $\rho \in H_{00}^{1/2}(\Gamma)$. Moreover, subtracting the second and fourth equations evaluated at the same $\eta_1 = \eta_2 = \eta \in \Lambda$ yields

$$\langle \eta, \psi - u_1 \rangle_\Lambda - \langle \eta, \psi - u_2 \rangle_\Lambda = \langle \eta, u_2 - u_1 \rangle_\Lambda. \tag{2.13}$$

Obviously, this duality is well defined only if the trace of $u_2 - u_1$ belongs to $H_{00}^{1/2}(\Gamma)$. Therefore, we set $u \in \mathcal{X}_{00}$ such that $u_1 = u|_{\Omega_1}$ and $u_2 = u|_{\Omega_2}$; Eq. (2.13) can be then rewritten as $b(u, \eta) = 0$ for all $\eta \in \Lambda$, i.e., the second equation in Eq. (2.5). The first equation in Eq. (2.5) is found by adding the first and last equations in (2.12) tested for all $v \in \mathcal{X}_{00}$ such that $v_1 = v|_{\Omega_1}$, $v_2 = v|_{\Omega_2}$; observe that also in this case it is necessary to restrict the search space for v to \mathcal{X}_{00} , in order to ensure the well-posedness of $b(v, \lambda) = \langle \lambda, v_2 - v_1 \rangle_\Gamma$.

Although W2.2 and W2.4 are equivalent, their discretizations are not. Indeed, in the three-field method, it is necessary to define the discretizations of the variational spaces of σ_1 , σ_2 and ψ . In contrast, when discretizing W2.2, the third equation of W2.4 is not approximated but solved exactly and the second and fourth are merged into a single equation. As we have shown, setting $\sigma_1 = -\sigma_2$ is efficient because it allows us to automatically satisfy the third equation in Eq. (2.12), thus reducing the number of variables. Our approach limits to one the number of spaces to be discretized for each interface, thus allowing better control over the stability of the method.

2.5 Reinterpreting the inf-sup condition of the coupled problem

Problems W2.2 and W2.3 are saddle-point problems and, as such, their well-posedness is subject to specific conditions which we introduced in Section 1.4. It is worth contextualizing some of these concepts to the specific case of the Lagrange multipliers-based strategy for the coupling of PDEs, which we discussed in this chapter.

Let us assume that the continuous weak formulation is well-posed. As a matter of fact, we recall that if the space Λ is characterized as in (2.3), the continuous problem W2.2 has a unique solution [BDW99]. In the particular case of W2.3, the inf-sup condition requires the existence of a positive constant $\beta^{h\delta}$ such that

$$\inf_{\eta^\delta \in \Lambda^\delta} \sup_{v^h \in \mathcal{X}^h} \frac{b(v^h, \eta^\delta)}{\|v^h\|_{\mathcal{X}} \|\eta^\delta\|_{\Lambda}} = \inf_{\eta^\delta \in \Lambda^\delta} \sup_{v^h \in \mathcal{X}^h} \frac{\langle \eta^\delta, [v^h]_{\Gamma} \rangle_{\Lambda}}{\|v^h\|_{\mathcal{X}} \|\eta^\delta\|_{\Lambda}} \geq \beta^{h\delta}.$$

Loosely speaking, this is verified whenever the space of the Lagrange multiplier Λ^δ is small enough compared to \mathcal{X}^h . In light of the discretization strategy outlined in Section 2.3, the inf-sup condition is then associated with the number of basis functions of the subdomains sharing the interface—specifically, the basis functions with support on the interface itself—and the dimensionality of the Lagrange multiplier space N_{λ}^δ . Given a fixed mesh size, the stability of the problem is bound to deteriorate as N_{λ}^δ increases. Viceversa, it is possible to recover the well-posedness of the discrete problem whenever the inf-sup condition is violated by decreasing the mesh size in the subdomains. We validate these observations in the numerical experiments in Section 2.6.

It is also useful to rewrite the error estimates in Theorem 1.2 in terms of the variables u^h and λ^δ . These become

$$\|u - u^h\|_{\mathcal{X}} \leq C_1 \inf_{v^h \in \mathcal{V}^{h\delta}} \|u - v^h\|_{\mathcal{X}} + C_2 \inf_{\eta^h \in \Lambda^\delta} \|\lambda - \eta^h\|_{\Lambda} \quad (2.14)$$

$$\|\lambda - \lambda^\delta\|_{\Lambda} \leq C_3 \inf_{v^h \in \mathcal{V}^{h\delta}} \|u - v^h\|_{\mathcal{X}} + C_4 \inf_{\eta^\delta \in \Lambda^\delta} \|\lambda - \eta^\delta\|_{\Lambda}, \quad (2.15)$$

and

$$\inf_{v^h \in \mathcal{V}^{h\delta}} \|u - v^h\|_{\mathcal{X}} \leq \left(1 + \frac{\kappa_b}{\beta^{h\delta}}\right) \inf_{v^h \in \mathcal{X}^h} \|u - v^h\|_{\mathcal{X}}. \quad (2.16)$$

The constants in Eq. (2.14) and Eq. (2.15) take the values

$$C_1 = \left(1 + \frac{\kappa_a}{\alpha^h}\right), \quad C_2 = \frac{\kappa_b}{\alpha^h}, \quad C_3 = C_1 \frac{\kappa_a}{\beta^{h\delta}}, \quad C_4 = \left(1 + \frac{\kappa_b}{\beta^{h\delta}} + \frac{\kappa_a \kappa_b}{\alpha^h \beta^{h\delta}}\right),$$

where κ_a and κ_b are the continuity constants of $a(\cdot, \cdot)$ and $b(\cdot, \cdot)$, and α^h is the coercivity constant of $a(\cdot, \cdot)$.

These estimates show that whenever the space of Lagrange multipliers is rich enough (namely, the second term in Eq. (2.14) becomes negligible compared to the first one), the approximation of u is essentially bounded by the best approximation error in \mathcal{X} . However, as we have already discussed, this richness may lower the inf-sup constant $\beta^{h\delta}$ and therefore loose the approximation (2.16). It is, therefore, important to find the correct balance.

We remark that from Eq. (2.7) we have that increasing the size of the Lagrange multipliers space is equivalent to lowering the size of $\mathcal{V}^{h\delta}$ and, consequently, the supremum in its right-hand side. The two error estimates in (2.7) and (2.14) are therefore equivalent ways of expressing the fact that, if the continuity over the interface Γ is enforced strongly enough, the error converges to zero as the error due to the spatial discretization in \mathcal{X}^h . As we show in the next section, we are able to recover the usual convergence orders for u with respect to the mesh size h when using finite elements and IGA.

Finally, we observe that the inf-sup condition is also linked to the stability with respect to the problem data. The estimates derived in Lemma 1.1 for W2.3 read

$$\begin{aligned} \|u^h\|_{\mathcal{X}} &\leq \frac{1}{\alpha^h} \|f\|_{\mathcal{X}'}, \\ \|\lambda^\delta\|_{\Lambda} &\leq \frac{1}{\beta^{h\delta}} \left(1 + \frac{\kappa_a}{\alpha^h}\right) \|f\|_{\mathcal{X}'}. \end{aligned}$$

We highlight that, in the specific case of the primal hybrid method for coupling that we consider here, the inf-sup constant $\beta^{h\delta}$ affects only the stability of the Lagrange multiplier λ^δ because the constraint is $b(v^h, \eta^\delta) = 0$ for all $\eta^\delta \in \Lambda^\delta$ (hence, by using the notation of Eq. 1.15, $g = 0$).

2.6 Numerical results

In this section, we focus on the performance of the method presented in Section 2.2 on two-dimensional and three-dimensional problems. The numerical simulations in two dimensions are performed in MATLAB. The three-dimensional results in Section 2.6.2 and Section 2.7 have been obtained using LifeV [BDF⁺17].

For all the simulations, we employ standard piecewise polynomial Lagrangian basis functions defined over suitable triangulations in the subdomains or B-Splines basis functions, as presented in Section 2.6.3. Regarding the choice of basis functions for Λ^δ , we already anticipated in Section 2.2 that, in this chapter, we investigate the possibility of using low-frequency Fourier basis functions built on the interface Γ .

Remark 2.4. All the linear systems arising in the numerical simulations, except those in Section 2.6.2, are solved by means of the backslash operator in MATLAB without any preconditioning.

2.6.1 Choice of basis functions for the Lagrange multipliers

In this section, we focus on the discretization of one dimensional interfaces. Given an interface with length L , we consider $\xi_1 = 1$ and, for $i = 1, \dots, N_\omega^\delta$

$$\xi_{2i}(s) = \sin(\omega_i \pi s), \quad \xi_{2i+1}(s) = \cos(\omega_i \pi s), \quad (2.17)$$

where s is the arc length of the interface Γ , $\omega_i = i/L$, and N_ω^δ is the number of considered frequencies; it holds that $N_\lambda^\delta = 2N_\omega^\delta + 1$. With this definition, the set $\{\xi_i\}_{i=1}^{N_\lambda^\delta}$ forms an orthogonal basis with respect to the $L^2(0, 2L)$ scalar product. We choose to employ such basis—instead of the standard Fourier basis orthogonal (or orthonormal) with respect to the $L^2(0, L)$ scalar product—because, by considering basis functions with periodicity L , we would impose an unnecessary periodicity constraint, in particular, the equality of the functions in Λ^δ and their derivatives at the extrema of Γ . As a result, we empirically observed that by employing the standard $L^2(0, L)$ orthonormal Fourier basis functions the optimal convergence of the FE method is retrieved for larger values of N_λ^δ compared to the choice in Eq. (2.17). However, employing nonorthonormal basis functions (2.17) has a dramatic influence on the condition number of the resulting linear system, which has exponential growth when the number of basis functions on the interface increases; see Fig. 2.3 (left).

In order to retain the convergence order attained by using the Fourier modes in Eq. (2.17) and, at the same time, control the condition number of the system, we propose an orthonormalization strategy based on the Gram–Schmidt algorithm or, equivalently, on the QR decomposition [Ruh83]. Even though the coefficients of the orthonormal basis generated by (2.17) with these algorithms could be analytically derived, their exact expression quickly becomes complex with N_λ^δ becoming large. With our approach, we aim at obtaining an approximation of such coefficients relying on a fine sampling of the basis functions on the interval $(0, L)$. We remark that, in addition to allowing us to effortlessly compute a large number of orthonormal basis functions, our approach has the advantage to be general enough to be applied to any set of nonorthonormal basis functions.

Let $\{\xi_i\}_{i=1}^{N_\lambda^\delta}$ be the set of nonorthonormal basis functions defined on Γ . Moreover, let $\{x_i\}_{i=1}^{N_s}$ be distinct sample points distributed over the interval $(0, L)$, where L still denotes the length of the interface. We now introduce the functions $\{\kappa_i\}_{i=1}^{N_s}$, which we identify with the set of standard Lagrangian piecewise linear basis functions centered at each sample point x_i , and the associated mass matrix $M_{ij} = \int_\Gamma \kappa_i \kappa_j$. Furthermore, let $V = [\mathbf{v}_1 | \dots | \mathbf{v}_{N_\lambda^\delta}] \in \mathbb{R}^{N_s \times N_\lambda^\delta}$ be the matrix of evaluations of the basis functions at the sample points, namely $V_{ij} = \xi_j(x_i)$. We remark that, for each $i = 1, 2, \dots, N_\lambda^\delta$, we have $\|\xi_i\|_{L^2(0, L)}^2 \approx \mathbf{v}_i^T M \mathbf{v}_i$. Since M is a positive-definite matrix, it admits a unique Cholesky decomposition and there exists $C \in \mathbb{R}^{N_s \times N_s}$ such that $M = C^T C$.

Let us now consider the unit matrix $Q \in \mathbb{R}^{N_s \times N_\lambda^\delta}$ and the upper triangular matrix

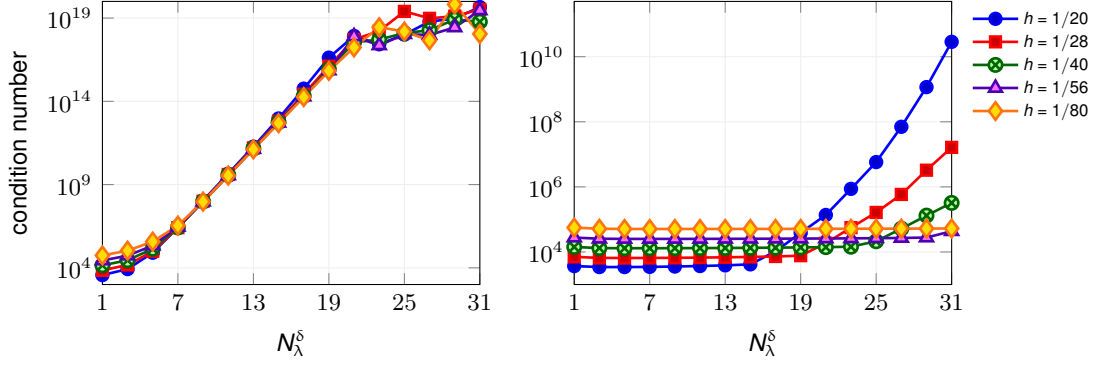


Figure 2.3 – Condition number of the discretized system for the Poisson problem on two subdomains (see Section 2.6.2) vs number of basis functions for the Lagrange multiplier space N_λ^δ . On the left, we consider the nonorthonormal basis functions in Eq. (2.17), while on the right we consider their orthonormalization.

$R \in \mathbb{R}^{N_\lambda^\delta \times N_\lambda^\delta}$ such that the truncated QR decomposition of CV reads

$$CV = QR.$$

By construction, we have

$$(C^{-1}Q)^T MC^{-1}Q = Q^T C^{-T} C^T C C^{-1}Q = I,$$

thus, the columns of $C^{-1}Q$ represent evaluations at the sample points of functions orthonormal on $(0, L)$ with respect to the L^2 product. The matrix R performs the change of variable from the frame of reference of the new orthonormal basis functions to the frame of reference of the nonorthonormal basis functions. If the sampling is sufficiently fine, we speculate that the elements of matrix R^{-1} will approximate the coefficients which are computed by applying the Gram–Schmidt algorithm to the continuous nonorthonormal basis functions $\{\xi_i\}_{i=1}^{N_\lambda^\delta}$ and, in particular, that

$$\bar{\xi}_i^\delta = \sum_{j=1}^{N_\lambda^\delta} \xi_j^\delta R_{ji}^{-1}.$$

From a practical perspective, matrix R^{-1} is suitable to compute coupling matrix $\bar{B}^{h\delta}$ with respect to the orthonormal Fourier basis functions, knowing the coupling matrix computed without orthonormalization $B^{h\delta}$. Indeed, we have

$$\bar{B}_{ij}^{h\delta} = \int_\Gamma \bar{\xi}_i^\delta \varphi_j^h = \sum_{k=1}^{N_\lambda^\delta} \left(\int_\Gamma \xi_k^\delta \varphi_j^h \right) R_{ki}^{-1} = \sum_{k=1}^{N_\lambda^\delta} B_{kj}^{h\delta} R_{ki}^{-1},$$

Algorithm 2.1 Transformation of coupling matrices.

Input: Coupling matrix $B^{h\delta}$, computed with $\{\xi_i\}_{i=1}^{N_\lambda^\delta}$ (a set of nonorthonormal Lagrange multiplier basis functions), $N_s \in \mathbb{N}$

Output: Coupling matrix $\bar{B}^{h\delta}$, computed with $\{\hat{\xi}_i\}_{i=1}^{N_\lambda^\delta}$ (a set of orthonormal Lagrange multiplier basis functions)

- 1: Sample $\{x_i\}_{i=1}^{N_s}$ points equally distributed on the interface Γ
 - 2: Construct $\{\kappa_i\}_{i=1}^{N_s}$ (linear Lagrangian FE functions centered in x_i)
 - 3: Assemble $M_{ij} = \int_\Gamma \kappa_i \kappa_j$ and compute the Cholesky decomposition $M = C^T C$
 - 4: Assemble $V = [\mathbf{v}_1 | \dots | \mathbf{v}_{N_\lambda^\delta}] \in \mathbb{R}^{N_s \times N_\lambda^\delta}$ such that $V_{ij} = \xi_j(x_i)$.
 - 5: Compute the QR decomposition of $CV = QR$
 - 6: Compute $\bar{B}^{h\delta} = R^{-T} B^{h\delta}$
-

or equivalently $\bar{B}^{h\delta} = R^{-T} B^{h\delta}$. Observe that, being that R is an upper triangular matrix, the application of R^{-T} is performed at negligible cost; furthermore, matrix R^{-T} depends only on the choice of the nonorthonormal basis functions and can be then computed a priori. The orthonormalization algorithm is summarized in Algorithm 2.1. We remark that, with this approach, the orthonormal basis functions are never explicitly computed. Moreover, since the discrete space is exactly the same, the approximation properties and the convergence orders are not changed. Fig. 2.3 (right) shows that, after the orthonormalization of the Fourier basis functions (2.17) by the algorithm we presented, the system is more stable and the condition number increases with the number of Fourier basis functions N_λ^δ dependently on the refinement level of the mesh h .

2.6.2 The Poisson problem on two-way partitioned domains

Results in two dimensions on the unit square

Let us consider the global Poisson problem (2.2) on the domain $\Omega = (0, 1)^2$, where we take f such that $u = 100xy(1-x)(1-y)\sin(1/3 - xy^2)$ is the exact solution. We divide Ω into $\Omega_1 = (0, 0.5) \times (0, 1)$ and $\Omega_2 = (0.5, 1) \times (0, 1)$.

We numerically solve the problem on Ω_1 and Ω_2 by employing structured triangular conforming and nonconforming meshes with varying mesh size h . The conforming meshes are obtained by subdividing the domain in the x - and y -direction in the same number of elements. On the other hand, the nonconforming meshes are built by taking $N + 1$ elements in the y -direction of Ω_2 , N being the number of elements in the y -direction in Ω_1 as well as the total number of elements in the x -direction.

Fig. 2.4 shows how the solutions on Ω_1 and Ω_2 obtained with a conforming mesh with $N = 20$ elements in each direction change with respect to the number of basis functions on the interface. The results are obtained with quadratic Lagrangian polynomials in both subdomains. From the contour lines plots in the top row, it appears that the two solutions

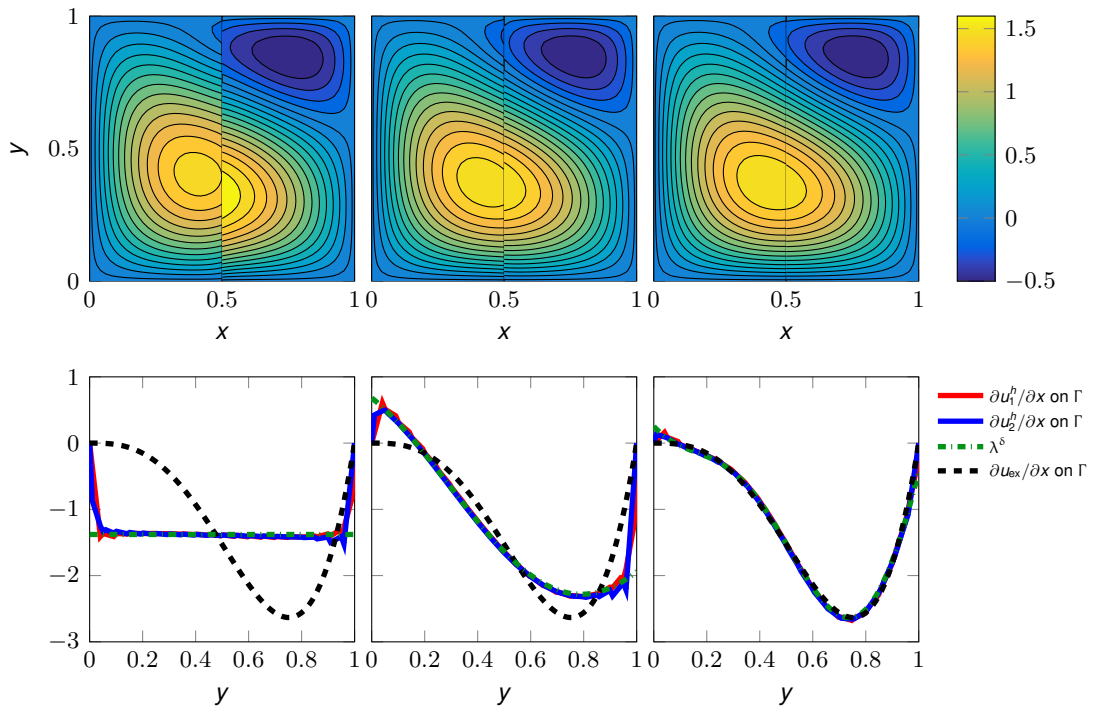


Figure 2.4 – Contour lines of the solution (top row) and derivatives of the solution at both sides of the interface, approximated Lagrange multiplier and exact derivative at the interface (bottom row) when $N_\lambda^\delta = 1$ (left column), $N_\lambda^\delta = 3$ Fourier modes (middle column) and $N_\lambda^\delta = 5$ Fourier modes (right column) are used to characterize the space Λ^δ .

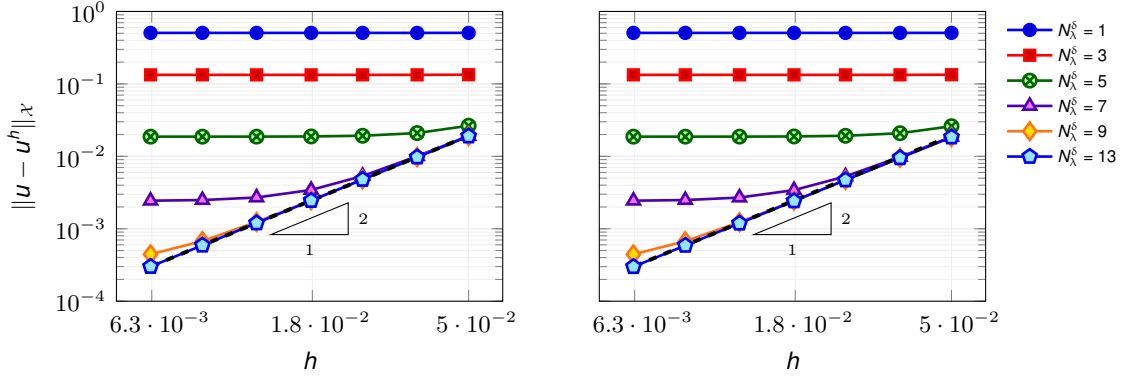


Figure 2.5 – Convergence of error in broken norm with respect to the mesh size h and number of basis functions on the interface N_λ^δ with conforming (left) and nonconforming meshes (right). The black dashed line shows the convergence of the error with a conforming mesh with comparable mesh size.

match quite accurately at the interface with 5 Fourier basis functions ($N_\omega^\delta = 2$). In the second row of Fig. 2.4, we plot the approximation by finite differences of the derivative of the solution with respect to x in the two domains, which is equal to the normal derivative of u_1^h and to the opposite of the normal derivative on u_2^h on Γ respectively. Observe that, as we already highlighted in Remark 2.2, the Lagrange multiplier λ^δ takes the role of the normal derivative of u^h on Γ .

Let us now address the convergence of the global solution to the exact one with respect both to the mesh size h and the number of basis functions on the interface N_λ^δ . To this end, we consider meshes with a total number of elements in the x -direction $N = 20, 28, 40, 56, 80, 114, 160$, and we solve the problems by employing quadratic Lagrangian basis functions in both subdomains. Fig. 2.5 depicts the error decay in \mathcal{X} -norm (the broken norm) with respect to h , as well as the convergence of the error obtained by solving the problem on a single conforming mesh of Ω (in black dashed line). When employing both conforming and nonconforming meshes, the error is optimal—in the sense that we recover the theoretical order of convergence h^2 of quadratic finite elements for the H^1 -error—when N_λ^δ is large enough, e.g., $N_\lambda^\delta \geq 13$. In this case, the error is equal to the one given by the single domain approach. If N_λ^δ is too small, on the contrary, the solution is unable to converge to the exact solution with h and reaches a stagnation point. We remark that this result is perfectly consistent with Strang’s second lemma (2.7) and with the estimate (2.14): whenever the space of Lagrange multiplier is rich enough (which is equivalent to requiring that $\mathcal{V}^{h\delta}$ be a good approximation of \mathcal{V}), the best approximation error of the interpolation is recovered.

Remark 2.5. In our numerical simulations with nonconforming meshes, we observed that instabilities arise when using coarse meshes and low-order quadrature rules for the computations of the approximate integrals of $B_1^{h\delta}$ and $B_2^{h\delta}$ in Eq. (2.10). When considering, for example, $h = 1/20, 1/28, 1/40$ and 2 Gauss quadrature nodes, the error

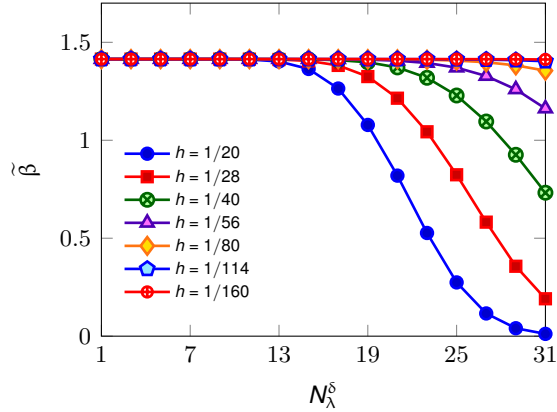


Figure 2.6 – Decaying of the inf-sup constant $\tilde{\beta}$ with respect to N_λ^δ .

starts increasing when N_λ^δ becomes large (e.g., $N_\lambda^\delta \geq 15$). By increasing the order of the quadrature rule and choosing 4 Gauss quadrature nodes, this issue is completely fixed. We did not encounter stability problems when using conforming meshes, even with low-order quadrature rules.

Fig. 2.6 shows the variation of the estimate of the inf-sup constant $\tilde{\beta}$ —computed as the square root of the minimum eigenvalue of the generalized eigenvalue problem (1.19), as described in Section 2.5—when the number of basis functions on the interface changes; the estimate refers to the simulation of the Poisson equations with conforming meshes and quadratic polynomial basis functions. In Fig. 2.6, each curve presents a plateau phase in which the inf-sup constant stays approximately constant at around $\tilde{\beta} \approx 1.41$ with the increment of N_λ^δ . The width of such a plateau phase increases when h becomes smaller. Indeed, we observe that $\tilde{\beta}$ starts decreasing for smaller values of N_λ^δ when the meshes are coarser and that, conversely, for finer meshes the inf-sup constant varies relatively little in the range $N_\lambda^\delta \in (1, 31)$. We remark that, combined with the condition number shown in Fig. 2.3 (right), this result ensures that, for each refinement level, we are able to obtain the optimal convergence of the FE method when the basis functions are orthonormal. Indeed, refining the mesh has the effect of both increasing the range of stability of the linear system—see Fig. 2.3 (right)—and increasing the number of basis functions on the interface that can be employed without reaching the fast decaying region of $\tilde{\beta}$ in Fig. 2.6. With regard to this last point, we recall that it is important to prevent the inf-sup constant to become too small because it appears at the denominator of the constant multiplying the best approximation errors in u and on λ in the error estimates (2.14) and (2.15).

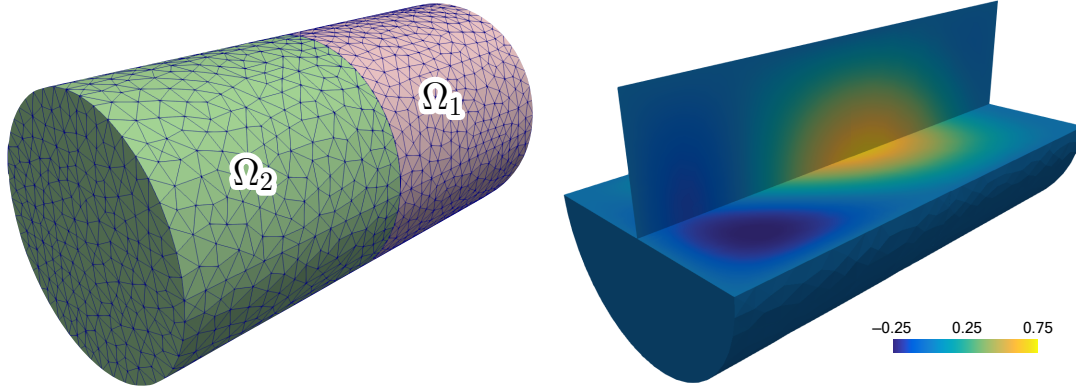


Figure 2.7 – On the left, subdomains of the cylinder with corresponding meshes (sizes: $h_1 = 1.61 \cdot 10^{-1}$ and $h_2 = 1.76 \cdot 10^{-1}$). On the right, the solution u obtained with $N_\lambda^\delta = 28$.

Results in three dimensions on a cylinder

We consider the Poisson problem (2.2) on a three-dimensional domain. In particular, we take the cylinder

$$\Omega = \{\mathbf{x} = [x, y, z]^T \in \mathbb{R}^3 : x \in (0, 2), y^2 + z^2 < R^2\},$$

where $R = 1/2$, and we set the forcing term f such that

$$u = 25 \left(\frac{x}{2} - \frac{\exp(x) - 1}{\exp(2) - 1} \right) (R^2 - y^2 - z^2) \sin(x - y) \sin(x - z)$$

is the exact solution.

We consider subdomains Ω_1 and Ω_2 obtained by cutting the domain with the plane corresponding to $x = 1$. On each subdomain, we define a family of triangulations \mathcal{T}_1^h and \mathcal{T}_2^h composed of tetrahedra such that $\mathcal{T}^h = \mathcal{T}_1^h \cup \mathcal{T}_2^h$ is not conforming at the interface. Figure 2.7 (left) displays one of such global triangulations of the computational domain. We consider FE spaces composed of standard Lagrangian linear and quadratic basis functions on Ω_1 and Ω_2 respectively.

Concerning the discretization of the Lagrange multipliers space, in this test case we exploit the fact that the interface Γ is a circle (before the geometric discretization of Ω). We define a polar system of coordinates centered in $\mathbf{x}_0 = (1, 0, 0)$ on the plane $x = 1$ and we denote by $r(y, z) = \sqrt{y^2 + z^2}$ and $\theta(y, z)$ the two parametric variables representing the radial distance from point \mathbf{x}_0 and the angular variable, respectively. Function $\theta(\cdot, \cdot)$ is defined as $\theta(y, z) = \arctan(z/y)$ if $y \geq 0$ and $\theta(y, z) = \arctan(z/y) + \pi$ if $y < 0$. We define in each of the parametric directions a set of basis functions. In the radial direction, we take $\xi_{i+1,r}^\delta(r) = \cos(\pi ir/4R)$, for $i = 0, \dots, N_{\omega,r}^\delta$; we employ only cosines

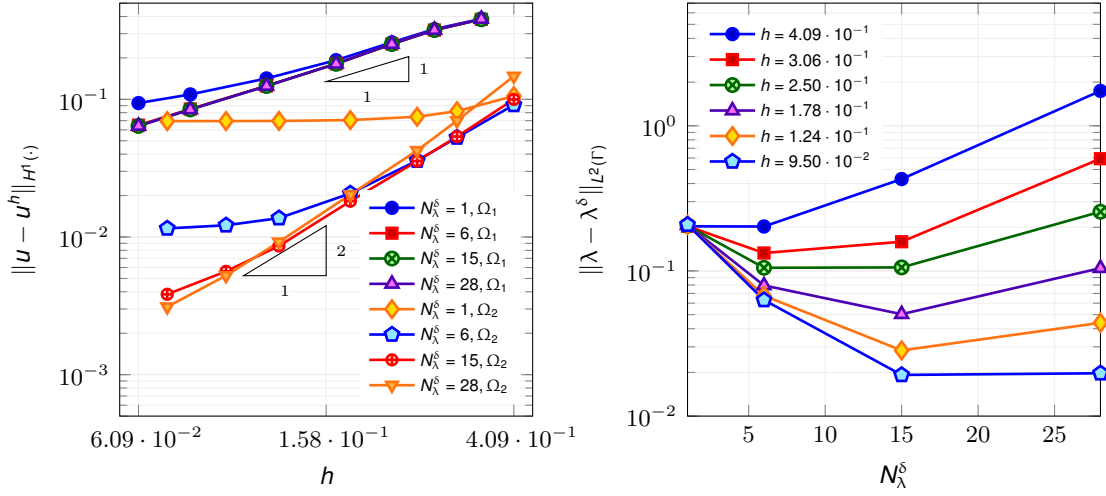


Figure 2.8 – Convergence of the H^1 -error in the solution u in Ω_1 and Ω_2 (left) and convergence of the L^2 -error in the Lagrange multiplier λ (right). In Ω_1 and Ω_2 we use linear and quadratic finite elements, respectively.

with varying frequencies in order to avoid discontinuities in the derivatives at $r = 0$. In the angular direction, we take $\xi_{1,\theta}(\theta) = 1$, $\xi_{2i,\theta}(\theta) = \sin(i\pi\theta)$ and $\xi_{2i+1,\theta}(\theta) = \cos(i\pi\theta)$, with $i = 1, \dots, N_{\omega,\theta}^\delta$. The two-dimensional basis functions on the interface are then defined by means of the tensor product of these two sets of univariate basis functions, namely we set $\xi_{ij}^\delta(r, \theta) = \xi_{i,r}^\delta(r)\xi_{j,\theta}^\delta(\theta)$ and the total number of such basis functions is $N_\lambda^\delta = (N_{\omega,r}^\delta + 1)(2N_{\omega,\theta}^\delta + 1)$. In our tests, we consider $N_{\omega,r}^\delta = N_{\omega,\theta}^\delta$ to simplify the analysis of the results, but the numbers of basis functions in the two parametric directions are a priori independent.

Figure 2.8 shows, on the left, the convergence of the H^1 -error of u in Ω_1 and Ω_2 and, on the right, the convergence of the L^2 -error in the Lagrange multiplier λ . If N_λ^δ is large enough, consistently with what observed in Section 2.6.2, the error in the solution converges in each subdomain with the theoretical rate of the employed FE space: order one in Ω_1 , where linear FE basis functions are used, and order two in Ω_2 , which has been discretized by using quadratic FE basis functions. If, on the other hand, the Lagrange multiplier space is not sufficiently rich, the error does not converge when h goes to zero. We remark that, being that the error in Ω_1 is larger than that in Ω_2 by one order of magnitude, the theoretical convergence of order one is achieved for smaller N_λ^δ in that subdomain. The L^2 -error in the Lagrange multiplier, however, is not monotonic with respect to the number of basis functions N_λ^δ ; see Figure 2.8 (right). In particular, for larger mesh sizes the error rapidly increases with N_λ^δ , whereas for smaller mesh sizes the error reaches a minimum for N_λ^δ strictly smaller than $N_\lambda^\delta = 28$. This effect is clearly related to the inf-sup stability of the problem and to the fact that, as we already noticed in Section 2.6.2, the inf-sup constant β rapidly goes to zero with N_λ^δ if the mesh size is too large. The effect of the inf-sup condition being violated is also visible in the plot on

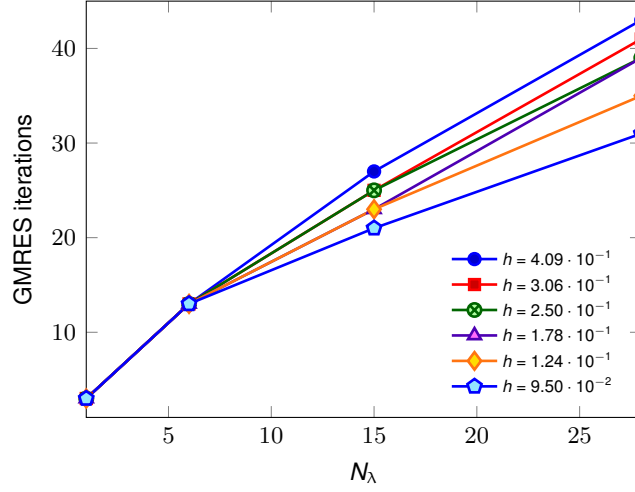


Figure 2.9 – GMRES iterations for solving the (preconditioned) linear system on the problem presented in Section 2.6.2 vs number of basis functions for the Lagrange multiplier space.

the left, where, for coarse meshes, the error in Ω_2 increases with the addition of basis functions for Λ^δ ; however, the increase in the error in the solution is not as evident as that on the Lagrange multiplier. One plausible explanation for the inf-sup stability of the problem having a larger impact on the accuracy of λ than the accuracy of u is that, by combining Eq. (2.15) and Eq. (2.16), we see that β^2 appears at the denominator in the error estimate of the Lagrange multiplier, whereas the constants in the error estimate for u in Eq. (2.14) depend on $1/\beta$.

For this test case we solve the linear system by the GMRES method with tolerance 10^{-8} . We also employ the block diagonal preconditioner

$$P = \begin{bmatrix} A_1^h & & \\ & A_2^h & \\ & & M^\delta \end{bmatrix},$$

where A_1^h and A_2^h are the stiffness matrices in the two subdomains and $M_{ij}^\delta = \int_\Gamma \xi_j^\delta \xi_i^\delta$ is the mass matrix of Λ^δ , which is computed by selecting the finest mesh on the interface Γ for the numerical integration. During the application of the preconditioner, each diagonal block is solved by an LU decomposition. Fig. 2.9 shows the varying of the number of GMRES iterations with respect to N_λ^δ and the mesh size. Clearly, the number of iterations increases by enriching the basis functions on the interface. Nevertheless, it also appears that the number of iterations decreases when the mesh size becomes smaller. This is a consequence of the fact that, for a fixed N_λ^δ , the conditioning and the inf-sup stability of the coupling improve when h becomes smaller (as shown in Fig. 2.3 and Fig. 2.6).

We consider a more involved preconditioning strategy, which also takes advantage of the

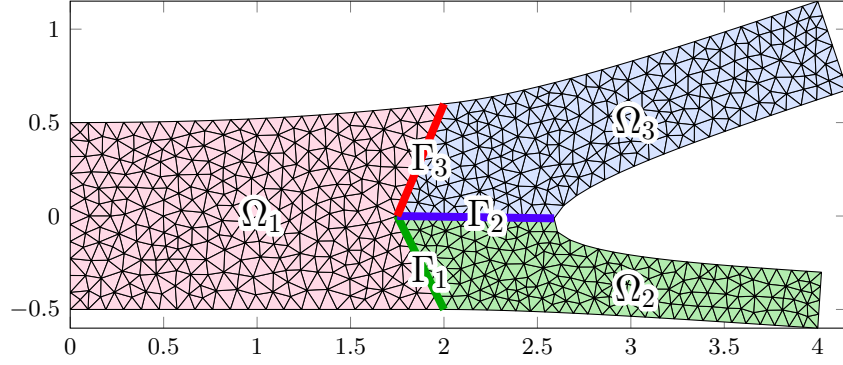


Figure 2.10 – Three-way partition of the model bifurcation. The displayed mesh corresponds to the size $h = 1.32 \cdot 10^{-1}$.

saddle-point structure of the global problem, in Chapter 4.

2.6.3 The steady Navier–Stokes equations on three-way partitioned domains

In this section, we test the flexibility of our method by solving the steady Navier–Stokes equations in domain $\Omega \in \mathbb{R}^2$ depicted in Fig. 2.10, which represents an idealized bifurcation in an arterial tree. The equations read

$$\begin{aligned}
 \rho_f(\mathbf{u} \cdot \nabla)\mathbf{u} - \nabla \cdot \sigma_f &= \mathbf{f} && \text{in } \Omega, \\
 \nabla \cdot \mathbf{u} &= 0 && \text{in } \Omega, \\
 \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_D, \\
 \sigma_f(\mathbf{u}, p)\mathbf{n} &= \mathbf{h} && \text{on } \Gamma_N,
 \end{aligned} \tag{2.18}$$

where the symbols have the same meaning as in Eq. (1.1). In all the test cases we consider, we take $\Gamma_D = \Gamma_{\text{in}} \cup \Gamma_w$, where $\Gamma_{\text{in}} = \{0\} \times (-r_{\text{in}}, r_{\text{in}})$ with $r_{\text{in}} = 0.5$ is the inflow boundary on the left and Γ_w is the wall of the vessel. The Neumann boundary $\Gamma_N = \Gamma_{\text{out}_1} \cup \Gamma_{\text{out}_2}$ is composed of the two outflows on the right. The Dirichlet boundary data is $\mathbf{g} = (y^2/r_{\text{in}}^2 - 1)U\mathbf{n}$ on Γ_{in} (where \mathbf{n} is the outward normal to the inflow boundary), namely we impose a longitudinal parabolic inflow with maximum velocity magnitude U , and $\mathbf{g} = 0$ on Γ_w (no-slip condition). On the outflows Γ_N we set homogeneous Neumann conditions, i.e., $\mathbf{h} = 0$. In the numerical simulations, we set $\mu = 1$, $U = 1$ and $\mathbf{f} = 0$.

By following steps analogous to the ones presented in Section 1.2, it is straightforward to derive the weak formulation of the steady Navier–Stokes equations.

Chapter 2. Non-conforming coupling of PDEs

(W2.5) given \mathbf{f} , \mathbf{g} , and \mathbf{h} regular enough, find $(\mathbf{u}, p) \in \mathcal{V} \times \mathcal{Q}$, such that

$$\begin{aligned} a(\mathbf{u}, \mathbf{v}) + c(\mathbf{u}, \mathbf{u}, \mathbf{v}) + b(\mathbf{v}, p) &= \langle \mathbf{f}, \mathbf{v} \rangle + \langle \mathbf{h}, \mathbf{v} \rangle_{\Gamma_N} & \forall \mathbf{v} \in \mathcal{V}_0, \\ b(\mathbf{u}, q) &= 0 & \forall q \in \mathcal{Q}, \end{aligned}$$

and such that $\gamma|_{\Gamma_D} \mathbf{u} = \mathbf{g}$. The bilinear forms, trilinear forms and functional spaces are defined as in W1.1.

Let us consider the finite dimensional approximations of velocity $\mathbf{u}^h = \sum_{i=1}^{N_u^h} \boldsymbol{\varphi}_i^h u_i^h$ and pressure $p^h = \sum_{i=1}^{N_p^h} \phi_i^h p_i^h$. Upon discretization of Eq. (2.18) by a Galerkin method—in this section, we consider the FE method and IGA—the system can be rewritten in algebraic form as

$$\underbrace{\begin{bmatrix} K^h + C^h(\mathbf{u}^h) & (D^h)^T \\ D^h & \end{bmatrix}}_{\mathcal{A}^h(\underline{\mathbf{w}}^h)} \underbrace{\underline{\mathbf{w}}^h}_{\mathbf{F}^h} = \begin{bmatrix} \mathbf{r}^h \\ \mathbf{0} \end{bmatrix}, \quad (2.19)$$

where $\underline{\mathbf{w}}^h = [u_1^h, \dots, u_{N_u^h}^h, p_1^h, \dots, p_{N_p^h}^h] \in \mathbb{R}^{N_u^h + N_p^h}$ is a vector containing N_u^h dofs for the velocity and N_p^h dofs for the pressure, K^h is the stiffness matrix, $C^h(\mathbf{u}^h)$ is the convective term matrix, D^h is the divergence term matrix, and $\mathbf{r}_i^h = \langle \mathbf{f}, \boldsymbol{\varphi}_i^h \rangle + \langle \mathbf{h}, \boldsymbol{\varphi}_i^h \rangle_{\Gamma_N}$.

Let us now introduce a partition of the domain Ω into three subdomains $\{\Omega_i\}_{i=1}^3$ separated by three interfaces $\{\Gamma_i\}_{i=1}^3$, as shown in Fig. 2.10. Since the subdomains are topologically arranged as the ones presented in Section 2.3.1, the matrix of the global system maintains the structure of that in Eq. (2.11). The global system reads

$$\begin{bmatrix} \mathcal{A}_1^h(\underline{\mathbf{w}}_1) & & & (B_{11}^{h\delta})^T & & -(B_{31}^{h\delta})^T \\ & \mathcal{A}_2^h(\underline{\mathbf{w}}_2) & & -(B_{12}^{h\delta})^T & -(B_{22}^{h\delta})^T & \\ & & \mathcal{A}_3^h(\underline{\mathbf{w}}_3) & & (B_{23}^{h\delta})^T & (B_{33}^{h\delta})^T \\ B_{11}^{h\delta} & -B_{12}^{h\delta} & & & & \\ & -B_{22}^{h\delta} & B_{23}^{h\delta} & & & \\ -B_{31}^{h\delta} & & B_{33}^{h\delta} & & & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{w}}_1^h \\ \underline{\mathbf{w}}_2^h \\ \underline{\mathbf{w}}_3^h \\ \underline{\boldsymbol{\lambda}}_1^h \\ \underline{\boldsymbol{\lambda}}_2^h \\ \underline{\boldsymbol{\lambda}}_3^h \end{bmatrix} = \begin{bmatrix} \mathbf{F}_1^h \\ \mathbf{F}_2^h \\ \mathbf{F}_3^h \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \quad (2.20)$$

where \mathcal{A}_i^h and \mathbf{F}_i^h are the matrix and right-hand side of system (2.19) referred to the i^{th} subdomain, whereas $B_{ij}^{h\delta}$ is—as explained in Section 2.3.1—the matrix discretizing the coupling of Ω_j and interface Γ_i . For $i = 1, 2, 3$, the vector $\underline{\boldsymbol{\lambda}}_i^h$ contains the Lagrange multipliers corresponding to the i^{th} interface Γ_i .

Remark 2.6. When applied to the Navier–Stokes equations in two dimensions, the method requires assigning to each interface two sets of basis functions discretizing the two components of the normal stress. To see why this is the case, consider the situation in which Ω is subdivided into Ω_1 and Ω_2 ; let us denote as always the interface of the two partitions by Γ . Multiplying the momentum equation by a test function $\mathbf{v} \in [H_{\Gamma_D}^1(\Omega)]^2$

and integrating by parts on Ω_1 leads to

$$\mu_f \int_{\Omega_1} \nabla \mathbf{u} : \nabla \mathbf{v} - \int_{\Omega_1} p \nabla \cdot \mathbf{v} - \int_{\Gamma} \sigma_f(\mathbf{u}, p) \mathbf{n} \cdot \mathbf{v} = \int_{\Omega_1} \mathbf{f} \cdot \mathbf{v} + \int_{\partial\Omega_1 \cap \Gamma_N} \mathbf{h} \cdot \mathbf{v}.$$

The integral on Γ is the coupling term. Each of the two components of the normal stress $\sigma_f(\mathbf{u}, p) \mathbf{n}$ must be discretized by a set of basis functions. In this thesis, we choose for simplicity to use the same set for each component of the normal stress.

We consider again Fourier basis functions for the approximation of the normal stresses. Since, as explained in Remark 2.6, we need two Lagrange multipliers to represent each normal stress, the number of basis functions on the i^{th} interface Γ_i is found as $N_{\lambda,i}^\delta = 2(2N_{\omega,i}^\delta + 1)$, where $N_{\omega,i}^\delta$ is the number of frequencies. We consider the same number of frequencies on each of the interfaces, i.e., $N_\omega^\delta = N_{\omega,1}^\delta = N_{\omega,2}^\delta = N_{\omega,3}^\delta$. Therefore, the total number of basis functions for the Lagrange multiplier space is simply given by $N_\lambda^\delta = \sum_{i=1}^3 N_{\lambda,i}^\delta = 6(2N_\omega^\delta + 1)$. This choice is uniquely made to reduce the number of free parameters and to ease the analysis of the numerical results. Similarly to what noted in Section 2.6.2, where we employed the same number of frequencies in the two parametric directions of the two-dimensional interface, one could equally decide to vary the number of basis functions on each interface, if such decision was motivated by the application at hand.

Remark 2.7. The discretized (nonlinear) Navier–Stokes equations (2.20) are solved by means of the Newton–Raphson method with stopping tolerance 10^{-8} . We observe that employing orthogonal basis functions on the interfaces in this context is beneficial because this allows us to control the condition number of the Jacobian system. In the numerical simulations, however, we were able to obtain the optimal convergences (i.e., increase N_λ^δ as much as required) without orthonormalizing the basis functions of the Lagrange multipliers. The nonlinear solver always converged in three iterations or less.

FE-FE coupling

We focus on the convergence of the problem when the discrete spaces in the three subdomains are built by means of FE functions. We define a family of globally nonconforming triangulations \mathcal{T}^h characterized by the mesh size h , namely the maximum edge length over Ω . We remark that, since the exact geometry features curved boundaries, these triangulations are associated with a geometric error; this is not the case when some of the subdomains are discretized by IGA, as in Section 2.6.3. Differently from what is done in Section 2.6.2, we only focus on nonconforming meshes, and we exclusively employ the inf-sup stable Taylor–Hood [HT74] elements with quadratic basis functions for the velocity and linear basis functions for the pressure (i.e., P2-P1 elements). To compute the error of the solution obtained with the nonconforming method, we consider a reference solution \mathbf{u} obtained with P3-P2 basis functions on a fine conforming mesh

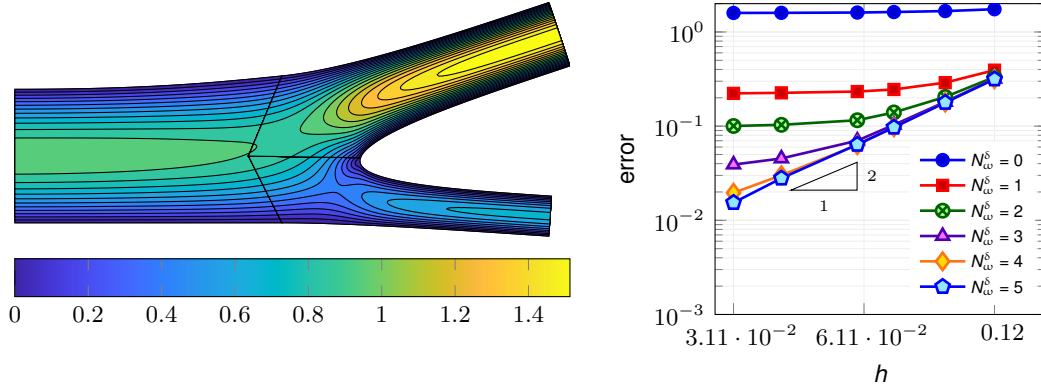


Figure 2.11 – On the left, solution of the Navier–Stokes computed on a mesh with size $h = 7.13 \cdot 10^{-2}$ and employing $N_{\omega}^{\delta} = 4$ frequencies (corresponding to $N_{\lambda}^{\delta} = 54$ basis functions for Λ^{δ}). On the right, convergence of the error (as defined in Eq. (2.21)) with respect to the mesh size h , for different values of N_{λ}^{δ} .

($h_{\text{fine}} = 1.12 \cdot 10^{-2}$, corresponding to 1'509'020 and 335'887 dofs for velocity and pressure respectively).

Fig. 2.11 (left) displays the contour lines of the solution corresponding to mesh size $h = 7.13 \cdot 10^{-2}$ and $N_{\omega}^{\delta} = 4$. For this discretization of the Lagrange multiplier space, the contour lines at both sides of each interface match quite accurately, which indicates that the continuity of the global solution is well recovered. Fig. 2.11 (right) shows that, if N_{λ}^{δ} is large enough, the following error estimate for P2-P1 elements

$$\left(\sum_{i=1}^3 \|\mathbf{u} - \mathbf{u}^h\|_{H^1(\Omega_i)}^2 \right)^{1/2} + \left(\sum_{i=1}^3 \|p - p^h\|_{L^2(\Omega_i)}^2 \right)^{1/2} \leq Ch^2 \quad (2.21)$$

holds. In particular, we observe that when only low frequencies are considered, the error in the Lagrange multiplier $\boldsymbol{\lambda}$ is larger than the FE error and the global error remains approximately constant when h tends to zero. For $N_{\omega}^{\delta} = 4$ and $N_{\omega}^{\delta} = 5$, on the other hand, the error in $\boldsymbol{\lambda}$ is negligible and the rate of convergence of finite elements is fully recovered. We remark that, compared to the total number of dofs for velocity and pressure in the three subdomains—which ranges in the intervals 3'232 – 48'978 for Ω_1 , 1'463 – 39'080 for Ω_2 and 2'760 – 93'202 for Ω_3 —the number of basis functions of Λ^{δ} considered in Fig.2.11 (right) is negligible (we take at most $N_{\lambda}^{\delta} = 66$).

Fig. 2.12 shows the distribution of error $\|\mathbf{u} - \mathbf{u}^h\|_2$ in domain Ω for two nonconforming meshes, when different numbers of frequencies are considered. Note that, being that the color scale is logarithmic, we set the error in the Dirichlet boundaries (which is zero) to the arbitrary value 10^{-16} . In the first two rows, corresponding to the choice $N_{\omega}^{\delta} = 0$ (top

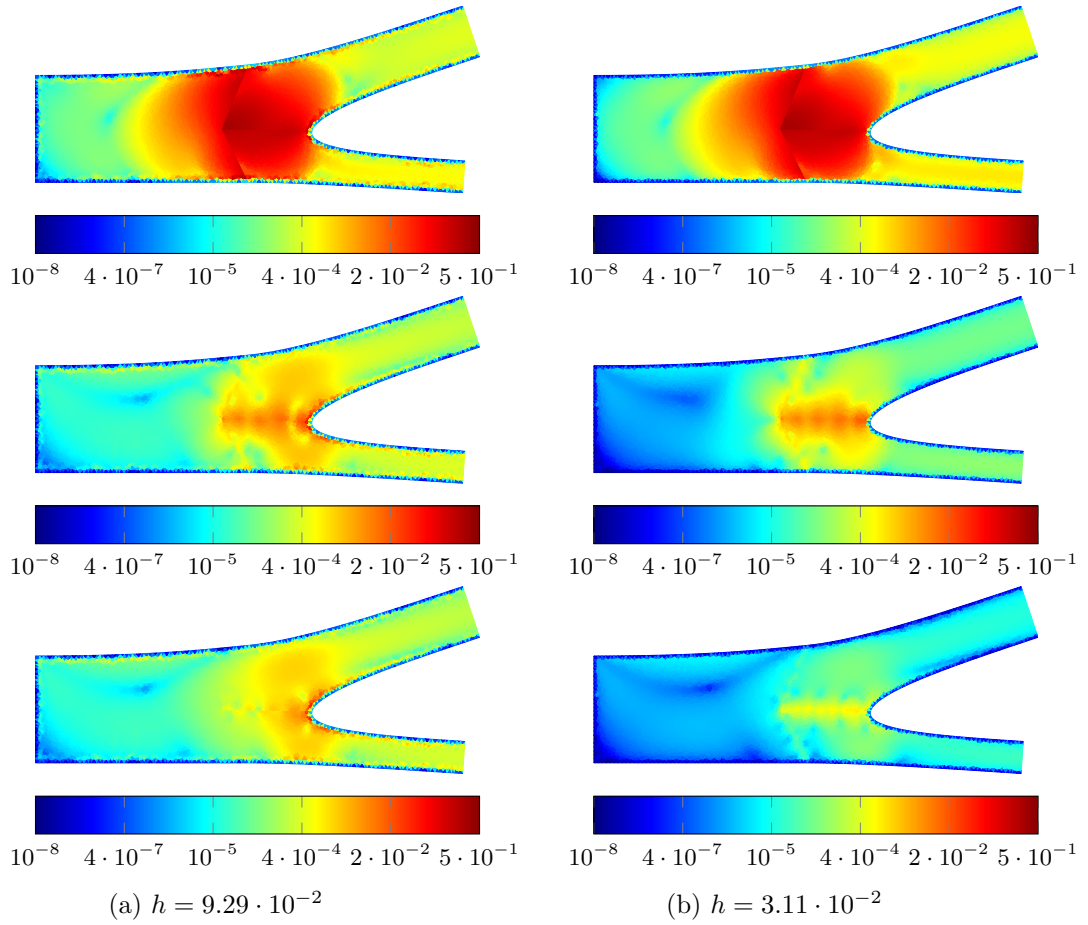


Figure 2.12 – Error distribution on the bifurcation when employing $N_{\omega}^{\delta} = 0$ (top row), $N_{\omega}^{\delta} = 2$ (middle row) and $N_{\omega}^{\delta} = 4$ (bottom row) and a coarse mesh (left column, $h = 9.29 \cdot 10^{-2}$) and a fine mesh (right column, $h = 3.11 \cdot 10^{-2}$).

row) and $N_\omega^\delta = 2$ (middle row), the error is mostly concentrated around the interfaces (in particular Γ_2), and the maximum values reached for the two different mesh sizes are similar; this effect is caused by the poor approximation of the Lagrange multiplier spaces. In the bottom row, instead, the maximum values of the error reached in the finer mesh are significantly smaller than those obtained in the coarser one. Moreover, in the coarse mesh, the error is not localized in the region of any interface (that is, the FE error dominates the error in the Lagrange multiplier reconstruction), whereas, in the finer mesh, the larger errors are still found around Γ_2 .

FE-IGA coupling

We now investigate the possibility of coupling different numerical schemes by spectral Lagrange multipliers. Specifically, we consider IGA for the discretization of Ω_1 and Ω_3 , and the FE method for the discretization of Ω_2 .

For the sake of conciseness, we do not focus on the details of IGA, and we limit ourselves to recall the general ideas of the method. The interested reader is referred to, e.g., [CHB09, HCB05] for a complete overview, and to [BBWW15] for an application of the mortar method to IGA.

IGA is built upon the isoparametric concept, which prescribes the use of the same basis functions for the geometry generation and the discretization of the functional spaces where the solution of the differential problem is sought in. The most common choices for such basis functions are B-Splines and NURBS (Non-Uniform Rational B-Splines), which are commonly employed in all modern CAD (Computer-Aided Design) software. In this work, we solely focus on B-Splines, which are sufficient to represent the bifurcation in Fig. 2.10. The principal advantages of the approach are: (i) the high global continuity of the basis functions, which often entails higher accuracy than standard FE basis functions with an equal polynomial degree (which are only C^0 -continuous across the edges of the mesh), and (ii) the exact correspondence of geometry and computational domain, that is, the method does not require any mesh generation and—consequently—no geometrical approximation of curved boundaries.

Univariate B-Splines basis functions of order p are generated by means of the Cox-De Boor recursion formula [DB72] from a knot vector $\Xi = [\xi_1, \dots, \xi_{n+p+1}]$; we remark that we decided to adopt the usual notation ξ_i commonly found in the literature to denote the individual knots, even though we previously associated such letter to the basis functions of Λ^δ . In this paper we consider open knot vectors, i.e., knot vectors such that $\xi_1 = \dots = \xi_{p+1}$ and $\xi_{n+1} = \dots = \xi_{n+p+1}$. The continuity of the basis functions across each internal knot ξ_i is determined by its multiplicity, i.e., the continuity of the basis functions across such point is C^{p-m_i} . Multivariate B-Splines are obtained by the tensor product of two sets of univariate B-Splines. In the following, we will denote $\{\hat{\varphi}_i\}_{i=1}^{N_{\text{bf}}}$ a set

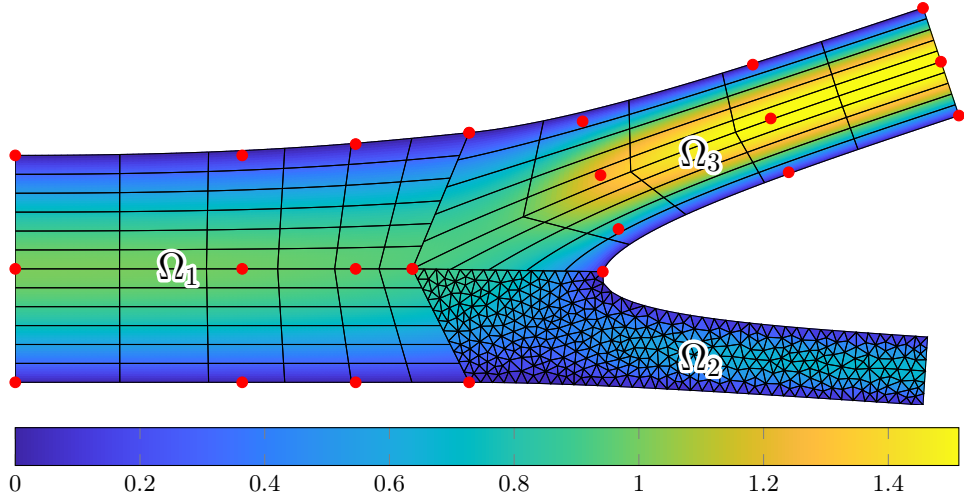


Figure 2.13 – Global solution obtained by considering 6×12 elements and P3-P2 B-Splines basis functions in Ω_1 , a FE mesh with size $h = 7.16 \cdot 10^{-2}$ and P2-P1 basis functions in Ω_2 , 5×10 elements and P3-P2 B-Splines basis functions in Ω_3 , and $N_\omega^\delta = 5$ frequencies for the discretization of Λ^δ . The control points are marked with red dots.

of bivariate basis functions; note that, as the polynomial degrees in the two parametric dimensions is a priori different, we omit to specify the degree of the bivariate polynomials. Moreover, we introduce the parametric domain $\widehat{\Omega} \in \mathbb{R}^2$, which is the support of the bivariate functions.

B-Splines surfaces are obtained from bivariate B-Splines by means of their linear combination with control points $\{\mathbf{b}_i\}_{i=1}^{N_{\text{bf}}}$, where each \mathbf{b}_i belongs either to \mathbb{R}^2 or to \mathbb{R}^3 if the surface is embedded in two or three dimensions respectively (here, we consider the former case). Therefore, the function

$$\mathbf{x}(\widehat{\mathbf{x}}) = \sum_{i=1}^{N_{\text{bf}}} \widehat{\varphi}_i(\widehat{\mathbf{x}}) \mathbf{b}_i : \widehat{\Omega} \rightarrow \Omega \quad (2.22)$$

defines a mapping—which we assume invertible for all $\widehat{\mathbf{x}} \in \widehat{\Omega}$ —from the parametric space $\widehat{\Omega}$ to the physical space Ω . By the isoparametric concept, we define a set of basis functions on Ω as $\varphi_i = \widehat{\varphi}_i \circ \mathbf{x}^{-1}$, for all $i = 1, \dots, N_{\text{bf}}$. We remark that the resulting functional space, even if intrinsically linked to the geometry definition, can be enriched by knot insertion (h -refinement), order elevation (p -refinement) and a combination of the two techniques (k -refinement) [CHR07]; such modifications of the knot vectors in the parametric dimensions do not affect the geometry of the physical domain.

Fig. 2.13 shows as an example the partitioned domain Ω with a FE triangulation in Ω_2 and a subdivision into elements of the physical domains Ω_1 and Ω_3 . These are obtained from mappings of the form (2.22), acting from the parametric domains $\widehat{\Omega}_1 = \widehat{\Omega}_3 =$

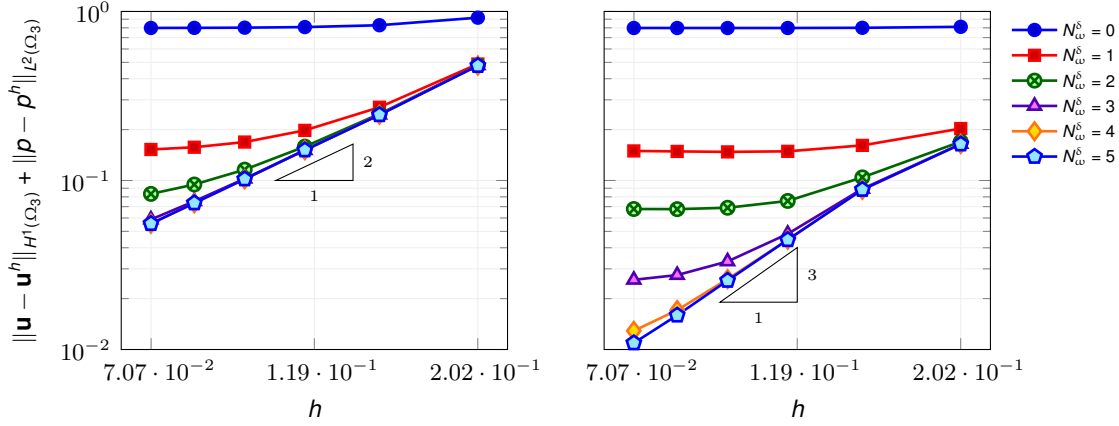


Figure 2.14 – Convergence of the error in Ω_3 with respect to the mesh size h and number of basis functions on the interface N_λ^δ . The basis functions in Ω_3 are P2-P1 B-Splines (left) and P3-P2 (right).

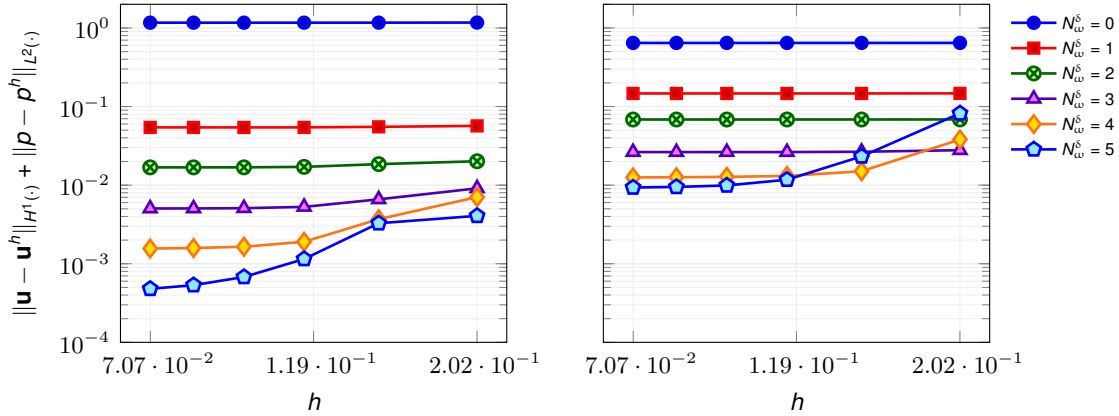


Figure 2.15 – Error in Ω_1 (left) and Ω_2 (right) with respect to the mesh size h in Ω_3 . In the latter subdomain, we employ P3-P2 B-Splines basis functions.

$[0, 0, 0, 0, 1, 1, 1, 1] \times [0, 0, 0.5, 1, 1]$. The resulting bivariate space is then composed of P3/C2 (piecewise cubic with continuity up to the second derivatives) basis functions in the first direction, and P1/C0 (piecewise linear with C^0 -continuity) basis functions. The corresponding control points are marked with red dots.

In order to approximate the two components of the velocity and the pressure appearing in the Navier–Stokes equations, we define functional spaces as presented above. In particular, in Ω_1 and Ω_2 we employ Taylor–Hood elements—which, in the context of IGA, have been analyzed, e.g., in [BDFS11]—of degree p and continuity $p - 1$ for the pressure and degree $p + 1$ and continuity p for the velocity, whereas in Ω_2 we only consider P2-P1 basis functions. A global solution obtained by using P3-P2 basis functions in Ω_1 , P2-P1 basis functions in Ω_2 and P2-P1 basis functions in Ω_3 is shown in Fig. 2.13.

2.7. A time-dependent fluid-structure interaction benchmark

In this simulation, we test the convergence of the error with respect to the mesh size of a single subdomain, namely Ω_3 . To this end, we fix the number of dofs in Ω_1 and Ω_2 and progressively refine the mesh in Ω_3 . In Ω_1 , we employ P3-P2 basis functions and 3'200 elements, for a total of 27'390 and 3'655 dofs for the velocity and the pressure respectively; in Ω_2 we consider P2-P1 FE basis functions on a mesh with 12'448 elements, which correspond to 50'590 and 12'848 dofs for the velocity and the pressure respectively. Fig. 2.14 shows the convergence of the error with respect to the mesh size in Ω_3 , for various levels of discretization of the space Λ^δ , when P2-P1 (on the left) and P3-P2 (on the right) Taylor–Hood basis functions are employed. The error is computed with respect to the fine FE solution obtained with P3-P2 basis functions we already considered in Section 2.6.3. The results are consistent with what was found in the previous sections, and in both cases, we are able to recover the optimal convergence of IGA (quadratic on the left, cubic on the right). Fig. 2.15 shows the behavior of the error in the first two subdomains when P3-P2 basis functions are employed in Ω_3 on a mesh with varying size. These results show that the accuracy of the stress reconstruction at the interfaces has a strong impact on the accuracy of the solution in Ω_1 and Ω_2 . Smaller meshes sizes in Ω_3 along with a good approximation of the Lagrange multipliers, indeed, lead to smaller errors in the two subdomains even if the number of dofs of the respective local solutions remains constant.

2.7 A time-dependent fluid-structure interaction benchmark

When deriving the fluid-structure interaction problem presented in Section 1.6, we assumed the fluid velocity-structure displacement continuity and the equilibrium of the stresses at the fluid-structure interface. These two conditions are equivalent to the ones that are imposed in the primal hybrid formulations of the problems presented in this chapter, with the sole difference being that in all the cases presented so far, the underlying physical model (i.e., the PDE) was the same in all subdomains. Therefore, it is natural to investigate the possibility of employing an approach based on spectral basis functions defined on the fluid-structure interface for the numerical approximation of the fluid-structure interaction problem.

Let us look at a slight modification of a classic fluid-structure interaction benchmark which has originally been proposed in [Nob01]. The same problem has been considered in [CDFQ11, DDFQ06, WC14, For16]. The geometry consists of a cylinder of length $L = 5$ cm and radius $R_{\text{in}} = 0.5$ cm; the curved portion of the surface is extruded in the perpendicular direction to reach the outer radius $R_{\text{out}} = 0.6$ cm. The internal cylinder constitutes the fluid domain $\widehat{\Omega}_f$, whereas the external layer represents the structure domain $\widehat{\Omega}_s$ (the described configuration is the reference one for both subdomains). A sketch of the problem geometry is depicted in Fig. 2.16. The fluid and structure properties are: $\rho_f = 1.0$ g/cm³, $\mu_f = 0.03$ g/(cm s), $\widehat{\rho}_s = 1.2$ g/cm³, $\nu_s = 0.3$ and $E_s = 3 \cdot 10^6$ dyne/cm².

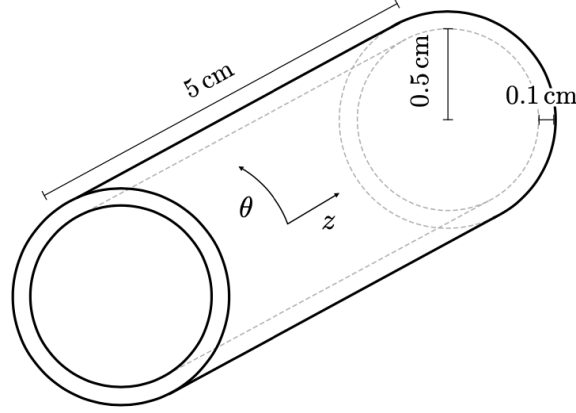


Figure 2.16 – Sketch of the fluid (inner cylinder) and structure (outer layer) domains.

As for the boundary conditions, we proceed as follows. We impose a normal stress

$$\mathbf{n}^T(\sigma_f \mathbf{n}) = -\frac{1}{2}(1 - \cos(\omega t)) P_{\max}, \quad \omega = \frac{2\pi}{\tilde{t}}, \quad P_{\max} = 1.33 \cdot 10^4 \text{ dyne/cm}^2,$$

for $t \leq \tilde{t} = 3 \cdot 10^{-3}$ s and $\mathbf{n}^T(\sigma_f \mathbf{n}) = 0$ for $t > \tilde{t}$ at the fluid inlet, whereas on the opposite face we set homogeneous Neumann boundary conditions. We remark that the boundary conditions are the only change we adopt with respect to the original benchmark (in which the stress at the inlet is constant and equal to P_{\max} for $t \leq \tilde{t}$). The structure is clamped (i.e., the displacement is set to zero) at the inlet and outlet; homogeneous Neumann boundary conditions are imposed on the curved outer boundary of the structure. We consider the time interval $(0 \text{ s}, T = 1.6 \cdot 10^{-2} \text{ s})$ and the timestep $\Delta t = 2 \cdot 10^{-4}$ s.

On the fluid-structure interface, we set the kinematic and dynamic conditions introduced in Section 1.6. These are enforced by means of spectral basis functions defined on the fluid-structure interface $\widehat{\Gamma}$ in the reference configuration. In order to provide the analytical representation of such functions, we introduce the system of cylindric coordinates shown in Fig. 2.16 and we denote by z and θ the variables in the longitudinal and angular directions, respectively. The basis functions for the Lagrange multiplier spaces are obtained as tensor product of two sets of basis functions taking values over the two cylindrical coordinates, i.e.,

$$\widehat{\xi}_{ij}^\delta(z, \theta) = \widehat{\xi}_{i,z}^\delta(z) \widehat{\xi}_{j,\theta}^\delta(\theta), \quad i = 1, \dots, N_{\lambda,z}^\delta, j = 1, \dots, N_{\lambda,\theta}^\delta.$$

Although the two sets of basis functions $\widehat{\xi}_{i,z}^\delta$ and $\widehat{\xi}_{i,\theta}^\delta$ may be chosen independently, we consider definition (2.17) for both. We recall that, due to (2.17), we have $N_{\lambda,z}^\delta = 2N_{\omega,z}^\delta + 1$ and $N_{\lambda,\theta}^\delta = 2N_{\omega,\theta}^\delta + 1$, where $N_{\omega,z}^\delta$ and $N_{\omega,\theta}^\delta$ are the considered frequencies in the two directions. Owing to the axial symmetry of the problem, moreover, we fix $N_{\omega,\theta}^\delta = 1$,

2.7. A time-dependent fluid-structure interaction benchmark

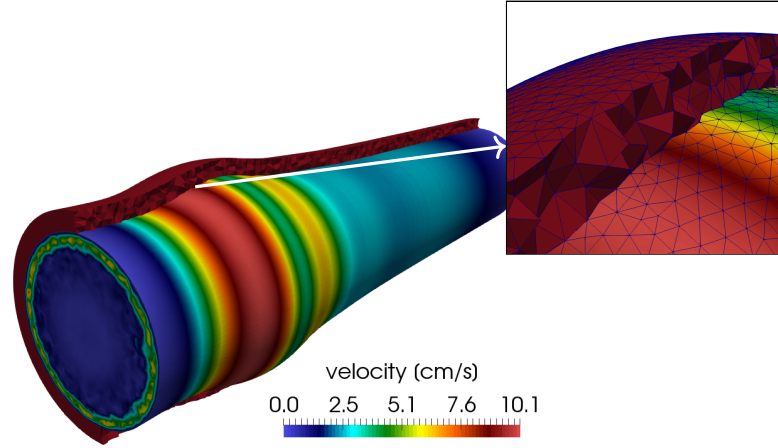


Figure 2.17 – Velocity solution of the fluid velocity on the deformed domain (displacement is magnified by 10 times) at $t = 4 \cdot 10^{-3}$ s and detail of the computational mesh. The global mesh is conforming.

namely we consider only the constant function and a single frequency for sin and cos in the angular direction. This is sufficient, because the problem is axisymmetric: therefore, the stress of the solution is constant over each circular section of the geometry and directed as the normal to the surface. We are interested in assessing the effects of the number of frequencies $N_{\omega,z}^\delta$ employed in the longitudinal direction on the convergence of the solution. As noted in Remark 2.6, when dealing with the Navier–Stokes equations it is necessary to discretize each component of the normal part of the stress. Being this three-dimensional, the total set of basis functions for the Lagrange multiplier space has therefore dimensionality

$$3N_\lambda^\delta = 3N_{\lambda,z}^\delta N_{\lambda,\theta}^\delta = 3 \left(2N_{\omega,z}^\delta + 1 \right) \left(2N_{\omega,\theta}^\delta + 1 \right) = 9 \left(2N_{\omega,z}^\delta + 1 \right).$$

We consider a conforming mesh and we discretize the Navier–Stokes equations by means of P1-P1 elements with VMS-SUPG stabilization, whereas the structure displacement is found as combination of piecewise linear polynomials. This approach leads to 326'850 dofs for the velocity, 108'950 dofs for the pressure, 244'146 dofs for the displacement and 326'850 dofs for the fluid displacement (which, we recall, is found as harmonic extension of the structure displacement inside the fluid domain). Choosing a conforming mesh allows us to consider a reference solution obtained by identifying the dofs of fluid velocity with those of the first derivative of the structure displacement at the interface $\widehat{\Gamma}$. From the theoretical standpoint, this is equivalent to discretize the Lagrange multiplier space with the P1 FE basis functions defined over the nodes of the fluid-structure interface $\widehat{\Gamma}$. Hence, the discrete Lagrange multiplier in the reference solution features 74'256 dofs, namely the number of mesh nodes on $\widehat{\Gamma}$.

Fig. 2.18 depicts the pressure distribution in the fluid domain at $t = 4 \times 10^{-3}$ s, $t =$

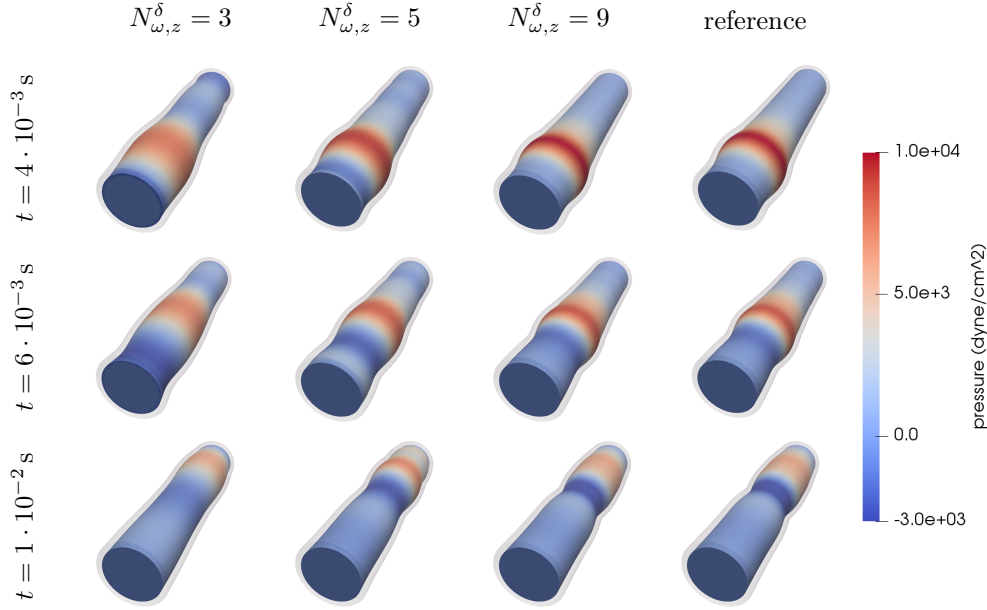


Figure 2.18 – Pressure wave in the fluid domain at three timesteps and for different discretization choices of the Lagrange multiplier space on the fluid-structure interface. The deformation of the domain is amplified by 10 times for demonstration purposes.

6×10^{-3} s and $t = 1 \times 10^{-2}$ s for different numbers of frequencies in the longitudinal direction $N_{\omega,z}^{\delta}$ and for the reference solution. The pressure wave along with the expansion of the fluid domain move from the inlet to the outlet regardless of the discretization of the Lagrange multiplier space. A similar analysis is offered in Fig.2.19, in which velocity magnitude and pressure are plotted as functions of the longitudinal coordinate. Results close to the reference ones are obtained for $N_{\omega,z}^{\delta} = 9$ but in the face of a much smaller number of total dofs (171) for the Lagrange multiplier.

Owing to the technical difficulties related to computing errors on Ω_f and Ω_s (which are deformable and therefore change in time), we simplify the analysis of the convergence of the solutions obtained with our method to the reference one by looking at the global L^2 norms of the quantities of interest. These are shown in function of time in Fig. 2.20. We observe that—as already pointed out in the discussion over the results presented in Fig. 2.19—a good agreement with the reference solution is obtained for $N_{\omega,z}^{\delta} = 9$, whereas considering a more limited number of frequencies in the longitudinal direction leads to considerably less accurate approximations. Moreover, we observe for particularly small sizes of the Lagrange multiplier space ($N_{\omega,z}^{\delta} = 1$) the discrete solution is unstable and the norms of velocity, pressure and structure displacement keep growing in time until the solver is unable to converge.

2.7. A time-dependent fluid-structure interaction benchmark

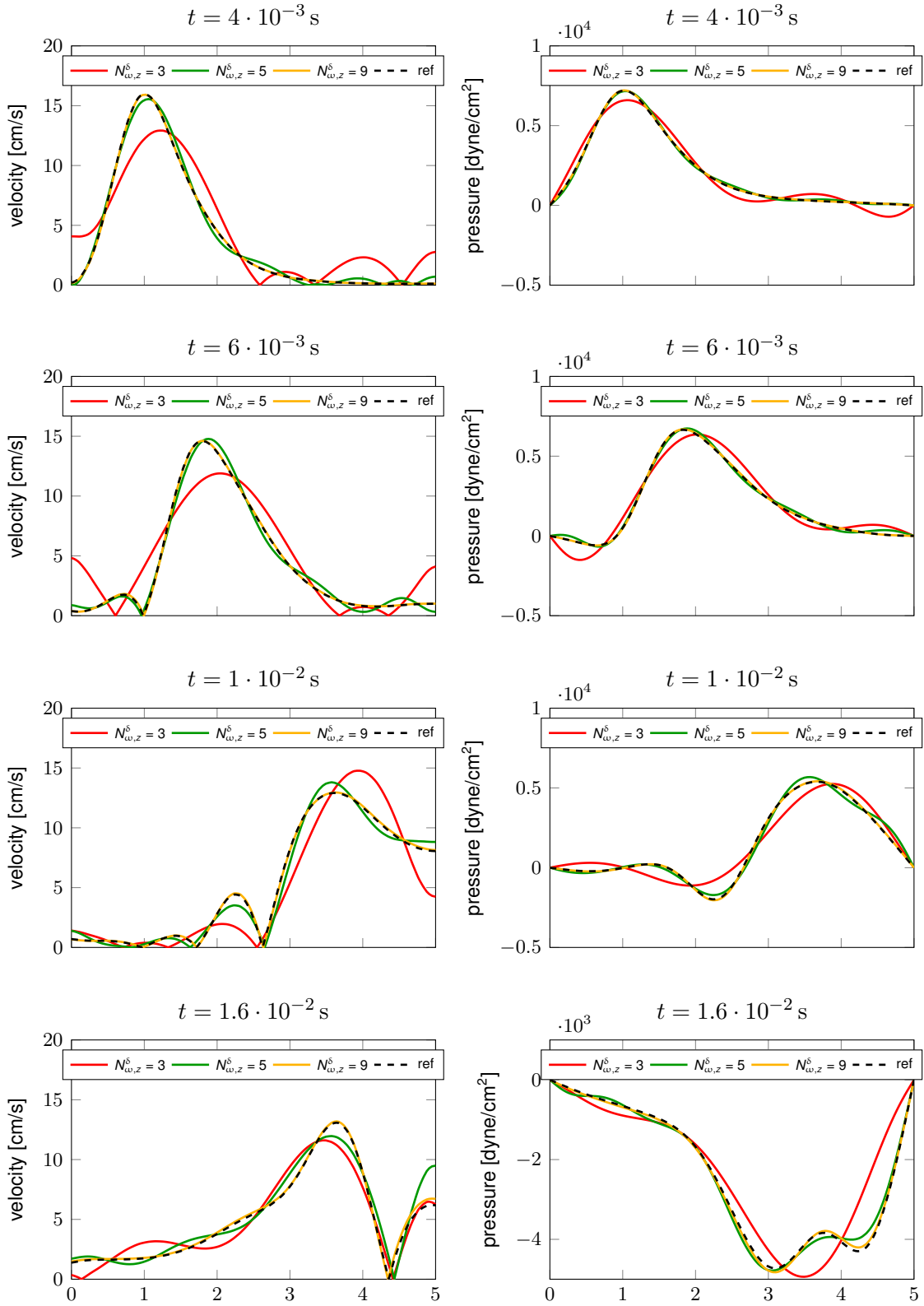


Figure 2.19 – Velocity magnitude and pressure along the centerline of the fluid domain.

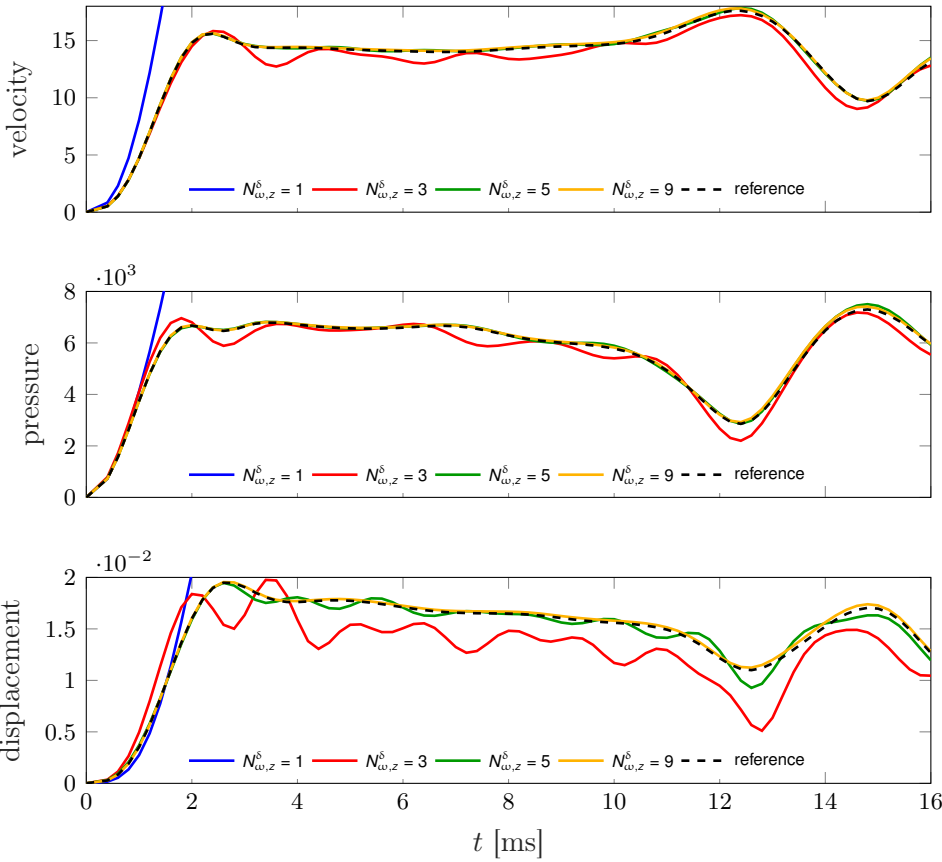


Figure 2.20 – L^2 norms of velocity, pressure and displacement for different degrees of sizes of the Lagrange multiplier space on the interface.

2.8 Concluding remarks

We presented a nonconforming method for the coupling of PDEs defined on domains partitioned into nonoverlapping subdomains. At the continuous level, our method and the mortar method are based on the same weak formulations in which the continuity constraints over the primal (the solution) and the dual (the stress) variables are enforced via Lagrange multipliers. As we described in this chapter, our choice of discretizing the space of Lagrange multipliers independently of the spatial discretization in the subdomains offers the advantage of a straightforward implementation of the method and the possibility of tuning the accuracy of the coupling as required by the application; we limited ourselves to considering Fourier basis functions defined over the interface. However, the saddle-point nature of the problem poses constraints over the richness of the discretized space for the Lagrange multipliers compared to the dofs of the primal variable: we empirically verified that the inf-sup constant can be controlled dependently on the mesh size, in the sense that finer meshes allow us to consider a larger number of Fourier basis functions, without violating the inf-sup stability. In the numerical experiments, we showed that the optimal convergence of the FE method was recovered for the Poisson problem; this was confirmed both when using conforming and nonconforming meshes, and when using different polynomial degrees in the subdomains. We also verified that the error obtained when employing a sufficiently rich basis for the Lagrange multiplier spaces is equal to that obtained considering a global conforming mesh with comparable size (see Figure 2.5). Since this represents a lower bound for the error of the mortar method as well, it can be said that—with an appropriate choice of basis functions for the Lagrange multiplier space—our method performs at least as well as the mortar method in terms of accuracy. We showed that the method can be easily extended to the case of nonelliptic equations, such as the Navier–Stokes equations, and to cases in which multiple partitions of the domain are discretized with different numerical methods, namely the FE method and IGA. In these numerical simulations were also able to retain optimal convergence rates by choosing an appropriate number of basis functions for the approximation of the Lagrange multiplier space. Lastly, we drew a parallel between the primal hybrid formulation for coupled problems and the fluid-structure interaction problem introduced in Chapter 1. By performing the coupling of fluid velocity and structure displacement by means of spectral basis functions, we showed that it is possible to obtain accurate results on a classical benchmark by considering a small space of Lagrange multipliers.

We believe that some aspects of our coupling strategy need further work. For example, the choice of basis functions at the interface N_λ^δ is, for the moment, based on a trial and error approach, and the optimal one could be guided by a priori estimates on the approximation error that still need to be developed. Furthermore, an efficient way to estimate the inf-sup stability constant without the need to solve any expensive eigenvalue problems would significantly increase the applicability of the method to realistic scenarios. A current limitation of the method is that, despite it being easy to implement when the

Chapter 2. Non-conforming coupling of PDEs

interface has an analytical description, it becomes more involved in case of arbitrary interfaces, as in fluid-structure problems defined in vascular vessels.

3 Elements of the Reduced Basis method

This chapter provides an introduction to the Reduced Basis (RB) method with a particular focus on the topics that are of importance to the subsequent chapters. In particular, in Section 3.1 we start by motivating the need for Model Order Reduction (MOR) and discussing the ideas at the root of the RB method. This is formalized in Section 3.2, where we introduce the concepts of offline and online phases; this section expands upon the concepts presented in the introduction to the RB method in [PPMD20]. In Section 3.3, we briefly touch on the challenges characterizing the application of the RB method to parametrized domains and, specifically, the need to retain the divergence-free property whenever required (e.g., when treating the velocity field in the Stokes or Navier–Stokes equations). Section 3.4 deals with the issues related to the solvability of the reduced system in the particular case of saddle-point problems. Finally, in Section 3.5 we consider a numerical example that makes use of all the concepts presented throughout the chapter and that is also strictly related to the MOR framework presented in Chapter 4.

3.1 Motivation

In previous chapters, we described how the numerical solution of PDEs by classical discretization methods entails transforming the weak formulation of the problem into a finite-dimensional one. In the FE method, for example, this is achieved by performing the Galerkin projection of the continuous formulation onto the space spanned by the FE basis functions defined over a triangulation of the domain. In this case, the convergence of the numerical solution to the exact one is mainly determined by the polynomial degree of the basis functions and the maximum mesh size across the mesh elements. Varying either of these two characteristics to improve accuracy is associated with an increase in the number of dofs and the condition number of the corresponding linear system. For this reason, accurate approximations are often computationally expensive. In some contexts, however, one might be interested in solving a particular PDE multiple times for a variety of values of some underlying geometrical or physical parameters, and the costs linked to

the solution of a single high-fidelity simulation might be too high. In these multi-query scenarios, therefore, surrogate models—less accurate but considerably more efficient than the high-fidelity ones—are an attractive choice. Such simplified models are often obtained through MOR techniques that aim at decreasing the number of dofs.

The RB method is a MOR strategy which deals with the particular case of parametrized PDEs. The main idea is the following. Given a specific parametrized PDE, it is reasonable to assume that two solutions obtained with a similar choice of parameters will be “close” to each other (in some suitable norm). As an example, let us consider the case in which we are interested in finding the temperature distribution in a room where the physical properties of the air (e.g., its diffusivity) are parameterized. Two slightly different air compositions are unlikely to give rise to drastically different evolutions of the temperature in the room, provided that the prescribed boundary conditions are the same in both scenarios. Given a specific choice of air parameters, the solutions for values of the parameters in its vicinity could be then obtained as a modification of the corresponding temperature distribution. Taking this idea further, one might also imagine that, even when two very different types of air are considered, the temperature distributions share many characteristics in common (for one, the boundary conditions need to be satisfied in both cases). These observations suggest that any admissible temperature distribution in the room might be effectively described, instead of as the set of temperature values in every point of the room, as the composition of a small number of characteristic *modes* encoding the essential features of the problem. As we shall see in the next section, these ideas are indeed the basis of the RB method.

3.2 The Reduced Basis method in a nutshell

In this section, we provide a non-comprehensive introduction to the RB method, which is only intended to set the theoretical basis for the remainder of the thesis. For a complete overview, we refer the reader to [QMN15, HRS16].

Let us consider an open and bounded domain Ω and a steady differential problem of the form

$$\mathcal{L}(u; \underline{\mu}) = \mathcal{G}(\underline{\mu}), \tag{3.1}$$

where $u \in \mathcal{V}$ (\mathcal{V} being a suitable functional space) is the solution, $\underline{\mu} \in \mathcal{D} \subset \mathbb{R}^{N_\mu}$ is a vector of geometrical and/or physical parameters, \mathcal{L} is a generic differential operator, and \mathcal{G} is a functional encoding the data of the problem, such as forcing term and boundary conditions. In this section we assume that \mathcal{L} is an elliptic operator.

As already mentioned, the standard approach to solve Eq. (3.1) by a Galerkin method corresponds to transforming the continuous problem into a finite dimensional one,

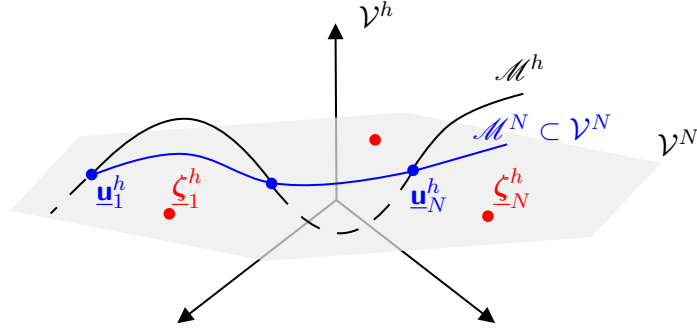


Figure 3.1 – Illustration of the RB method in the case $N_\mu = 1$. Image inspired by [QMN15].

often referred to as *Full Order Model* (FOM). With respect to the model problem in Eq. (3.1), this translates to $u^h = \sum_{i=1}^{N^h} u_i^h \varphi_i^h$, where $\varphi_i^h \in \mathcal{V}^h \subset \mathcal{V}$ are FE basis functions; $\underline{\mathbf{u}}^h = [u_1^h, \dots, u_{N^h}^h]^T$ is typically called vector of degrees of freedom. Assuming that the differential operator \mathcal{L} can be mathematically described in the weak sense by a bilinear form as $\ell(\varphi, \psi)$, for $\varphi \in \mathcal{V}$ and $\psi \in \mathcal{V}$, we identify the matrix $L^h(\underline{\boldsymbol{\mu}})_{ij} = \ell(\varphi_j^h, \varphi_i^h; \underline{\boldsymbol{\mu}}) \in \mathbb{R}^{N^h \times N^h}$, and similarly $\underline{\mathbf{G}}^h(\underline{\boldsymbol{\mu}})_i = \int_\Omega \mathcal{G}(\underline{\boldsymbol{\mu}}) \varphi_i^h \in \mathbb{R}^{N^h}$. For example, for the linear differential operator $\mathcal{L}(u; \eta) = -\eta \Delta u$, describing a Poisson equation with parameter η , we have $\ell(\varphi_j^h, \varphi_i^h; \eta) = \int_\Omega \eta \nabla \varphi_j^h \cdot \nabla \varphi_i^h$. The resulting linear system of dimension $N^h \times N^h$

$$L^h(\underline{\boldsymbol{\mu}}) \underline{\mathbf{u}}^h = \underline{\mathbf{G}}^h(\underline{\boldsymbol{\mu}}) \quad (3.2)$$

is possibly very large and expensive to solve. The main idea of the RB method is to construct a low dimensional basis for the solution u out of a number of solutions N_s (snapshots) of the FOM, which are computed during the so-called *offline phase*. In the *online phase*, the reduced solution is obtained as a linear combination of the RB functions; system (3.2) is casted in the form of a small linear system where the unknowns represent the coefficients of such a linear combination. In the remainder of this section, we address the offline and online phases more in depth.

3.2.1 The offline phase: basis construction

The goal of the offline phase is to construct a set of RB functions $\{\underline{\boldsymbol{\zeta}}_i^h\}_{i=1}^{N_s}$ suitable to approximate the (potentially nonlinear) manifold $\mathcal{M}^h = \{\underline{\mathbf{u}}^h(\underline{\boldsymbol{\mu}}) : \underline{\boldsymbol{\mu}} \in \mathcal{D}\}$ over which the solution $\underline{\mathbf{u}}^h(\underline{\boldsymbol{\mu}})$ resides. The RB functions depend on a set of snapshots $\underline{\mathbf{u}}_1^h, \dots, \underline{\mathbf{u}}_{N_s}^h$, which are obtained by solving the FOM for values of the parameters $\underline{\boldsymbol{\mu}}_1, \dots, \underline{\boldsymbol{\mu}}_{N_s}$ sampled in \mathcal{D} . The manifold \mathcal{M}^N in which the reduced solution is sought for in the online phase is included in the set of linear combinations of such basis functions, i.e., $\mathcal{M}^N \subset \mathcal{V}^N = \text{span}\{\underline{\boldsymbol{\zeta}}_1^h, \dots, \underline{\boldsymbol{\zeta}}_{N_s}^h\}$. For demonstration purposes, in Fig. 3.1 we show a case in which

Chapter 3. Elements of the Reduced Basis method

$N_\mu = 1$. In this scenario, the nonlinear manifold \mathcal{M}^h is described as a curve within \mathcal{V}^h which is not contained into a plane. The RB functions $\underline{\zeta}_i^h$ generate the hyperplane \mathcal{V}^N which contains the reduced manifold \mathcal{M}^N (blue line in the figure).

There exist two main strategies for the construction of the RB functions: greedy algorithms and the Proper Orthogonal Decomposition (POD) method.

Greedy algorithms

We briefly summarize the main concepts which set the basis for greedy algorithms in the context of the RB method. For a more complete overview, we refer to, e.g., [BCD⁺11, HSZ14].

The term greedy algorithm has been originally introduced in [Edm71]. Traditionally, by this name we denote iterative procedures in which we aim at minimizing a certain optimality criterion by enriching the solution set of an optimization problem with new elements (one in each iteration of the procedure). When dealing with the task of generating the set of RB functions, each step of a greedy algorithm adds a new basis function to the set such that a certain a posteriori error estimate is minimized. In particular, let us suppose that at iteration k we have constructed a basis $\{\underline{\zeta}_1^h, \dots, \underline{\zeta}_k^h\}$ for $\mathcal{V}^{(k)} \in \mathcal{V}^h$ from the orthonormalization of a set of snapshots $\underline{\mathbf{u}}^h(\underline{\mu}_1) = \underline{\mathbf{u}}_1^h, \dots, \underline{\mathbf{u}}^h(\underline{\mu}_k) = \underline{\mathbf{u}}_k^h$. We can also identify this basis with the matrix

$$V^{(k)} = [\underline{\zeta}_1^h | \dots | \underline{\zeta}_k^h].$$

Then, basis $V^{(k+1)}$ is computed by orthonormalization of $\{\underline{\zeta}_1^h, \dots, \underline{\zeta}_k^h, \underline{\mathbf{u}}^h(\underline{\mu}_{k+1})\}$. Parameter $\underline{\mu}_{k+1}$ is found through the solution of the optimization problem

$$\underline{\mu}_{k+1} = \arg \max_{\underline{\mu} \in \mathcal{P}} \|\underline{\mathbf{u}}^h(\underline{\mu}) - V^{(k)} \underline{\mathbf{u}}^N(\underline{\mu})\|_{X^h}, \quad (3.3)$$

where $X^h \in \mathbb{R}^{N^h \times N^h}$ is a suitable norm matrix and $\underline{\mathbf{u}}^N(\underline{\mu})$ is computed as explained in Section 3.2.2.

From the discussion so far, it is evident that greedy algorithms lead to an efficient offline phase, in the sense that they allow to minimize the number N_s of FE solutions (snapshots) to be computed (compared, as we shall see in the following section, to a POD approach, which is based on the assumption that $N_s \geq N$). A major drawback of greedy algorithms is that they are based on the a posteriori estimate of the projection error (3.3). This is problematic for two main reasons: (i) the maximization is performed over the whole set of parameters \mathcal{D} , and (ii) the evaluation of the error requires the computation of the high-fidelity solution $\underline{\mathbf{u}}^h(\underline{\mu})$. While the former shortcoming can be circumvented by sampling a large enough number of parameters in \mathcal{D} , this still requires

3.2. The Reduced Basis method in a nutshell

Algorithm 3.1 POD. Given the snapshots matrix S , the algorithm returns matrix V which is composed of orthonormal POD modes.

Input: Snapshot matrix $S = [\mathbf{u}_1^h | \dots | \mathbf{u}_{N_s}^h] \in \mathbb{R}^{N^h \times N_s}$, POD tolerance ε

Output: Orthonormal basis $V \in \mathbb{R}^{N^h \times N}$

- 1: **if** $N_s < N^h$ **then**
 - 2: Compute $C = S^T S$
 - 3: Solve the eigenvalue problem $C \underline{\psi}_i = \sigma_i^2 \underline{\psi}_i$ for $i = 1, \dots, N_s$
 - 4: Set $\underline{\zeta}_i^h = \frac{1}{\sigma_i} S \underline{\psi}_i$ for $i = 1, \dots, N_s$
 - 5: **else**
 - 6: Compute $K = S S^T$
 - 7: Solve the eigenvalue problem $K \underline{\zeta}_i^h = \sigma_i^2 \underline{\zeta}_i^h$ for $i = 1, \dots, N_s$
 - 8: Find \widehat{N} as the smallest integer such that Eq. (3.4) is satisfied
 - 9: Set $N = \widehat{N}$ and form the basis $V := [\underline{\zeta}_1^h | \dots | \underline{\zeta}_N^h]$
-

multiple evaluations of the FOM unless an inexpensive surrogate of the a posteriori error estimate is available. Therefore, greedy algorithms are sometimes too expensive and impractical, especially for realistic applications.

The Proper Orthogonal Decomposition method

In light of the limitations of greedy algorithms, in this thesis we opt for the POD method, which requires a larger number of snapshots N_s but is in turn more general. We refer, e.g., to [KV01, RP03] and [KV02] for applications of POD to parabolic and fluid problems, respectively, and to [RP03] for [KV01] for a comprehensive study of the properties of POD when applied to the solution of Ordinary Differential Equations (ODEs). In the context of cardiac simulations, this technique has been successfully employed both in fluid (e.g., in [BFI⁺16] to simulate blood flow in patient-specific coronary artery bypass grafts) and structural simulations (e.g., in [PCVL⁺20], where POD is used to reduce the space of admissible displacements of the heart muscle). In the POD approach, the reduced basis is usually constructed by singular value decomposition (SVD) [GVL12, TBI97] out of the set of snapshots, which are obtained by sampling N_s parameters $\underline{\mu}_1, \dots, \underline{\mu}_{N_s}$ in \mathcal{D} and by solving the corresponding FOM. Formally, we arrange the snapshots in matrix form as $S = [\mathbf{u}_1^h | \dots | \mathbf{u}_{N_s}^h] \in \mathbb{R}^{N^h \times N_s}$ and we seek matrices $U = [\underline{\zeta}_1^h | \dots | \underline{\zeta}_{N_s}^h] \in \mathbb{R}^{N^h \times N_s}$, $\Sigma \in \mathbb{R}^{N_s \times N_s}$ and $Z \in \mathbb{R}^{N_s \times N_s}$ such that $S = U \Sigma Z^T$; the columns of U and Z are orthonormal and Σ is diagonal. In the context of POD, $\underline{\zeta}_1^h, \dots, \underline{\zeta}_{N_s}^h$ are often called modes or POD modes. We remark that in classic SVD the matrices U and Σ are of size $N^h \times N^h$ and $N^h \times N_s$, respectively; here we consider the “economic” version of the algorithm. Matrix Σ takes the form $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_{N_s})$ and its diagonal is composed of the singular values of matrix S ordered from largest to smallest, i.e., $\sigma_1 \geq \dots \geq \sigma_{N_s} \geq 0$. Let us define $V := [\underline{\zeta}_1^h | \dots | \underline{\zeta}_N^h] \in \mathbb{R}^{N^h \times N}$ as the matrix composed of the first N modes

Chapter 3. Elements of the Reduced Basis method

and let us recall that, given a matrix $W = [\mathbf{w}_1 | \dots | \mathbf{w}_N] \in \mathbb{R}^{N^h \times N}$ whose columns are orthonormal, the projection of a generic vector $\mathbf{x} \in \mathbb{R}^{N^h}$ onto $\text{span}\{\mathbf{w}_1 | \dots | \mathbf{w}_N\}$ is given by $\Pi_W \mathbf{x} = WW^T \mathbf{x}$. Then, the following proposition holds.

Proposition 3.1. *Let $\mathcal{V}_N = \{W \in \mathbb{R}^{N^h \times N} : W^T W = I\}$, which can be interpreted as the set of all matrices composed of N -dimensional collections of orthonormal vectors in \mathbb{R}^h . Then,*

$$\sum_{i=1}^{N_s} \|\underline{\mathbf{u}}_i^h - VV^T \underline{\mathbf{u}}_i^h\|_2^2 = \min_{W \in \mathcal{V}_N} \sum_{i=1}^{N_s} \|\underline{\mathbf{u}}_i^h - WW^T \underline{\mathbf{u}}_i^h\|_2^2 = \sum_{i=N+1}^{N_s} \sigma_i^2.$$

We refer to [QMN15] for a proof of Proposition 3.1. In other words, V is the N -dimensional basis minimizing the projection error of the snapshots over its column space; moreover, such error is strictly related to the magnitude of the singular values $\sigma_{N+1}, \dots, \sigma_{N_s}$. Thus, a common heuristic to choose N is to set it equal to the smallest integer \tilde{N} such that

$$\frac{\sum_{i=1}^{\tilde{N}} \sigma_i^2}{\sum_{i=1}^{N_s} \sigma_i^2} \geq 1 - \varepsilon^2, \quad (3.4)$$

where ε is a user-provided tolerance. The left-hand side of Eq. (3.4) is the relative information content of the POD basis, namely the percentage of energy of the snapshots retained by the first \tilde{N} modes. The size of the reduced basis N selected by following criterion (3.4) is typically much smaller than the size of the FOM N^h , i.e., $N \ll N^h$. A strategy for POD based on the correlation matrices $C = S^T S \in \mathbb{R}^{N \times N}$ and $K = SS^T \in \mathbb{R}^{N^h \times N^h}$ is reported in Algorithm 3.1.

Remark 3.1. Given a symmetric positive definite matrix X^h which is a norm matrix for $\|\cdot\|_{\mathcal{V}}$ in the FE space, i.e., $\|u\|_{\mathcal{V}} = (\underline{\mathbf{u}}^h)^T X^h \underline{\mathbf{u}}^h$, it is possible to perform the POD such that the basis U is orthonormal with respect to X^h (i.e., $U^T X^h U = I$). In order to achieve this, we observe that, since X^h is symmetric positive definite, it admits a Cholesky decomposition $X^h = H^T H$, H being upper triangular. Matrix U is then found as $U = H^{-1} \tilde{U}$, where \tilde{U} is computed by SVD of $HS = \tilde{U} \tilde{\Sigma} \tilde{Z}^T$. When constructing the reduced basis for fluid simulations, following this approach allows us to achieve the optimality expressed in Proposition 3.1 with respect to norms more suited to the specific variables of interest (namely, H^1 norm for the velocity and L^2 norm for the pressure).

Remark 3.2. The POD approach presented in Algorithm 3.1, which is based on the correlation matrices C and K , suffers from the defect that it requires multiplying the snapshot matrix by itself. From the numerical point of view, this can lead to round-off errors affecting the accuracy of the smallest singular values. As an example, in Fig. 3.2 we study the singular values of matrix $A = U^T \text{diag}(\exp(-[1, \dots, M]/2))V \in \mathbb{R}^{N \times M}$, where $U \in \mathbb{R}^{M \times N}$ and $V \in \mathbb{R}^{M \times M}$ are random unitary matrices, when $N = 1000$ and $M = 41$. The comparison between the singular values σ_i computed with the `svd` function in MATLAB, the ones deriving from the correlation matrix approach and the exact ones highlights that loss of accuracy does indeed occur for small σ_i . From the perspective of

3.2. The Reduced Basis method in a nutshell

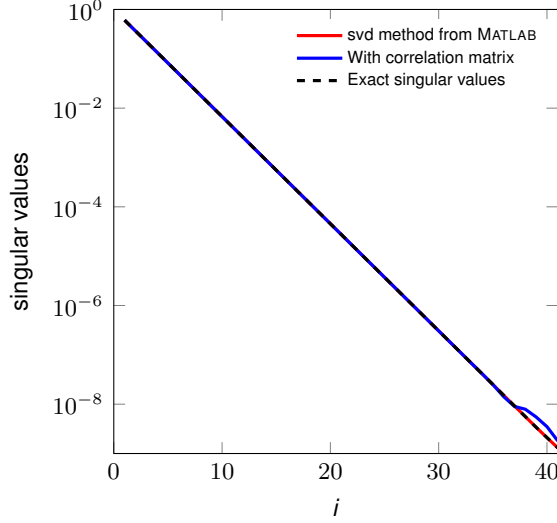


Figure 3.2 – Comparison of singular values computed with the `svd` method in MATLAB and with the correlation matrix approach presented in Algorithm 3.1.

Algorithm 3.1, this might be problematic because the resulting basis is not orthonormal. A better approximation of the smallest singular values can be obtained, for example, by resorting to the thin singular value decomposition [GVL12].

3.2.2 The online phase: solution of the reduced problem

Let us observe that it is legitimate to associate with each POD mode a corresponding functional representation $\zeta_j^h = \sum_{i=1}^{N^h} (\zeta_j^h)_i \varphi_i^h$. The RB approximation consequently reads $u^N = \sum_{i=1}^N u_i^N \zeta_i^h$, $\underline{\mathbf{u}}^N = [u_1^N, \dots, u_N^N]^T$ being the vector of reduced dofs. Then, evaluating the weak formulation of problem (3.1) at test and trial functions in $\text{span}\{\zeta_i^h\}_{i=1}^N$, we find the reduced linear system

$$L^N(\underline{\boldsymbol{\mu}})\underline{\mathbf{u}}^N = \underline{\mathbf{G}}^N(\underline{\boldsymbol{\mu}}), \quad (3.5)$$

where $L^N(\underline{\boldsymbol{\mu}})_{ij} = \ell(\zeta_j^h, \zeta_i^h) \in \mathbb{R}^{N \times N}$ and $\underline{\mathbf{G}}^N(\underline{\boldsymbol{\mu}})_i = \int_{\Omega} \mathcal{G}(\underline{\boldsymbol{\mu}}) \zeta_i^h \in \mathbb{R}^N$. The assembly and solution of system (3.5) correspond to the online phase. The transformation of the reduced vector of dofs into its FE counterpart is simply performed by $\underline{\mathbf{u}}^h \approx V \underline{\mathbf{u}}^N \in \mathbb{R}^{N^h}$.

By exploiting the expansion $\zeta_j^h = \sum_{i=1}^{N^h} (\zeta_j^h)_i \varphi_i^h$ it is easy to find that $L^N(\underline{\boldsymbol{\mu}}) = V^T L^h(\underline{\boldsymbol{\mu}}) V$ and $\underline{\mathbf{G}}^N(\underline{\boldsymbol{\mu}}) = V^T \underline{\mathbf{G}}^h(\underline{\boldsymbol{\mu}})$. Therefore, the assembly of the reduced system can be done by constructing the full order matrix and right-hand side and by computing their projection onto the RB space. If the problem features an affine decomposition, namely there exist

parameter-dependent coefficients θ_q^L for $q = 1, \dots, Q_L$ and θ_q^G for $q = 1, \dots, Q_G$ such that

$$L^h(\underline{\boldsymbol{\mu}}) = \sum_{q=1}^{Q_L} \theta_q^L(\underline{\boldsymbol{\mu}}) L_q^h, \quad \underline{\mathbf{G}}^h(\underline{\boldsymbol{\mu}}) = \sum_{q=1}^{Q_G} \theta_q^G(\underline{\boldsymbol{\mu}}) \underline{\mathbf{G}}_q^h, \quad (3.6)$$

a considerable speedup is achieved by precomputing the matrices $L_q^N = V^T L_q^h V$ and the vectors $\underline{\mathbf{G}}_q^N = V^T \underline{\mathbf{G}}_q^h$ in the offline phase, and by assembling the reduced elements of system (3.5) as

$$L^N(\underline{\boldsymbol{\mu}}) = \sum_{q=1}^{Q_L} \theta_q^L(\underline{\boldsymbol{\mu}}) L_q^N, \quad \underline{\mathbf{G}}^N(\underline{\boldsymbol{\mu}}) = \sum_{q=1}^{Q_G} \theta_q^G(\underline{\boldsymbol{\mu}}) \underline{\mathbf{G}}_q^N.$$

3.2.3 On the efficiency of the online phase: the Discrete Empirical Interpolation Method

Unfortunately, in most practical scenarios an affine decomposition of the form (3.6) is not readily available. In such cases, a common strategy to efficiently perform the assembly of system (3.5) employs the (Discrete) Empirical Interpolation Method (DEIM) [BMNP04, CS10] and its matrix variant MDEIM [NMA15]. The former is a procedure to approximate a general parametric multidimensional function $\mathbf{f}(\underline{\boldsymbol{\mu}})$ which we assume for the time being to be defined in \mathcal{D} and to take values in \mathbb{R}^h . Specifically, given several realizations of the function $\mathbf{f}_1, \dots, \mathbf{f}_M$ for different values of the parameter $\underline{\boldsymbol{\mu}}_1, \dots, \underline{\boldsymbol{\mu}}_M$, a suitable basis for its approximation, represented in matrix form as $\Psi = [\underline{\boldsymbol{\psi}}_1 | \dots | \underline{\boldsymbol{\psi}}_{N_f}]$, is generated by POD. Given a new parameter value $\underline{\tilde{\boldsymbol{\mu}}}$, the corresponding function evaluation is then found as

$$\underline{\mathbf{f}}(\underline{\tilde{\boldsymbol{\mu}}}) \approx \sum_{j=1}^{N_f} \theta_j^f(\underline{\tilde{\boldsymbol{\mu}}}) \underline{\boldsymbol{\psi}}_j = \Psi \underline{\boldsymbol{\theta}}^f(\underline{\tilde{\boldsymbol{\mu}}}).$$

The coefficients of the expansion onto the basis $\underline{\boldsymbol{\theta}}^f(\underline{\tilde{\boldsymbol{\mu}}}) = [\theta_1^f(\underline{\tilde{\boldsymbol{\mu}}}), \dots, \theta_{N_f}^f(\underline{\tilde{\boldsymbol{\mu}}})]^T \in \mathbb{R}^{N_f}$ are found by solving an interpolation problem in a selected number of components of $\underline{\mathbf{f}}(\underline{\tilde{\boldsymbol{\mu}}})$. In particular, the interpolation problem reads $\underline{\mathbf{f}}_{\mathcal{I}}(\underline{\tilde{\boldsymbol{\mu}}}) = \Psi_{\mathcal{I}} \underline{\boldsymbol{\theta}}^f(\underline{\tilde{\boldsymbol{\mu}}})$, where subscript \mathcal{I} denotes the restriction of the rows to a predefined set of indices, and the evaluation of $\underline{\mathbf{f}}$ at $\underline{\tilde{\boldsymbol{\mu}}}$ is approximated as $\underline{\mathbf{f}}(\underline{\tilde{\boldsymbol{\mu}}}) \approx \Psi \Psi_{\mathcal{I}}^{-1} \underline{\mathbf{f}}_{\mathcal{I}}(\underline{\tilde{\boldsymbol{\mu}}})$. The choice of interpolation indices \mathcal{I} clearly plays a crucial role in the accuracy of the approximation. In DEIM, these indices are chosen iteratively. Given the indices at iteration k , which we denote $\mathcal{I}^{(k)}$, the new index added to the set is the one corresponding to the largest component in the residual vector

$$\mathbf{r}_k = \underline{\boldsymbol{\psi}}_k - \Psi^{(k)} \left(\Psi_{\mathcal{I}^{(k)}}^{(k)} \right)^{-1} \left(\underline{\boldsymbol{\psi}}_k \right)_{\mathcal{I}^{(k)}},$$

3.2. The Reduced Basis method in a nutshell

Algorithm 3.2 DEIM. The argmax operator applied to a vector returns the index of its largest component.

- Input:** Function evaluations $F = [\mathbf{f}_1 | \dots | \mathbf{f}_M] \in \mathbb{R}^{N^h \times M}$, POD tolerance ε
Output: Orthonormal basis $\Psi \in \mathbb{R}^{N^h \times N_f}$
- 1: Compute $\Psi = [\underline{\psi}_1 | \dots | \underline{\psi}_{N_f}]$ from POD of F with tolerance ε
 - 2: Set $i = \operatorname{argmax} |\underline{\psi}_1|$
 - 3: $\mathcal{I} = \{i\}$
 - 4: **for** $k = 2 \dots N_f$ **do**
 - 5: Compute the residual $\mathbf{r}_k = \underline{\psi}_k - \Psi^{(k)} \left(\Psi_{\mathcal{I}^{(k)}}^{(k)} \right)^{-1} (\underline{\psi}_k)_{\mathcal{I}^{(k)}}$
 - 6: Set $i = \operatorname{argmax} |\mathbf{r}_k|$
 - 7: $\mathcal{I} = \mathcal{I} \cup i$
-

where $\Psi^{(k)}$ is the basis matrix restricted to the first k columns. The DEIM procedure is summarized in Algorithm 3.2.

The adaptation of DEIM to matrices (MDEIM) deals with the approximation of functions $F(\underline{\boldsymbol{\mu}})$, which we again assume for simplicity to be defined in \mathcal{D} and to take values in $\mathbb{R}^{N^h \times N^h}$. From the practical point of view, the only difference between DEIM and MDEIM is the fact that, in the latter, matrices need to be vectorized before processing. This obviously makes the process resource consuming, as the basis for $F(\underline{\boldsymbol{\mu}})$ is of dimension $(N^h)^2$.

It is worthwhile to discuss what DEIM and MDEIM entail from the practical point of view in the context of the approximation of structures requiring the assembly of elements onto FE meshes. Let us assume that matrix $L^h(\underline{\boldsymbol{\mu}})$ and vector $\mathbf{G}^h(\underline{\boldsymbol{\mu}})$ appearing in Eq. (3.2) do not admit an affine decomposition of the form (3.6). By resorting to (M)DEIM, we are able to recover the approximated affine decompositions

$$L^h(\underline{\boldsymbol{\mu}}) \approx \sum_{q=1}^{M_L} \Theta_q^L(\underline{\boldsymbol{\mu}}) L_q^h, \quad \mathbf{G}^h(\underline{\boldsymbol{\mu}}) \approx \sum_{q=1}^{M_G} \Theta_q^G(\underline{\boldsymbol{\mu}}) \mathbf{G}_q^h, \quad (3.7)$$

In this case, L_q^h are defined as $L_q^h = \operatorname{mat}(\underline{\psi}_q^L)$ (i.e., the matrix version of the q^{th} MDEIM basis function for L^h) and $\mathbf{G}_q^h = \underline{\psi}_q^G$. Given a new value of parameter $\underline{\boldsymbol{\mu}}$, coefficients $\Theta_q^L(\underline{\boldsymbol{\mu}})$ and $\Theta_q^G(\underline{\boldsymbol{\mu}})$ are the components with index q of vectors

$$\underline{\Theta}^L(\underline{\boldsymbol{\mu}}) = \Psi_{\mathcal{I}^L}^L \operatorname{vec} \left(L^h(\underline{\boldsymbol{\mu}}) \right)_{\mathcal{I}^L}, \quad \underline{\Theta}^G(\underline{\boldsymbol{\mu}}) = \Psi_{\mathcal{I}^G}^G \left(\mathbf{G}^h(\underline{\boldsymbol{\mu}}) \right)_{\mathcal{I}^G}.$$

Here, $\Psi^L = [\underline{\psi}_1^L | \dots | \underline{\psi}_{M^L}^L] \in \mathbb{R}^{(N^h)^2 \times M^L}$ and $\Psi^G = [\underline{\psi}_1^G | \dots | \underline{\psi}_{M^G}^G] \in \mathbb{R}^{N^h \times M^G}$. In order to find $\underline{\Theta}^L(\underline{\boldsymbol{\mu}})$ and $\underline{\Theta}^G(\underline{\boldsymbol{\mu}})$, it is required to evaluate $L^h(\underline{\boldsymbol{\mu}})$ and $\mathbf{G}^h(\underline{\boldsymbol{\mu}})$ at $\dim(\mathcal{I}^L)$ and $\dim(\mathcal{I}^G)$ components, respectively. Every entry of a FE vector or matrix, however, corresponds to the evaluation of an integral defined over the support of the FE basis

functions. The definition of a *reduced mesh*—i.e., a mesh comprising the set of elements which are required for the application of the algorithms—is a necessary step to ensure efficiency.

We remark that the need to assemble the FE structures at certain locations is a characteristic that makes the use of affine composition (3.7) considerably more cumbersome than that in Eq. (3.6). As a matter of fact, the evaluation of $\theta_q^L(\underline{\boldsymbol{\mu}})$ and $\theta_q^G(\underline{\boldsymbol{\mu}})$ does not require, in general, the knowledge of the mesh elements and connectivity, which implies that these do not need to be loaded in memory at the beginning of the RB simulation.

3.3 The Reduced Basis method in parametrized domains

In some applications, it is useful to consider domains in which some geometrical features are parametrized. The typical approach obtains the physical domain as a parametrized geometrical deformation of a predetermined reference domain. For example, in [LR10] the authors propose an algorithm based on free-form deformation for the shape optimization of airfoils, and in [MQR12] the same technique is applied to a two-dimensional bypass. In [HU18], the authors consider geometrical parametrizations based on reference configurations to study the properties of a recently developed strategy to MOR combining the RB method with Neural Networks. An approach that does not involve the definition of a reference domain and that is rooted in the RB method combined with the shifted boundary method [MS18] has been proposed in [KSN⁺19]. We refer to [DSDMQ19] for another example of the use of the RB method in parametrized domains.

From a practical point of view, deformable domains are challenging for many reasons. Here, we focus on two in particular, which are relevant for the applications presented in this thesis: (i) the lack of an affine-decomposition in most practical cases, and (ii) the need to treat with particular care divergence-free fields (which are a feature of the solutions to Stokes and Navier–Stokes equations, for instance).

Regarding the first point, we note that an affine-decomposition is available only in particular and simple cases (as an example, we refer the reader to, e.g., the backward-facing step channel presented in [QMN15, Neg15]). In general applications, (M)DEIM represents a viable strategy to approximate such affine-decompositions. Depending on the range of admissible geometrical parameters and on the specifics of the problem at hand, however, the computational burden associated with this approach may be considerably high. Moreover, the number of affine terms tends to increase with the complexity of the geometrical parametrization rapidly.

The treatment of divergence-free fields can be performed with the Piola transformation, which is the topic of the next section.

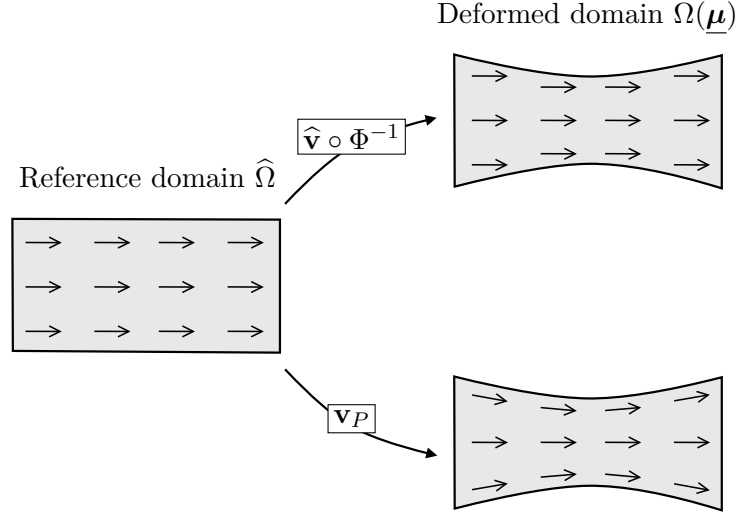


Figure 3.3 – Effect of the Piola transformation. The vector field defined on the reference domain $\hat{\Omega}$ is mapped without scaling on top and with scaling by the Piola transformation on the bottom.

3.3.1 The Piola transformation

Let us consider a reference domain $\hat{\Omega}$ and a vector field $\hat{\mathbf{v}} \in H(\text{div}; \hat{\Omega})$ such that $\nabla_{\hat{\mathbf{x}}} \cdot \hat{\mathbf{v}} = 0$. We introduce a transformation Φ which, for the purposes of this chapter, is parametrized with respect to $\underline{\mu} \in \mathcal{D}$, and which maps $\hat{\Omega}$ into the physical configuration $\Omega = \Phi(\hat{\Omega}; \underline{\mu})$. In general, the mapped vector field $\mathbf{v} = \hat{\mathbf{v}} \circ \Phi^{-1}$ does not preserve the divergence-free property in the reference domain, i.e., $\nabla_{\mathbf{x}} \cdot \mathbf{v} \neq 0$. Moreover, when applied to fluids in pipes—which is clearly a common scenario in cardiovascular simulation—such trivial mapping leads to loss of physically relevant features. For instance, in Fig. 3.3, the streamlines of the vector field on the top right stay parallel to the ones in the reference domain and therefore do not account for the narrowing of the pipe.

To overcome these issues, let us consider the following definition, cf. [QMN15].

Definition 3.1. Let $\Phi : \hat{\Omega} \mapsto \Omega$ and $(J_{\Phi}(\hat{\mathbf{x}}))_{ij} = \partial \Phi_i / \partial \hat{x}_j$ be its Jacobian matrix. Then, given a field $\hat{\mathbf{v}} \in L^2(\hat{\Omega})$, the Piola transformation is defined as

$$\mathbf{v}_P(\mathbf{x}) = \frac{1}{|J_{\Phi}(\hat{\mathbf{x}})|} J_{\Phi}(\hat{\mathbf{x}}) \hat{\mathbf{v}}(\hat{\mathbf{x}}). \quad (3.8)$$

The following theorem, which can be found, e.g., in [BBF13], motivates the use of the Piola transformation in the context of the preservation of the divergence-free property.

Theorem 3.1. *Let \mathbf{v}_P be defined as in Eq. (3.8). Then, there holds*

$$J_{v_P}(\mathbf{x}) = \frac{1}{|J_\Phi(\hat{\mathbf{x}})|} J_\Phi(\hat{\mathbf{x}}) J_{\hat{\mathbf{v}}}(\hat{\mathbf{x}}) (J_\Phi(\hat{\mathbf{x}}))^{-1}, \quad (3.9)$$

where J_{v_P} and $J_{\hat{\mathbf{v}}}$ are the Jacobians of \mathbf{v}_P and $\hat{\mathbf{v}}$ respectively, and

$$\nabla_{\mathbf{x}} \cdot \mathbf{v}_P(\mathbf{x}) = \frac{1}{|J_\Phi(\hat{\mathbf{x}})|} \nabla_{\hat{\mathbf{x}}} \cdot \hat{\mathbf{v}}(\hat{\mathbf{x}}). \quad (3.10)$$

Proof. Eq.(3.9) can be proven by the chain rule, see [BBF13]. Eq. (3.10) immediately follows by considering the trace of both sides of Eq. (3.9) and by remembering that the trace of matrices is invariant with respect to a change of variables. \square

The reader is referred, e.g., to [BBF13, TR77, Cia88] for other properties of the Piola transformation.

Eq. (3.10) demonstrates that, upon treatment by the Piola transformation, it is possible to preserve the divergence-free property in the deformed domain. Additionally, the transformed vector field $\mathbf{v}_P(\mathbf{x})$ preserves the flow features which are otherwise lost. In Fig. 3.3 (bottom right), we show how the flow direction is affected when applying Eq. (3.8). The qualitative behavior of the flow is, in this case, what is reasonable to expect when the fluid encounters a narrowing of the pipe.

3.4 Supremizers enrichment for saddle-point problems

The numerical approximation of saddle-point problems is a delicate matter. As discussed in Section 1.4, the inf-sup condition is peculiar to these family of problems and, intuitively, it requires that the space linked to the Lagrange multiplier—e.g., the pressure in the Navier–Stokes equations—is sufficiently small compared to the one in which the primal variable resides. In the case of the FE method, this amounts to choose the discretization spaces appropriately: for instance, Taylor–Hood elements [HT74] satisfy the inf-sup condition in the (Navier)–Stokes equations. Unfortunately, even if the high-fidelity problem is well-posed, the reduced one is not guaranteed to satisfy the necessary stability conditions. Among the ways to deal with the loss of stability in the reduced system are the use of least squares Petrov–Galerkin approaches for the solution of the minimization problem associated with the nonlinear residual of the reduced equations [CBA17, DSDMQ19] and the supremizers enrichment [Roz05, BMQR15, DSM18]. In this section, we present the latter approach.

Let us consider the generic form of a discretized saddle-point problem (1.17). We recall that, in the discrete setting, one requirement for its well-posedness is the existence of

3.4. Supremizers enrichment for saddle-point problems

$\beta^h > 0$ such that

$$\inf_{q^h \in \mathcal{Q}^h \setminus \{0\}} \sup_{v^h \in \mathcal{V}^h \setminus \{0\}} \frac{b(v^h, q^h)}{\|v^h\|_{\mathcal{V}^h} \|q^h\|_{\mathcal{Q}^h}} \geq \beta^h. \quad (3.11)$$

We now introduce the reduced version of Eq. (1.17). By following a Galerkin approach, we obtain: find $u^N \in \mathcal{V}^N$ and $p^N \in \mathcal{Q}^N$ such that

$$\begin{aligned} a(u^N, v^N) + b(v^N, p^N) &= \langle f^N, v^N \rangle & \forall v^N \in \mathcal{V}^N, \\ b(u^N, q^N) &= \langle g^N, q^N \rangle & \forall q^N \in \mathcal{Q}^N, \end{aligned}$$

and the corresponding inf-sup condition is based on the existence of $\beta^N > 0$ such that

$$\inf_{q^N \in \mathcal{Q}^N \setminus \{0\}} \sup_{v^N \in \mathcal{V}^N \setminus \{0\}} \frac{b(v^N, q^N)}{\|v^N\|_{\mathcal{V}^N} \|q^N\|_{\mathcal{Q}^N}} \geq \beta^N.$$

As already mentioned, the existence of β^h is not sufficient to that of β^N , because the supremum is taken over a smaller set.

One key observation to the supremizers enrichment approach applied to the RB method is that the latter is rooted in the premise that the RB spaces \mathcal{V}^N and \mathcal{Q}^N are *small*. Therefore, given that there exists only a limited number of basis functions for \mathcal{Q}^N , it is reasonably inexpensive to add for each of these functions the corresponding function in \mathcal{V}^N (the supremizer) which realizes the supremum in Eq. (3.11).

Formally, let $\eta_1^h, \dots, \eta_{N_p}^h$ be the basis functions for \mathcal{Q}^N , i.e., $\mathcal{Q}^N = \text{span}\{\eta_i^h\}_{i=1}^{N_p}$. Our goal is to find $s_i^h \in \mathcal{V}^h$ for $i = 1, \dots, N_p$ such that

$$s_i^h = \arg \sup_{v^N \in \mathcal{V}^N \setminus \{0\}} \frac{b(v^N, \eta_i^h)}{\|v^N\|_{\mathcal{V}^N} \|\eta_i^h\|_{\mathcal{Q}^N}}$$

By introducing the vectors of dofs $\underline{\boldsymbol{\eta}}_i^h \in \mathbb{R}^{N_p}$ and $\underline{\boldsymbol{s}}_i^h \in \mathbb{R}^{N_u}$, this becomes

$$\underline{\boldsymbol{s}}_i^h = \arg \sup_{\underline{\boldsymbol{v}}^h \neq \mathbf{0}} \frac{(\underline{\boldsymbol{\eta}}_i^h)^T B^h \underline{\boldsymbol{v}}^h}{\sqrt{(\underline{\boldsymbol{v}}^h)^T X_u^h \underline{\boldsymbol{v}}^h} \sqrt{(\underline{\boldsymbol{\eta}}_i^h)^T X_p^h \underline{\boldsymbol{\eta}}_i^h}} = \arg \sup_{\underline{\boldsymbol{v}}^h \neq \mathbf{0}} \frac{(\underline{\boldsymbol{\eta}}_i^h)^T B^h \underline{\boldsymbol{v}}^h}{\sqrt{(\underline{\boldsymbol{v}}^h)^T X_u^h \underline{\boldsymbol{v}}^h}}, \quad (3.12)$$

where the last equality holds true because the supremum does not depend on the norm of $\underline{\boldsymbol{\eta}}_i^h$. In Eq. (3.12), $B^h \in \mathbb{R}^{N_p \times N_u}$ is the matrix discretizing the bilinear form $b(\cdot, \cdot)$ (e.g., the divergence matrix D^h in the case of the Navier–Stokes equations) and X_u^h and X_p^h are norm matrices such that $\|v^h\|_{\mathcal{V}^h}^2 = (\underline{\boldsymbol{v}}^h)^T X_u^h \underline{\boldsymbol{v}}^h$ and $\|q^h\|_{\mathcal{Q}^h}^2 = (\underline{\boldsymbol{q}}^h)^T X_p^h \underline{\boldsymbol{q}}^h$ for every $v^h \in \mathcal{V}^h$ and $q^h \in \mathcal{Q}^h$, respectively. It is easy to see (Proposition 1 in [BMQR15]) that the vector achieving the supremum in Eq. (3.12) is

$$X_u^h \underline{\boldsymbol{s}}_i^h = (B^h)^T \underline{\boldsymbol{\eta}}_i^h. \quad (3.13)$$

Algorithm 3.3 Supremizers enrichment. The ortho operator performs the orthonormalization either via POD or Gram-Schmidt procedure.

Input: Basis matrices $\widehat{V}_u = [\widehat{\underline{\zeta}}_1^h | \dots | \widehat{\underline{\zeta}}_{N_u}^h] \in \mathbb{R}^{N_u^h \times N_u}$ and

$\widehat{V}_u = [\widehat{\underline{\eta}}_1^h | \dots | \widehat{\underline{\eta}}_{N_p}^h] \in \mathbb{R}^{N_p^h \times N_p}$, matrices \widehat{B}^h and \widehat{X}_u^h

Output: Enriched basis $\widehat{V}_u^\dagger \in \mathbb{R}^{N_u^h \times (N_u + N_p)}$

- 1: **for** $i = 1, \dots, N_p$ **do**
 - 2: Solve $\widehat{X}_u^h \underline{\mathbf{s}}_i^h = (\widehat{B}^h)^T \widehat{\underline{\eta}}_i^h$
 - 3: Set $\widehat{V}_u^\dagger = \text{ortho}[\widehat{\underline{\zeta}}_1^h | \dots | \widehat{\underline{\zeta}}_{N_u}^h | \widehat{\underline{\mathbf{s}}}_1^h | \dots | \widehat{\underline{\mathbf{s}}}_{N_p}^h]$
-

Indeed, by the Cauchy–Schwarz inequality, we find

$$\frac{(\underline{\boldsymbol{\eta}}_i^h)^T B^h \underline{\mathbf{v}}^h}{\sqrt{(\underline{\mathbf{v}}^h)^T X_u^h \underline{\mathbf{v}}^h}} = \frac{(\underline{\mathbf{v}}^h)^T X_u^h \underline{\mathbf{s}}_i^h}{\|(X_u^h)^{1/2} \underline{\mathbf{v}}^h\|_2} = \frac{((X_u^h)^{1/2} \underline{\mathbf{s}}_i^h)^T (X_u^h)^{1/2} \underline{\mathbf{v}}^h}{\|(X_u^h)^{1/2} \underline{\mathbf{v}}^h\|_2} \leq \|(X_u^h)^{1/2} \underline{\mathbf{s}}_i^h\|_2 = \|\underline{\mathbf{s}}_i^h\|_{\mathcal{V}}.$$

It can be shown that by considering the augmented basis $V_u^\dagger = [V_u | \underline{\mathbf{s}}_1^h | \dots | \underline{\mathbf{s}}_{N_p}^h] \in \mathbb{R}^{N_u^h \times (N_u + N_p)}$ (where $V_u = [\underline{\zeta}_1^h | \dots | \underline{\zeta}_{N_u}^h]$ is the basis matrix for u computed by POD) the reduced problem is well-posed.

In the context of parametrized PDEs, this procedure presents difficulties that undermine the computational advantage deriving from a reduced dimensionality of the search space for the solution. Indeed, in general, Eq. (3.13) reads $X_u^h(\underline{\boldsymbol{\mu}}) \underline{\mathbf{s}}_i^h(\underline{\boldsymbol{\mu}}) = (B^h(\underline{\boldsymbol{\mu}}))^T \underline{\boldsymbol{\eta}}_i^h$. In other words, the constraint matrix D^h and the norm matrix X_u^h might be $\underline{\boldsymbol{\mu}}$ -dependent, and consequently, the supremizer itself would depend on $\underline{\boldsymbol{\mu}}$. Therefore, applying the supremizers procedure exactly is problematic because it requires solving N_p^h times problem (3.13) in the online phase of the algorithm. Another possibility is to consider an approximate approach. For example, in [BMQR15] the authors propose to solve problem (3.13) by replacing $\underline{\boldsymbol{\eta}}_i^h$ with the vector of dofs of every snapshot for p and by evaluating the matrices at the corresponding parameters. The set of supremizers is then orthonormalized by POD in order to keep the size of basis V_u^\dagger small.

Here, we consider a different approach. Owing to the fact that we are mostly interested in geometrical parameters mapping reference domains $\widehat{\Omega}$ into their physical configuration $\Omega(\underline{\boldsymbol{\mu}})$ (see Section 3.3), we define matrices \widehat{X}_u^h and \widehat{B}^h which are defined on $\widehat{\Omega}$. Then, the i^{th} supremizer is found as

$$\widehat{X}_u^h \widehat{\underline{\mathbf{s}}}_i^h = (\widehat{B}^h)^T \widehat{\underline{\boldsymbol{\eta}}}_i^h, \quad (3.14)$$

where $\widehat{\underline{\boldsymbol{\eta}}}_i^h$ is the i^{th} basis function for the dual variable on the reference domain; we remark that, in this case, the problem is independent of any parameter $\underline{\boldsymbol{\mu}}$. Basis function $\widehat{\underline{\boldsymbol{\eta}}}_i^h$ is the i^{th} basis function for p in the reference domain.

Remark 3.3. Although Eq. (3.14) is based on the assumption that we are dealing with

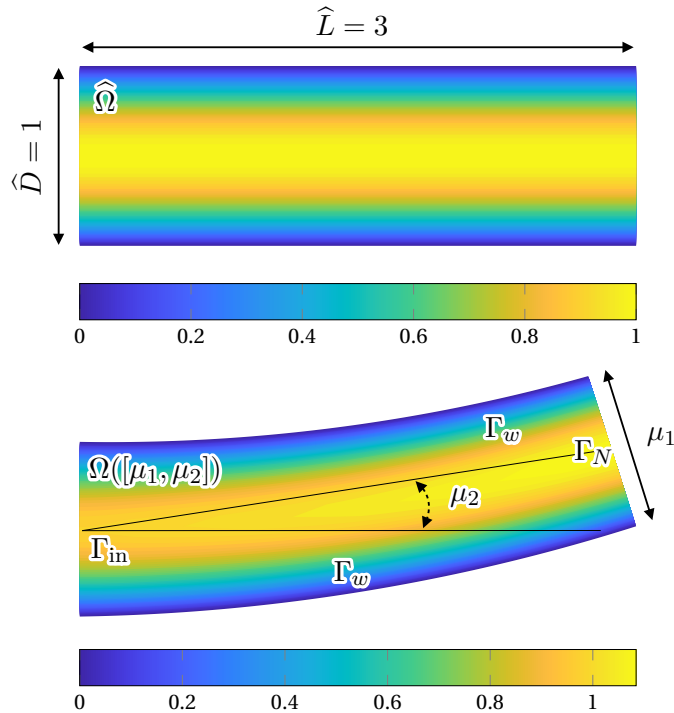


Figure 3.4 – Magnitude of the velocity field of the steady Stokes equations on the reference domain $\hat{\Omega}$ and on the deformed one $\Omega([\mu_1, \mu_2])$. The geometrical parameters are the outlet diameter μ_1 and the bending angle μ_2 .

geometrical parameters, it is actually applicable to a variety of cases in which other types of parameters are considered. For example, when considering the Navier–Stokes equations, the only scenarios which are not suitable to be treated with this approach are those in which the divergence term depends on some physical parameter. Such cases are rare: more commonly, one is interested in varying the density and viscosity of the fluid or other parameters governing the boundary conditions.

Our supremizers enrichment strategy is summarized in Algorithm 3.3.

3.5 The steady Stokes equations in a deformable pipe

With the purpose of illustrating the concepts presented in this chapter, we consider a simple two-dimensional steady case. This also serves as an introductory example to the MOR strategy for cardiovascular simulations discussed in Chapter 4. As a matter of fact, many of the conclusions drawn for this elementary problem can be translated to the more involved framework that we use in the remainder of the thesis.

Let us introduce a reference domain $\hat{\Omega} = (0, 3) \times (-0.5, 0.5)$, which represents a two

dimensional pipe characterized by length $\widehat{L} = 3$ and diameter $\widehat{D} = 1$. Moreover, we consider the deformed configuration $\Omega(\underline{\boldsymbol{\mu}}) = \Phi(\widehat{\Omega}; \underline{\boldsymbol{\mu}})$, where the vector of geometrical parameters $\underline{\boldsymbol{\mu}} = [\mu_1, \mu_2]$ is composed of the outlet diameter μ_1 and the bending angle μ_2 and, as shown in Fig. 3.4. The transformation $\Phi(\cdot, \underline{\boldsymbol{\mu}}) = \varphi_b(\cdot; \mu_2) \circ \varphi_s(\cdot; \mu_1)$ is written as composition of two distinct maps which perform the scaling of the outlet (φ_s) and the bending of the pipe (φ_b). In particular, we have

$$\begin{aligned} (\varphi_s(\widehat{\mathbf{x}}; \mu_1))_1 &= \widehat{x}_1, \\ (\varphi_s(\widehat{\mathbf{x}}; \mu_1))_2 &= \frac{1}{\widehat{L}} \left[\widehat{L} - (1 - \mu_1/\widehat{D})x_1 \right] x_2, \end{aligned} \quad (3.15)$$

and

$$\begin{aligned} (\varphi_b(\widehat{\mathbf{x}}; \mu_2))_1 &= (r + x_2) \sin(\alpha), \\ (\varphi_b(\widehat{\mathbf{x}}; \mu_2))_2 &= (r + x_2) \cos(\alpha) - r, \end{aligned} \quad (3.16)$$

where $r = \widehat{L}/\mu_2$ and $\alpha = \mu_2 x_1/\widehat{L}$. In Fig. 3.4 we show, on the top, the reference domain $\widehat{\Omega}$ and on the bottom the deformed one $\Omega([\mu_1, \mu_2])$ corresponding to the choice $\mu_1 = 0.9$ and $\mu_2 = -0.3$.

For a given $\underline{\boldsymbol{\mu}}$, we are interested in solving the steady Stokes equations, i.e., finding \mathbf{u} and p which depend on $\underline{\boldsymbol{\mu}}$ such that

$$\begin{aligned} -\mu_f \Delta \mathbf{u} + \nabla p &= \mathbf{0} && \text{in } \Omega(\underline{\boldsymbol{\mu}}), \\ \nabla \cdot \mathbf{u} &= 0 && \text{in } \Omega(\underline{\boldsymbol{\mu}}), \\ \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_D(\underline{\boldsymbol{\mu}}), \\ (\mu_f \nabla u - p \mathbf{I}) \mathbf{n} &= \mathbf{0} && \text{on } \Gamma_N(\underline{\boldsymbol{\mu}}), \end{aligned}$$

where $\mu_f = 1$ and $\Gamma_D(\underline{\boldsymbol{\mu}}) = \Gamma_{\text{in}}(\underline{\boldsymbol{\mu}}) \cup \Gamma_w(\underline{\boldsymbol{\mu}})$ (we refer the reader to Fig. 3.4 for the symbols regarding the portions of the boundary). The Dirichlet data \mathbf{g} is such that we impose a parabolic profile on Γ_{in} with maximum velocity $U = 1$ and $\mathbf{g} = \mathbf{0}$ on Γ_w . We restrict the range of admissible parameters to the intervals $\mu_1 \in [-0.8, 1.2]$ and $\mu_2 \in [-0.5, 0]$.

The discretized Stokes equations are obtained by the FE method via a discretization strategy similar to the one presented in Section 1.2. This leads to the high-fidelity problem

$$\begin{bmatrix} A^h(\underline{\boldsymbol{\mu}}) & (D^h(\underline{\boldsymbol{\mu}}))^T \\ D^h(\underline{\boldsymbol{\mu}}) & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^h \\ \underline{\mathbf{p}}^h \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{g}}^h(\underline{\boldsymbol{\mu}}) \\ \underline{\mathbf{0}} \end{bmatrix}, \quad (3.17)$$

where $A^h(\underline{\boldsymbol{\mu}}) \in \mathbb{R}^{N_u^h \times N_u^h}$ and $D^h(\underline{\boldsymbol{\mu}}) \in \mathbb{R}^{N_p^h \times N_u^h}$ are the $\underline{\boldsymbol{\mu}}$ -dependent stiffness and divergence matrices and $\underline{\mathbf{g}}^h(\underline{\boldsymbol{\mu}})$ is a vector encoding the Dirichlet boundary conditions; N_u^h and N_p^h are the number of dofs for velocity and pressure, respectively.

Our strategy to attempt the numerical reduction of Problem (3.17) is based on the

3.5. The steady Stokes equations in a deformable pipe

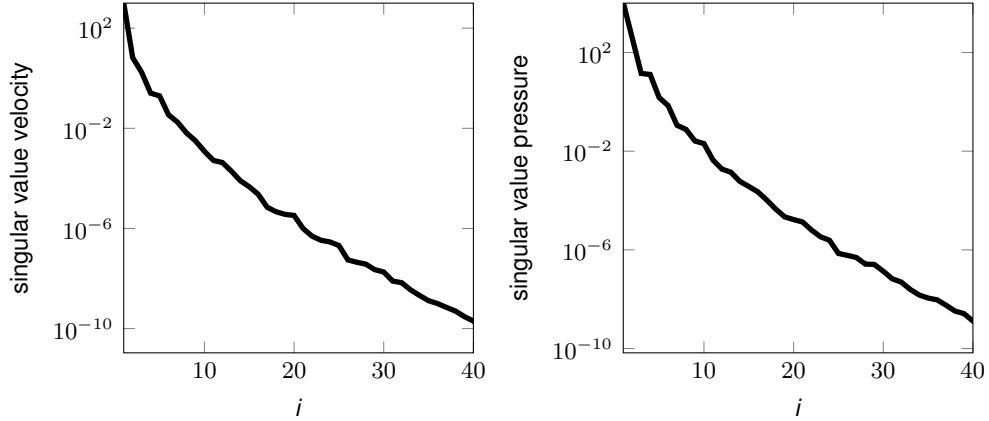


Figure 3.5 – Decay of the first 40 velocity (left) and pressure (right) singular values.

Table 3.1 – Bases sizes as functions of POD tolerances.

	$\varepsilon = 10^{-3}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-5}$	$\varepsilon = 10^{-6}$	$\varepsilon = 10^{-7}$	$\varepsilon = 10^{-8}$
velocity	7	10	14	20	26	32
pressure	4	5	7	10	13	16
stiffness	5	6	9	11	13	14
divergence	4	5	6	6	8	8

RB method. In particular, we solve the high-fidelity model for $N_s = 100$ parameters in $\mathcal{D} = [-0.8, 1.2] \times [-0.5, 0]$; each snapshot parameter $\underline{\mu}_1, \dots, \underline{\mu}_{N_s}$ is found at the grid points obtained by subdividing the intervals for μ_1 and μ_2 in 9 sub-intervals. The snapshots for velocity and pressure are organized in matrix form as $\widehat{S}_u = [\underline{\mathbf{u}}_1^h | \dots | \underline{\mathbf{u}}_{N_s}^h] \in \mathbb{R}^{N_u^h \times N_s}$ and $\widehat{S}_p = [\underline{\mathbf{p}}_1^h | \dots | \underline{\mathbf{p}}_{N_s}^h] \in \mathbb{R}^{N_p^h \times N_s}$. We remark that both \widehat{S}_u and \widehat{S}_p are denoted with a hat notation because the velocity and pressure snapshots are interpreted as functions in the reference domain (the former are scaled by means of the Piola transformation, as discussed in Section 3.3.1). The two basis matrices $\widehat{V}_u \in \mathbb{R}^{N_u^h \times N_s}$ and $\widehat{V}_p \in \mathbb{R}^{N_p^h \times N_s}$ are computed by POD with tolerances ε_u and ε_p . During the snapshots generation phase, we also construct a database composed of $A^h(\underline{\mu}_1), \dots, A^h(\underline{\mu}_{N_s})$ and $D^h(\underline{\mu}_1), \dots, D^h(\underline{\mu}_{N_s})$, which are used in the MDEIM procedure to increase the efficiency of the assembly part of the algorithm. In Fig. 3.5 we show the decay of the singular values of $\widehat{S}_u, \widehat{S}_p$. Owing to criterion (3.4), the fast decay of the singular values indicates a smaller size of the associated basis (see Table 3.1) and, consequently, this is a sign that the problem is *reducible*. We remark that $N_u^h = 39'042$ and $N_p^h = 9'600$ and, therefore, the reduced bases feature a considerably smaller size. The velocity basis \widehat{V}_u is augmented with the supremizers (one for each pressure mode) as described in Section 3.4. The total basis matrix is denoted \widehat{V}_u^\dagger .

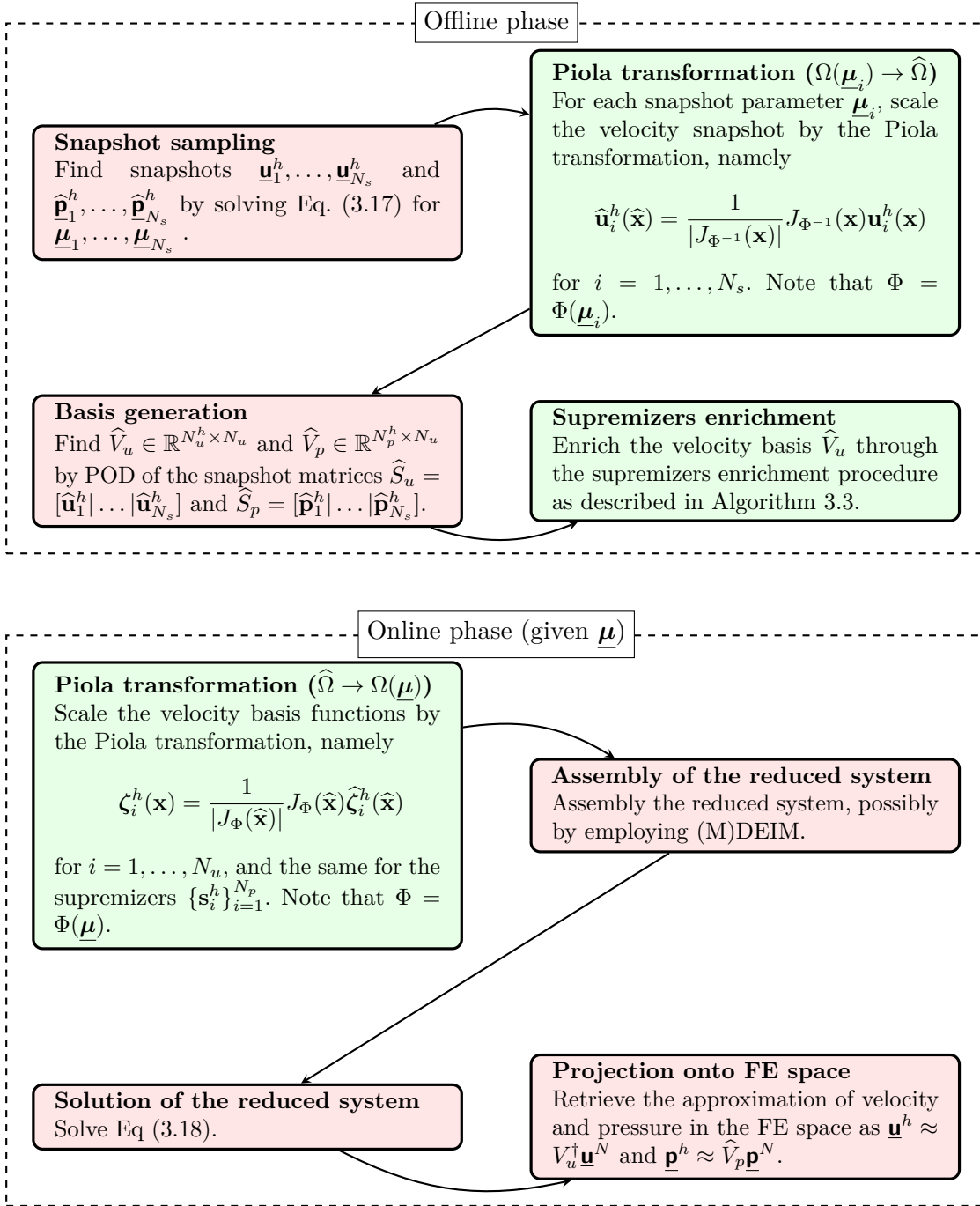


Figure 3.6 – Diagram of the RB algorithm. The green boxes are specific to the application of the steady Stokes equations in parametrized domains.

3.5. The steady Stokes equations in a deformable pipe

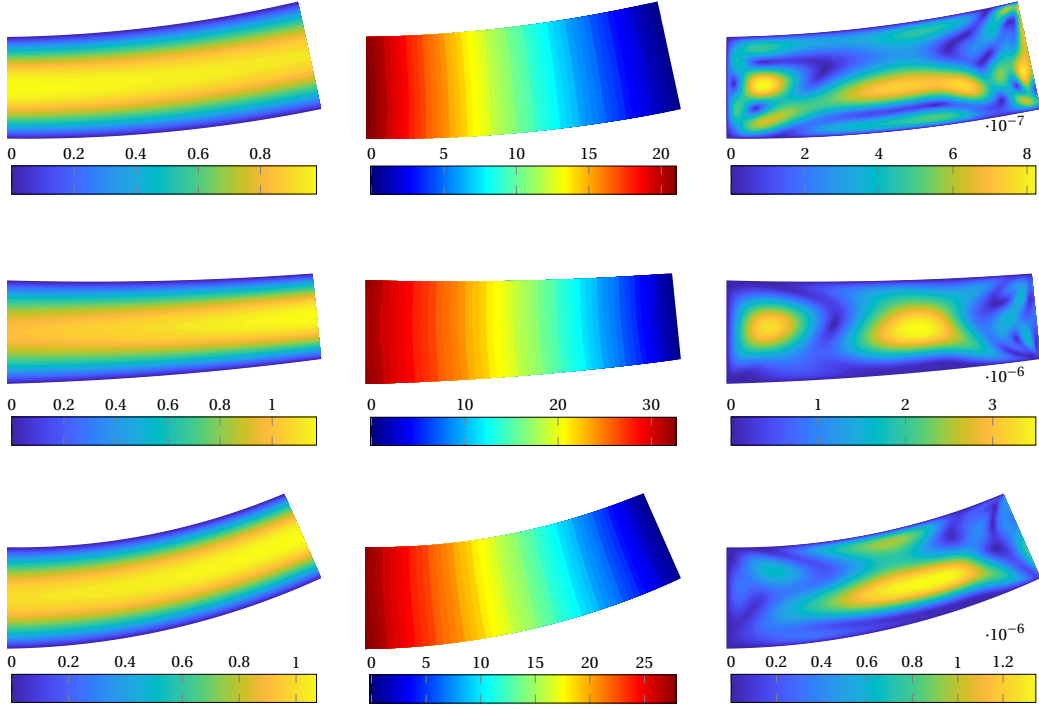


Figure 3.7 – RB velocity field magnitude (left column), RB pressure distribution (middle column) and velocity error magnitude with respect to FE solution (right column), for three different choices of the geometrical parameters. First row: $\mu_1 = -0.21$ and $\mu_2 = 1.08$. Second row: $\mu_1 = -0.10$ and $\mu_2 = 0.83$. Third row: $\mu_1 = -0.41$ and $\mu_2 = 0.92$.

The online phase consists of the assembly and solution of reduced system

$$\begin{bmatrix} A^N(\underline{\mu}) & (D^N(\underline{\mu}))^T \\ D^N(\underline{\mu}) & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}^N \\ \underline{\mathbf{p}}^N \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{g}}^N(\underline{\mu}) \\ \mathbf{0} \end{bmatrix}, \quad (3.18)$$

where $A^N(\underline{\mu}) = (V_u^\dagger)^T A^h(\underline{\mu}) V_u^\dagger$, $D^N(\underline{\mu}) = (\widehat{V}_p)^T D^h(\underline{\mu}) V_u^\dagger$ and $\underline{\mathbf{g}}^N(\underline{\mu}) = (V_u^\dagger)^T \underline{\mathbf{g}}^N(\underline{\mu})$, while $\underline{\mathbf{u}}^N$ and $\underline{\mathbf{p}}^N$ are the vectors of reduced dofs. We remark that basis V_u^\dagger is first scaled by the Piola transformation at the start of the simulation and, therefore, it is not written with the hat notation. To ensure an efficient online phase, vectors and matrices in Eq. (3.18) are assembled by DEIM and MDEIM, respectively. The entire RB pipeline—offline and online phases—is summarized in Fig. 3.6. The red boxes in the figure refer to steps that are typical to the standard RB approximation and the green ones are peculiar to this particular problem. Specifically, these stages are necessary because we are dealing with a saddle-point problem (hence the supremizers enrichment) in a parametrized domain (which requires the use of the Piola transformation).

Fig. 3.7 shows the velocity magnitude, pressure, and velocity error magnitude with respect to the FE solution obtained for three different choices of $\underline{\mu}$ (which are not included in the original sets of parameters considered during the snapshots generation phase). These plots refer to the choices $\varepsilon_u = \varepsilon_p = 10^{-6}$. The velocity and pressure manifest a qualitatively physical behavior: the former attains higher magnitude values when the outlet is narrower—which is a consequence of the incompressibility of the fluid—whereas the latter decreases in the downstream direction and is characterized by a gradient oriented as the centerline of the pipe itself. The velocity error magnitude is, in all cases, close to the POD tolerance, which shows that the set of snapshots is adequate to represent the variability of the solutions with respect to the geometrical parameters.

An analysis of the effect of the Piola transformation and supremizers enrichment is carried out in Fig. 3.8. All the errors are computed over 20 solutions of the FE and RB problems for values of the parameters $\underline{\mu}$ randomly sampled in the acceptable range $\mathcal{D} = [-0.8, 1.2] \times [-0.5, 0]$. In terms of runtime, the FE simulation takes on average $T = 0.6402$ s, and the speedups obtained by considering POD tolerances (the same for velocity and pressure) ranging from 10^{-3} to 10^{-8} are, respectively, 80, 69, 62, 51, 41, and 32. On the top left, we show that the L^2 error in pressure (L2u) and the H^1 error in velocity (H1u) decay with the same rate with respect to the POD tolerance, regardless of the use of the Piola transformation. This is beneficial to decrease the velocity basis size necessary to achieve a fixed H1u error (top right); however, due to the additional online step which is required to transform the basis from $\hat{\Omega}$ to $\Omega(\underline{\mu})$ (see Fig. 3.6), the runtime is higher in the case of the Piola transformation (bottom left). Finally, we observe that adding the supremizers for the velocity is required to preserve the stability of the discretized problem. On the bottom right, we indeed show that L2p and H1p are extremely large without stabilization, which is a consequence of the unboundedness of the condition number of the system matrix.

Remark 3.4. The use of the Piola transformation would allow, in principle, to discard the continuity equation in Eq. (3.18), as the velocity is guaranteed to be divergence-free in the physical configuration $\Omega(\underline{\mu})$. This would also imply that the saddle-point structure of the problem would be lost, thus making the pressure supremizers unnecessary. However, in these numerical results and the ones presented in the following chapters, the continuity equation is maintained for two main reasons. Firstly, the computational mesh in the reference configuration does not map into the one in the physical configuration. This is because nonaffine transformations—such as the ones that we consider in this thesis—do not map lines into lines and planes into planes (in other words, a triangular mesh is mapped to an invalid one featuring curved elements). The computational mesh in the deformed configuration is built upon the image of the reference set of vertices through the geometrical map. Consequently, the Piola transformation computed by using the analytical Jacobian is inexact, and the velocity modes are only approximately divergence-free. Secondly, retaining the continuity equations allows for the computation of the pressure, which would otherwise need to be found a posteriori.

3.5. The steady Stokes equations in a deformable pipe

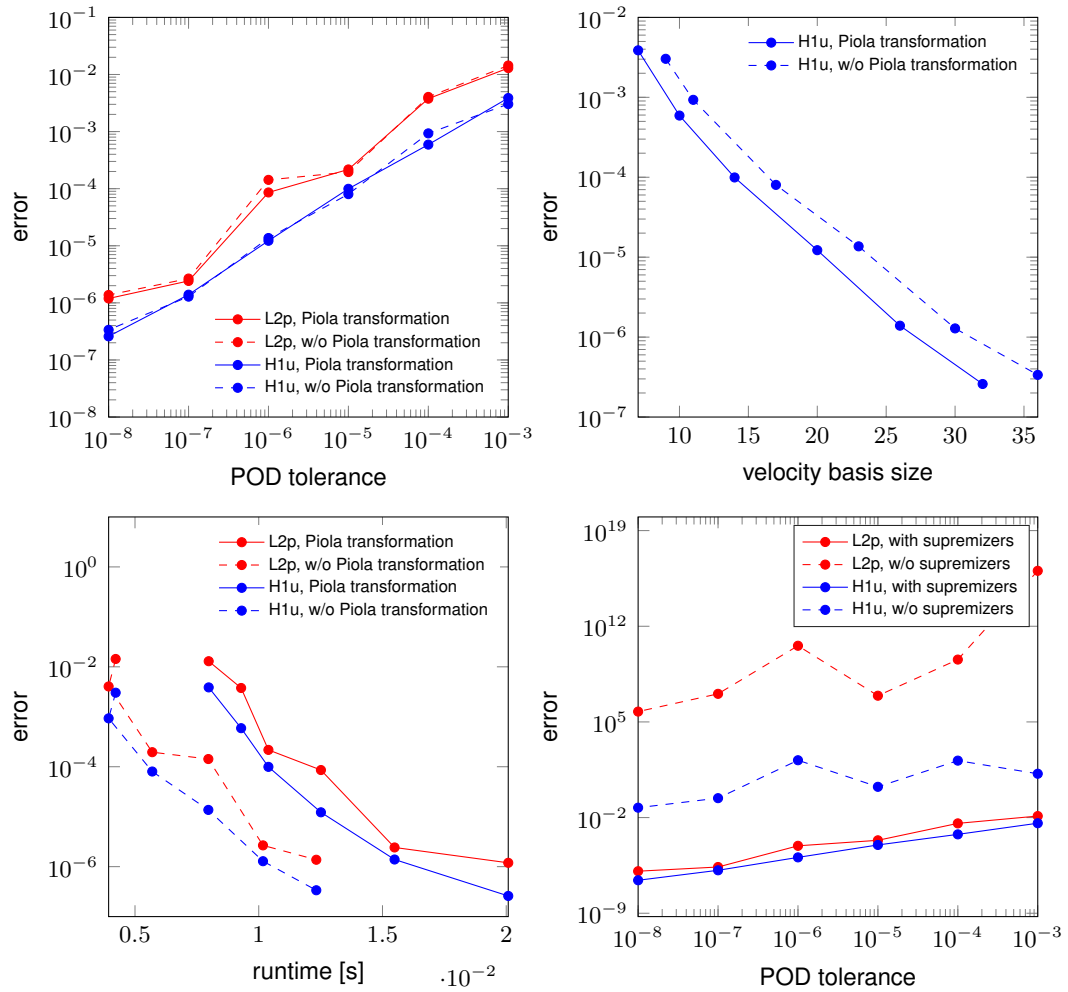


Figure 3.8 – Top left: L^2 error in pressure (L2p) and H^1 error in velocity (H1u), with and without scaling of the velocity by the Piola transformation vs POD tolerance (the same for pressure and velocity). Top right: H^1 errors in velocity vs basis size, with and without Piola scaling. Bottom left: L2p and H1u errors vs online runtime in seconds. Bottom right: L2p and H1u errors with and without supremizers vs POD tolerance.

3.6 Concluding remarks

The RB method is a MOR technique that aims at decreasing the number of variables involved in the solution of parametrized PDEs. During the computationally expensive offline phase, a large number of snapshots is generated by solving the FOM—e.g., the FE discretization of the differential problem of interest—multiple times. The snapshots are then employed to generate a reduced basis, either by greedy algorithms or (as we do in this thesis) by the POD method. The projection of the FOM onto the reduced space gives rise to a system that is characterized by a considerably smaller number of variables and which is assembled and solved during the online phase of the algorithm.

In this chapter, we described the method, and we also highlighted some aspects which require particular care when dealing with geometrical parameters and saddle-point problems. Specifically, we discussed how considering deformable domains is usually problematic due to the lack of an affine decomposition of the terms appearing in the PDEs. Furthermore, when dealing with divergence-free vector fields, we motivated using the Piola transformation from the physical configuration to the reference one (in the offline phase) and vice versa (in the online phase). Lastly, we considered a case of steady flow in a two-dimensional pipe modeled by the Stokes equations. This problem is relevant because, despite being simple, it features all the concepts presented in the chapter, and it is also strictly related to the topics treated in the subsequent parts of this thesis. We showed that by employing the RB approach, we were able to obtain considerable speedup with respect to the FE method while retaining good accuracy. We also analyzed the effects of the Piola transformation and supremizers enrichment in terms of efficiency and accuracy.

4 Modular Model Order Reduction of blood flow

In this chapter, we present a strategy for the MOR of blood flow in arteries based on a geometrical approximation of the vessel. Specifically, the geometry of interest is approximated as the composition of simple subdomains which are obtained from the parametrized deformation of a handful of geometrical building blocks (e.g., reference tubes and bifurcations). Each building block is equipped with a set of RB basis functions for velocity and pressure, which, after being mapped to the physical configuration, are used to find the local solutions in each subdomain. The advantage of this technique is that the number of variables involved is often much smaller than the number of dofs that are necessary to obtain accurate results using, for example, the FE method.

The chapter is structured as follows. In Section 4.1 we present the main idea of the approach; we also discuss other methods that take an analogous perspective on the task of the MOR of cardiovascular flow (or of systems with recurrent elements in general). In Section 4.2, we define the concept of modular approximation of arteries; we also address the numerical solution of the Navier–Stokes equations on the decomposed geometries by the FE method and the nonconforming Domain Decomposition method presented in Chapter 2. Furthermore, in Section 4.3 we devise an ad-hoc preconditioner that takes advantage of the peculiar block structure of the global system matrix. It is worth noticing that, although the chapter focuses on a ROM, addressing the solution of the partitioned problem with the FE method is necessary, as the RB functions in the subdomains are generated by POD of local solutions obtained from global problems in decomposed domains. This strategy of data collection (offline phase) is discussed in Section 4.4. In the same section, we also delineate the algorithm for the approximation of the global solution on a decomposed target geometry using the ROM (online phase). Finally, in Section 5.5 we summarize the main points of the method.

With the exception of Section 4.1 and some extensions, the contents of this chapter have been presented in [PPMD20].

4.1 An overview of Reduced Order Methods for networks of blocks with recurring features

The method presented in this chapter is rooted in the idea that potentially complex arterial trees are suitable to be decomposed into simpler elements, the blood flow through which shares recurring features. For example, it is reasonable to assume that the blood dynamics far from bifurcations and in similarly shaped portions of vessels with comparable sizes is characterized by approximately the same profile. However, in the usual high-fidelity discretization approach by the FE or Finite Volume methods, these properties of blood flow are not exploited, and a large number of dofs is often required to obtain meaningful results.

Maday and Rønquist [MR02] proposed a technique based on these observations to attempt the MOR of flow systems such as arteries. In the expensive offline phase of the method, some elementary geometrical entities are defined and equipped with one or more sets of RB functions. Given a complex geometry, this is divided into simple blocks, which are obtained from parametrized deformations of the reference ones. The local flow solutions are then obtained through the combination of the local RB functions, and the global one is found by coupling the local solutions through Lagrange multipliers. This technique is named Reduced Basis Element (RBE) method and has also been considered in [LMR06a, LMR06c, LMR07, IRQ10]. The method presented in this chapter can be seen as a particular implementation of the RBE method in which the coupling conditions are enforced using the discretization strategy discussed in Chapter 2.

Strictly related to the RBE method is the Reduced Basis Hybrid (RBH) method [IQR12, Iap12]. While the setting in every subdomain is the same as in the RBE method, in this case, the coupling strategy is based on a solution of the global flow problem on a coarse mesh, which is used to ensure the continuity of the stresses at the interfaces. Specifically, the global solution in the online phase is found as a linear combination of the usual RB functions (defined on the fine mesh) and the coarse solutions in each subdomain. The online phase amounts to finding the coefficients of such a linear combination. Although this approach allows circumventing defining the space of coupling Lagrange multipliers, it suffers from the drawback that it requires to deal with multiple levels of mesh refinements and that the coarse global problem must be solved in the online phase of the method. For this reason, we opt to employ a perspective more similar to that of the RBE method. It is worth noticing that, despite the RBE method being originally proposed with the cardiovascular context in mind, its application has been mostly restricted to two-dimensional cases of the steady Stokes equations. To the best of our knowledge, the results presented in [PPMD20] and in Chapter 5 are the first ones dealing with the approximation of the three-dimensional unsteady Navier–Stokes equations.

We end this section by mentioning two additional types of ROMs that are relevant to this thesis. Another method to tackle geometries with repetitive or recurring features is

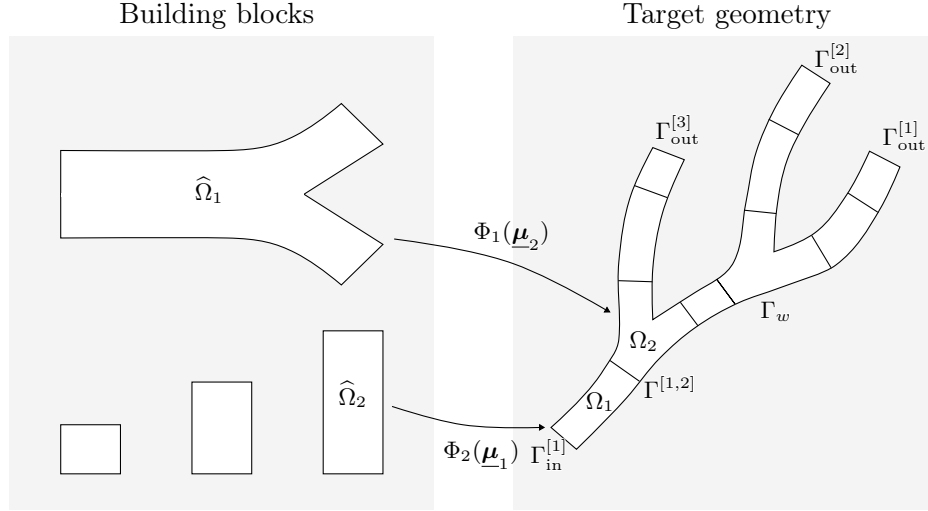


Figure 4.1 – Sketch of the Domain Decomposition of a target geometry. Each block in the target geometry is found from the parametrized geometrical deformation of a small number of reference building blocks.

based on the RBE method at the subdomain level, and the static condensation approach [HKP13] for the coupling. This technique has mostly been investigated in the context of structural simulations, see, e.g., [VHK⁺15, SP16]. Strictly related to the cardiovascular context is instead the Transversally Enriched Pipe Element Method (TEPEM), initially proposed in [MABB⁺17, MABBF19] and further employed for uncertainty quantification in [GMAB⁺20]. The main idea of the approach is to consider a domain decomposition of the geometry of interest into pipe-type elements. The longitudinal and transversal directions of the flow are approximated separately; the splitting into principal and transversal components is also exploited in, e.g., [PEV10, GPV18]. A common strategy for the discretization of these components employs a standard FE approximation in the longitudinal direction—which is assumed to be the dominant direction of the flow—and high-order polynomial functions in the transversal one.

4.2 Modular Domain Decomposition of arteries

We introduce a library of *building blocks* $\hat{\Omega}_i$, $i = 1, \dots, N_{bb}$. In the context of cardiovascular simulations, these reference building blocks are model cylinders and bifurcations, as shown in Fig. 4.1. The choice of building blocks is arbitrary. Here, we focus on simple blocks such as the ones depicted in Fig. 4.1 and Fig. 4.2, as we are mostly interested in presenting a proof of concept. In some scenarios, an attractive choice would be to enrich the set of building blocks with elements characterizing pathological conditions with geometrical abnormalities (e.g., stenoses). However, it is worth noticing that considering a large number of building blocks comes at the cost of a more expensive offline phase of

the ROM that we present in Section 4.4, as for each of the blocks we need to generated an ad-hoc reduced basis.

The target geometry is then approximated as a modular composition of subdomains $\Omega \approx \Omega^m(\mathcal{M}) = \bigcup_{j=1}^{N_\Omega} \Omega_j(\underline{\boldsymbol{\mu}}_j)$. Here, $\Omega_j := \Phi_{z(j)}(\widehat{\Omega}_{z(j)}; \underline{\boldsymbol{\mu}}_j)$ is an open and bounded subdomain obtained by applying a prescribed parametrized geometrical deformation $\Phi_{z(j)}$ to the $z(j)^{\text{th}}$ building block, $z : [1, \dots, N_\Omega] \mapsto [1, \dots, N_{\text{bb}}]$ is a map from the indices of the subdomains in the target geometry to the indices of the building blocks, and $\mathcal{M} := \{\underline{\boldsymbol{\mu}}_j\}_{j=1}^{N_\Omega}$ is the set of geometrical parameters. In the following, we indicate $z(j) := z_j$ for brevity. Each vector of parameters $\underline{\boldsymbol{\mu}}_j$ belongs to a space $\mathcal{D}_{z_j} \subset \mathbb{R}^{N_{\mu, z_j}}$ whose dimensionality depends on the corresponding reference building block. For each $i = 1, \dots, N_{\text{bb}}$ and given a parameter vector $\underline{\boldsymbol{\mu}}$, we focus on geometrical deformations of the form

$$\Phi_i(\widehat{\mathbf{x}}; \underline{\boldsymbol{\mu}}) = Q(\underline{\boldsymbol{\mu}})\boldsymbol{\varphi}_i(\widehat{\mathbf{x}}; \underline{\boldsymbol{\mu}}) + \mathbf{t}(\underline{\boldsymbol{\mu}}), \quad \forall \widehat{\mathbf{x}} \in \widehat{\Omega}_i, \quad (4.1)$$

where $Q(\underline{\boldsymbol{\mu}})$ is a rotation and scaling matrix, $\mathbf{t}(\underline{\boldsymbol{\mu}})$ is a translation vector, and $\boldsymbol{\varphi}_i(\cdot; \underline{\boldsymbol{\mu}})$ is a nonaffine geometrical deformation. We remark that the expression of Φ_i is such that it is composed of an affine part, which consists of the rotation, scaling and translation and which is independent of the reference building block $\widehat{\Omega}_i$, and a nonaffine part that is instead specific to $\widehat{\Omega}_i$. The types of building blocks we consider and the corresponding admissible nonaffine deformations are depicted in Fig. 4.2. The geometrical parameters in the tubes are the angle of the outlet normal α —due to the axial symmetry and to the rotation matrix Q in Eq. (4.1), a single angle is sufficient to represent a bending in any direction—in the deformed configuration α and the deformed lengths L and outlet radius R_{out} . In the bifurcation, the geometrical parameters are the angles describing the rotation of the reference outlet normals $\widehat{\mathbf{n}}_1$ and $\widehat{\mathbf{n}}_2$ onto the outlet normals \mathbf{n}_1 and \mathbf{n}_2 (i.e., three Euler angles per outlet, that is six geometrical parameters in total). The subdomains in the target geometry satisfy $\Omega_l(\underline{\boldsymbol{\mu}}_l) \cap \Omega_m(\underline{\boldsymbol{\mu}}_m) = \emptyset$ if $l \neq m$, and we define the interface $\Gamma^{[jm]}(\underline{\boldsymbol{\mu}}_j, \underline{\boldsymbol{\mu}}_m) := \overline{\Omega}_j(\underline{\boldsymbol{\mu}}_j) \cap \overline{\Omega}_m(\underline{\boldsymbol{\mu}}_m)$. The building blocks in the reference configuration are designed with circular inlet and outlet faces; the geometrical deformations are chosen such that the interfaces are circles for every possible choice of the geometrical parameters.

Remark 4.1. Although $\Gamma^{[lm]}$ and $\Gamma^{[ml]}$ represent the same physical surface, it is still beneficial to differentiate between the two as we associate with each interface the vectors \mathbf{n}_{lm} and \mathbf{n}_{ml} , i.e., the outward normal unit vectors with respect to Ω_l and Ω_m (clearly, $\mathbf{n}_{lm} = -\mathbf{n}_{ml}$). This distinction allows to simplify the notation in the weak formulation derived in Section 4.2.2.

For every subdomain $\Omega_j(\underline{\boldsymbol{\mu}}_j)$, we introduce the set of indices of the neighboring subdomains $N(j)$, and the sets $I_{\text{in}}(j)$ and $I_{\text{out}}(j)$ such that $\partial\Omega_j(\underline{\boldsymbol{\mu}}_j) \cap \Gamma_{\text{in}}^{[i]} \neq \emptyset$ for all $i \in I_{\text{in}}(j)$, and $\partial\Omega_j(\underline{\boldsymbol{\mu}}_j) \cap \Gamma_{\text{out}}^{[i]} \neq \emptyset$ for all $i \in I_{\text{out}}(j)$; here $\Gamma_{\text{in}}^{[i]}$ and $\Gamma_{\text{out}}^{[i]}$ represent the i^{th} inlet and outlet boundary of the geometry, as shown in Fig. 4.1. The dependance of the subdomains Ω_j

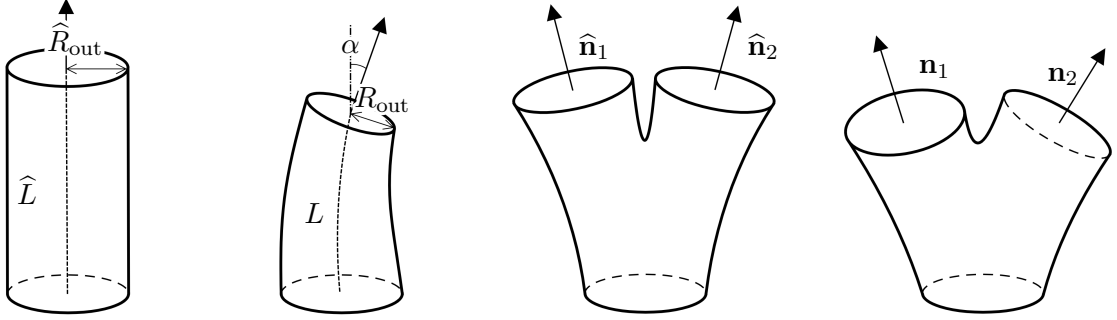


Figure 4.2 – Types of reference building blocks and affine transformations. On the left, tubes. On the right, bifurcation.

and interfaces $\Gamma^{[jm]}$, $\Gamma_{\text{in}}^{[i]}$ and $\Gamma_{\text{out}}^{[i]}$ on the geometrical parameters \mathcal{M} is omitted unless ambiguity arises in the remainder of the chapter.

4.2.1 Parametrized geometrical maps for the reference building blocks

We restrict ourselves to cases where the nonaffine deformations associated with the tubes are given in analytical form. In particular, we consider the three-dimensional versions of the maps defined in Section 3.5 with the slight modification that we allow for variations in the arclength (whereas formulas (3.15) and (3.16) are designed to preserve this quantity). In particular, we have that, for every i corresponding to a tube, $\varphi_i(\cdot; \underline{\mu}) = \varphi_{\text{b},i}(\cdot; \mu_2, \mu_3) \circ \varphi_{\text{s},i}(\cdot; \mu_1, \mu_2)$ with $\underline{\mu} = [\mu_1, \mu_2, \mu_3]$. Transformations $\varphi_{\text{s},i}(\cdot; \mu_1, \mu_2)$ and $\varphi_{\text{b},i}(\cdot; \mu_2, \mu_3)$ model the scaling of the outlet and arclength of the tube and its bending. The nonaffine scaling function is defined as

$$\begin{aligned} (\varphi_{\text{s},i}(\widehat{\mathbf{x}}; \mu_1, \mu_2))_1 &= \frac{1}{\widehat{L}_i} \left[\widehat{L}_i - (1 - \mu_1/\widehat{R}_i)\widehat{x}_3 \right] \widehat{x}_1, \\ (\varphi_{\text{s},i}(\widehat{\mathbf{x}}; \mu_1, \mu_2))_2 &= \frac{1}{\widehat{L}_i} \left[\widehat{L}_i - (1 - \mu_1/\widehat{R}_i)\widehat{x}_3 \right] \widehat{x}_2, \\ (\varphi_{\text{s},i}(\widehat{\mathbf{x}}; \mu_1, \mu_2))_3 &= \widehat{x}_3 \mu_2, \end{aligned} \quad (4.2)$$

where \widehat{R}_i and \widehat{L}_i are the diameter and the length of the tube in the reference configuration, μ_1 is the radius of the outlet in the physical domain, and μ_2 is the ratio between the arclength in the physical and reference configurations. We remark that the axis of the reference tube is parallel to the \widehat{x}_3 -axis and that the inlet is laid on the \widehat{x}_1 - \widehat{x}_2 plane. The bending function reads

$$\begin{aligned} (\varphi_{\text{b},i}(\widehat{\mathbf{x}}; \mu_2, \mu_3))_1 &= (r + \widehat{x}_1) \cos(\alpha) - r, \\ (\varphi_{\text{b},i}(\widehat{\mathbf{x}}; \mu_2, \mu_3))_2 &= \widehat{x}_2, \\ (\varphi_{\text{b},i}(\widehat{\mathbf{x}}; \mu_2, \mu_3))_3 &= (r + \widehat{x}_1) \sin(\alpha), \end{aligned} \quad (4.3)$$

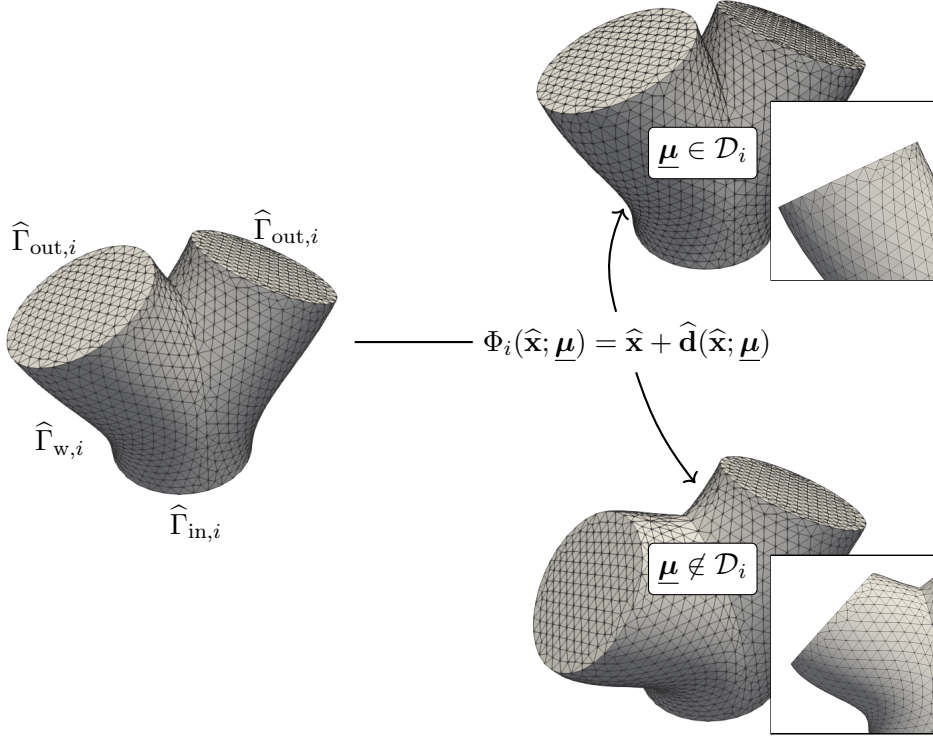


Figure 4.3 – Deformation of the model bifurcation. On the right, we show examples in which the geometrical parameter—i.e., the rotation angles of the outlets—is (top) and is not (bottom) included in the admissible set.

where $r = \hat{L}_i \mu_2 / \mu_3$, $\alpha = \mu_3 x_3 / \hat{L}_i$, and μ_3 is the bending angle. This bending function is designed so that the axis of the deformed tube lays on a circular arc and such that its arclength—which is now $\hat{L}_i \mu_2$, since the scaling function has already been applied—is unmodified. Note that, if $\mu_3 = 0$, we simply take $\varphi_{b,i}(\hat{\mathbf{x}}; \mu_2, \mu_3) = \hat{\mathbf{x}}$.

Defining an analytical transformation for a model bifurcation, as shown in Fig. 4.2, requires to perform the rotation of the outlets to match some prescribed orientation (which clearly depends on the target geometry). However, extending the deformation to the rest of the boundary and to the interior of the domain is a complex objective to achieve with a geometrical transformation in closed form. For any index i corresponding to a model bifurcation, we propose to define a map of the form

$$\Phi_i(\hat{\mathbf{x}}; \underline{\boldsymbol{\mu}}) = \hat{\mathbf{x}} + \hat{\mathbf{d}}(\hat{\mathbf{x}}; \underline{\boldsymbol{\mu}}), \quad (4.4)$$

where $\hat{\mathbf{d}}$ is a displacement field operating the desired rotation of the outlets but also defined inside the domain. In Eq. (4.4), $\underline{\boldsymbol{\mu}}$ is the parameter vector containing the rotation angles of the two outlets.

4.2. Modular Domain Decomposition of arteries

The method through which $\widehat{\mathbf{d}}$ is computed is evidently of importance. Here, we decide to determine the displacement field by defining a linear elasticity problem similar to the one in Eq. (1.22). Specifically, the problem reads

$$\begin{aligned} -\nabla_{\widehat{\mathbf{x}}} \cdot \Pi(\widehat{\mathbf{d}}) &= \mathbf{0} && \text{in } \widehat{\Omega}_i, \\ \widehat{\mathbf{d}} &= \widehat{\mathbf{g}} && \text{on } \widehat{\Gamma}_{D,i}, \\ \Pi(\widehat{\mathbf{d}}_s) \widehat{\mathbf{n}}_s &= \mathbf{0} && \text{on } \widehat{\Gamma}_{N,i}, \end{aligned} \tag{4.5}$$

where Π is the Piola–Kirchoff stress tensor defined as in Eq. (1.23), $\widehat{\Gamma}_{D,i} = \widehat{\Gamma}_{\text{in},i} \cup \widehat{\Gamma}_{\text{out},i}$ is the portion of the boundary of the inlet and the two outlets and $\widehat{\Gamma}_{N,i}$ is the wall of the bifurcation. Function $\widehat{\mathbf{g}}$ is identically zero on $\widehat{\Gamma}_{\text{in},i}$, i.e., $\widehat{\mathbf{g}} = \mathbf{0}$ on $\widehat{\Gamma}_{\text{in},i}$, and it is such that on the outlets $\widehat{\mathbf{d}}$ performs the mapping onto the desired target configurations. As we show in Fig. 4.3, this technique allows us to obtain smooth deformed domains. One drawback is that the range of the rotational angles $\underline{\mu}$ is limited by the fact that, as the geometry becomes more distorted, the quality of the mesh deteriorates (particularly close to the outlets). Moreover, while for small deformations the wall of the bifurcation is almost perpendicular to the outlet in its vicinity—the angle is exactly 90 degrees in the reference configuration—this property is not preserved for larger rotational angles. Ensuring that the angle at the outlets is “close” to perpendicular in the physical configuration is important to avoid large discontinuities in the boundary derivatives when other subdomains are attached to them. The set \mathcal{D}_i of admissible values for $\underline{\mu}$ is chosen such that the mesh quality is always preserved and such that the angles between the bifurcation wall and outlets are always (nearly) 90 degrees. The bounds for the geometrical parameters are chosen with a trial and error approach in which we select the largest possible ranges ensuring that the aforementioned properties are satisfied. During this process, we also make sure that the resulting mesh is topologically valid. This is typically easily verified, since we deal with small deformations.

Another possibility for the deformation of the reference building block is to consider parametrized transfinite maps. These are a general version of the Gordon–Hall transfinite interpolation approach, which is specific to quadrilaterals [GH73]. Parametrized transfinite maps have been proposed in [LMR06b, LMR09] and employed, e.g., in [IQR12, JIR14]. The main idea of these algorithms is to define weight and projection functions on every portion of the boundary. Similarly to our approach, each of these functions are computed through the solution of a PDE—typically, a Laplacian problem—in which ad-hoc conditions are set on every boundary portion. The deformed configuration is found by considering some portions of the boundary (e.g., vertices in two-dimensions) as parametrized, and then by evaluating the displacement over the whole boundary (owing to the weight functions) and subsequently projecting such displacement in the interior of the domain (owing to the projection functions). Although it is possible to extend these concepts to three-dimensions (see, e.g., [Iap12]), in this context, the method becomes considerably more cumbersome, and an approach based on the solution of Eq. (4.5) may

therefore be preferable.

4.2.2 The continuous Navier-Stokes equations on modular geometries

Let us consider the problem of approximating the blood flow in a vessel, mathematically represented by an open and bounded domain $\Omega \in \mathbb{R}^d$. In the following, we take $d = 3$, but the discussion is also valid for the case $d = 2$. We model the blood as an incompressible Newtonian fluid and, therefore, its dynamics is described by the Navier–Stokes equations

$$\begin{aligned}
 \rho_f \frac{\partial \mathbf{u}}{\partial t} + \rho_f (\mathbf{u} \cdot \nabla) \mathbf{u} - \nabla \cdot \sigma_f &= \mathbf{f} && \text{in } \Omega \times (0, T), \\
 \nabla \cdot \mathbf{u} &= 0 && \text{in } \Omega \times (0, T), \\
 \mathbf{u} &= \mathbf{g} && \text{on } \Gamma_D \times (0, T), \\
 \sigma_f(\mathbf{u}, p) \mathbf{n} &= \mathbf{h} && \text{on } \Gamma_N \times (0, T), \\
 \mathbf{u} &= \mathbf{u}_0 && \text{for } t = 0,
 \end{aligned} \tag{4.6}$$

where the symbols are the same as in Eq. 1.1. Since we deal with cardiovascular applications, we take $\Gamma_D = \Gamma_w \cup \left(\bigcup_{i=1}^{N_{\text{in}}} \Gamma_{\text{in}}^{[i]} \right)$ and $\Gamma_N = \bigcup_{i=1}^{N_{\text{out}}} \Gamma_{\text{out}}^{[i]}$; $\Gamma_{\text{in}}^{[1]}, \dots, \Gamma_{\text{in}}^{[N_{\text{in}}]}$, Γ_w and $\Gamma_N^{[1]}, \dots, \Gamma_{\text{out}}^{[N_{\text{out}}]}$ are the inlets, wall and outlets of the vessel, respectively. The inlet velocity profiles and outlet Neumann data are denoted $\mathbf{g}_1, \dots, \mathbf{g}_{N_{\text{in}}}$ and $\mathbf{h}_1, \dots, \mathbf{h}_{N_{\text{out}}}$; on the wall Γ_w we consider $\mathbf{u} = \mathbf{0}$. The physical parameters of the fluid ρ_f and μ_f are fixed for simplicity, although it is possible to include them in the set of the (geometrical) parameters describing the problem.

The weak formulation of problem (4.6) has been derived in Section 1.2. In this chapter, we aim at rewriting W1.1 to account for the modular decomposition of the original domain Ω^m . For each subdomain Ω_j , we introduce the spaces $\mathcal{V}_{g,j} := [H_{g,\Gamma_D}^1(\Omega_j)]^d$ (i.e., the space of function satisfying the Dirichlet boundary conditions), $\mathcal{Q}_j := L^2(\Omega_j)$ and $\mathcal{V}_{0,j} := [H_{\Gamma_D}^1(\Omega_j)]^d$. Moreover, for every interface $\Gamma^{[jm]}$ we define the spaces $\mathcal{L}^{[jm]} = [H_{00}^{-1/2}(\Gamma^{[jm]})]^d$. The coupling Lagrange multipliers belong to these spaces, as discussed in Chapter 2. For the sake of conciseness, let us adopt the following notation

$$\begin{aligned}
 \mathcal{M}_j(\boldsymbol{\varepsilon}, \boldsymbol{\varphi}, \psi; \boldsymbol{\omega}) &:= \int_{\Omega_j} \rho_f \boldsymbol{\varepsilon} \cdot \boldsymbol{\omega} + \int_{\Omega_j} \rho_f [(\boldsymbol{\varphi} \cdot \nabla) \boldsymbol{\varphi}] \cdot \boldsymbol{\omega} + \int_{\Omega_j} \sigma(\boldsymbol{\varphi}, \psi) : \nabla \boldsymbol{\omega} \\
 \mathcal{C}_j(\boldsymbol{\varphi}; \boldsymbol{\eta}) &:= \int_{\Omega_j} \nabla \cdot \boldsymbol{\varphi} \boldsymbol{\eta}, \\
 \mathcal{F}_j(\boldsymbol{\omega}) &:= \int_{\Omega_j} \mathbf{f} \cdot \boldsymbol{\omega} + \sum_{i \in I_{\text{out}}(j)} \int_{\Gamma_{\text{out}}^{[i]}} \mathbf{h} \cdot \boldsymbol{\omega}
 \end{aligned}$$

for every $\boldsymbol{\varepsilon} \in [L^2(\Omega_j)]^d$, $\boldsymbol{\varphi}, \boldsymbol{\omega} \in [H^1(\Omega_j)]^d$ and for every $\psi, \boldsymbol{\eta} \in L^2(\Omega_j)$. Assuming for simplicity that $\Omega = \Omega^m$, it can be shown that W1.1 is equivalent—in a sense that will be specified in Remark 4.3—to the following weak formulation:

(W4.1) given \mathbf{f} , \mathbf{g} , \mathbf{h} , \mathbf{u}_0 regular enough and for every $j = 1, \dots, N_\Omega$, find $\mathbf{u}_j \in \mathcal{V}_{g,j}$, $p_j \in \mathcal{Q}_j$ and $\{\boldsymbol{\lambda}^{[jm]}\}_{m \in N(j)} \in \prod_{m \in N(j)} \mathcal{L}^{[jm]}$, such that, for every $t \in (0, T)$,

$$\begin{aligned} \mathcal{M}_j \left(\frac{\partial \mathbf{u}_j}{\partial t}, \mathbf{u}_j, p_j; \mathbf{v} \right) + \sum_{m \in N(j)} \int_{\Gamma^{[jm]}} \boldsymbol{\lambda}^{[jm]} \cdot \mathbf{v} &= \mathcal{F}_j(\mathbf{v}) \quad \forall \mathbf{v} \in \mathcal{V}_{0,j}, \\ \mathcal{C}_j(\mathbf{u}_j; q) &= 0 \quad \forall q \in \mathcal{Q}_j, \end{aligned} \quad (4.7)$$

and such that $\mathbf{u}_j = \mathbf{u}_0|_{\Omega_j}$ for $t = 0$. Moreover, for every $m \in N(j)$, $\boldsymbol{\lambda}^{[jm]} = -\boldsymbol{\lambda}^{[mj]}$ and

$$\int_{\Gamma^{[jm]}} \boldsymbol{\eta} \cdot (\mathbf{u}_j - \mathbf{u}_m) = 0 \quad \forall \boldsymbol{\eta} \in \mathcal{L}^{[jm]}. \quad (4.8)$$

Remark 4.2. The notation in W4.1 is simplified for the sake of clarity. Indeed, the space in which \mathbf{u}_j is sought for would need to satisfy requirements similar to the ones imposed in W1.1 for the duality of its time derivative to be well defined.

Remark 4.3. The two weak formulations W1.1 and W4.1 are equivalent in the following sense. If (\mathbf{u}, p) satisfies W1.1, then $(\mathbf{u}|_{\Omega_j}, p|_{\Omega_j}, \{\boldsymbol{\sigma}(\mathbf{u}, p)\mathbf{n}_{jm}\}_{m \in N(j)})$ is also solution of the local problem W4.1, for every $j = 1, \dots, N_\Omega$. The Lagrange multipliers therefore play the role of the stress at the interfaces; for details, see, e.g., [Woh00, DIP19]. Conversely, if $(\mathbf{u}_j, p_j, \{\boldsymbol{\lambda}^{[jm]}\}_{m \in N(j)})$ are the local solutions of W4.1, then $(\mathbf{u}, p) = (\prod_{j=1}^{N_\Omega} \mathbf{u}_j, \prod_{j=1}^{N_\Omega} p_j)$ is solution of W1.1.

4.2.3 Discretization of the primal hybrid formulation of the flow problem

The discretization of differential problems by the FE method requires the definition of a computational mesh. In the case of the approximated modular geometry Ω^m , each building block $\widehat{\Omega}_i$ is equipped with its own triangulation $\widehat{\mathcal{T}}_i^h$. Therefore, the global mesh

$$\mathcal{T}^{h,m}(\mathcal{M}) = \bigcup_{j=1}^{N_\Omega} \mathcal{T}_j^h(\underline{\boldsymbol{\mu}}_j) = \bigcup_{j=1}^{N_\Omega} \Phi_{z_j}(\widehat{\mathcal{T}}_{z_j}^h; \underline{\boldsymbol{\mu}}_j) \quad (4.9)$$

is a composition of distinct meshes which do not necessarily satisfy conformity constraints at the interfaces. We recall that a mesh is conforming if, for every two elements, their intersection is either empty, a vertex, an edge or a whole face. In other words, a conforming mesh does not feature any *hanging nodes*.

Remark 4.4. Formula (4.9) is in fact imprecise. In the general case in which Φ_{z_j} is nonaffine, indeed, it holds that $\Phi_{z_j}(\widehat{K}) \neq K \in \mathcal{T}_j^h$ for every $\widehat{K} \in \widehat{\mathcal{T}}_{z_j}^h$, as nonaffine transformations do not map planes into planes. From the practical perspective, the meshes in the deformed configurations \mathcal{T}_j^h are obtained by applying transformation Φ_{z_j} to every vertex of $\widehat{\mathcal{T}}_j^h$; these become the vertices of every element $K \in \mathcal{T}_j^h$. In our case,

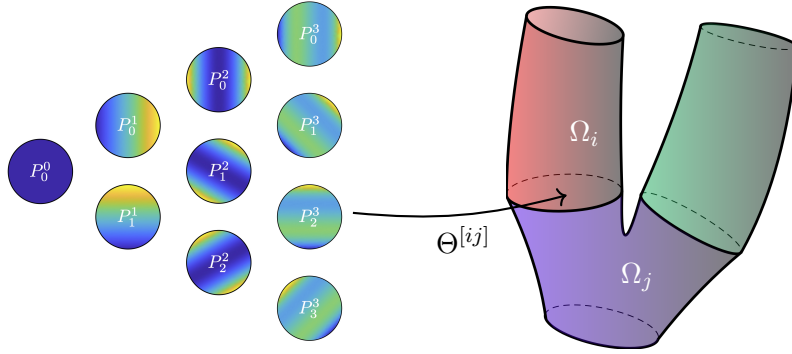


Figure 4.4 – Basis functions P_k^n on the unit disk \mathcal{D} , for $n \leq 3$, mapped onto the target interface $\Gamma^{[ij]}$. Minimum and maximum values are plotted in blue and yellow, respectively. The basis function P_0^0 is constant.

every K is a tetrahedron. We finally observe that if P1 Lagrangian finite elements are generated on the mesh, the nodes are located on the vertices of each element K and not in points belonging to either faces or the interior of K . Hence, in this particular case, it is equivalent to first generate the nodes on the reference configuration and then map each of them onto the physical ones, and, conversely, map every vertex first and then generate the nodes on the mapped set of vertices.

Being the global mesh in general nonconforming, we are compelled to consider nonconforming Domain Decomposition methods for the solution of the Navier–Stokes equations. These are formally defined as Domain Decomposition methods in which the search space for the discrete solution is not a subset of the continuous search space (in our case $\mathcal{V} \times \mathcal{Q}$). In this thesis, we adopt the algorithm presented in Chapter 2, which is based on the discretization of the Lagrange multipliers space via a small number of spectral basis functions defined on the interfaces. For our application, this choice is convenient because (i) the method allows recovering the h -convergence order of the primal discretization method—i.e., the FE method—even when a small number of spectral basis functions is considered, (ii) defining a spectral basis on each interface does not require to project nor to interpolate the traces of FE basis functions from one side to the other, and (iii) as already mentioned, the interfaces are circular in the target configuration, which allows us to employ a set of standard orthonormal basis functions on the two-dimensional disk.

We follow the same procedure presented in Section 1.2 for the discretization of local variables \mathbf{u}_j and p_j , which become

$$\mathbf{u}_j^h = \sum_{i=1}^{N_{u,z_j}^h} u_{i,j}^h \boldsymbol{\varphi}_{i,j}^h, \quad p_j^h = \sum_{i=1}^{N_{p,z_j}^h} p_{i,j}^h \psi_{i,j}^h.$$

We highlight that—being that the meshes \mathcal{T}_j^h are generated as transformations of the

4.2. Modular Domain Decomposition of arteries

reference ones $\widehat{\mathcal{T}}_{z_j}^h$ (at least approximately, see Remark 4.4)—the number of nodes N_{u,z_j}^h and N_{p,z_j}^h are indexed with respect to the reference building blocks.

As for the Lagrange multiplier $\boldsymbol{\lambda}^{[jm]}$, we consider the approximation

$$\boldsymbol{\lambda}^{[jm],\delta} = \sum_{i=1}^{N_\lambda^\delta} \lambda_i^{[jm],\delta} \boldsymbol{\xi}_i^{[jm],\delta},$$

where $\boldsymbol{\xi}_i^{[jm],\delta} = c_{jm} \widehat{\boldsymbol{\xi}}_i^\delta \circ (\Theta^{[jm]})^{-1} \in [L^2(\Gamma^{[jm]})]^d$, $c_{jm} \in \mathbb{R}$, $\widehat{\boldsymbol{\xi}}_i^\delta \in [L^2(\mathcal{D})]^d$ are a set of orthogonal functions on the unit disk $\mathcal{D} \in \mathbb{R}^{d-1}$ and $\Theta^{[jm]} : \mathcal{D} \mapsto \Gamma^{[jm]}$ is a bijective map from the unit disk \mathcal{D} to the target interface. Similarly, we define the maps $\Theta_{\text{in}}^{[m]} : \mathcal{D} \mapsto \Gamma_{\text{in}}^{[m]}$ from the unit disk \mathcal{D} to the inlet interfaces $\Gamma_{\text{in}}^{[m]}$, the basis functions $\boldsymbol{\xi}_{i,\text{in}}^{[m],\delta} = \widehat{\boldsymbol{\xi}}_i^\delta \circ (\Theta_{\text{in}}^{[m]})^{-1} \in [L^2(\Gamma_{\text{in}}^{[m]})]^d$, and

$$\boldsymbol{\lambda}_{\text{in}}^{[m],\delta} = \sum_{i=1}^{N_\lambda^\delta} \lambda_{i,\text{in}}^{[m],\delta} \boldsymbol{\xi}_{i,\text{in}}^{[m],\delta}.$$

These functions are used to set the inlet velocity profiles, allowing to effortlessly transition from the FE model to the reduced one through the process described in Section 4.4. The use of Lagrange multipliers is a classical way to weakly impose Dirichlet boundary conditions (see, e.g., [Bab73a]) and, compared to other popular approaches such as penalty methods [ZA98, Bab73b], it has the advantage of being variationally consistent. We remark that (i) we introduce the discretization parameter δ for the Lagrange multipliers to indicate that the degree of refinement is in fact independent of the mesh size in \mathcal{T}_j^h or \mathcal{T}_m^h , and (ii) we consider for simplicity the same number N_λ^δ of basis functions at each interface.

In this work, we construct $\{\widehat{\boldsymbol{\xi}}_i^\delta\}_{i=1}^{N_\lambda^\delta}$ as follows. Let us consider Chebyshev polynomials of the second kind U_n , which are defined through the recurrence relation $U_0(x) = 1$, $U_1(x) = 2x$, $U_{n+1}(x) = 2xU_n(x) - U_{n-1}(x)$. Then, for $0 \leq k \leq n$,

$$P_k^n(x, y) = \frac{1}{\sqrt{\pi}} U_n(x \cos(\omega x) + y \sin(\omega y)), \quad \omega = \frac{k}{n+1} \pi,$$

are orthonormal polynomials on the unit disk \mathcal{D} with respect to the weight function $W(x, y) = 1/\sqrt{\pi}$ [DX14]. Given $n \geq 0$, we set

$$\widehat{\boldsymbol{\Xi}}_n := \{\widehat{\boldsymbol{\xi}}_i^\delta\}_{i=1}^{N_\lambda^\delta} = \bigcup_{i=1}^d \bigcup_{\tilde{n}=0}^n \bigcup_{k=0}^{\tilde{n}} P_k^{\tilde{n}} \mathbf{e}_i,$$

where \mathbf{e}_i is the i^{th} canonic vector. It is trivial to find that $N_\lambda^\delta = d(n+1)(n+2)/2$.

Let us now address the discretization of the individual elements of W4.1, which is local

to subdomain Ω_j , the assembly of the global block system, and the discretization in time.

Discretization of the coupled momentum and continuity equations. For every $j = 1, \dots, N_\Omega$ and $m \in N(j)$, let us define $\underline{\lambda}^{[jm],\delta} = [\lambda_1^{[jm],\delta}, \dots, \lambda_{N_\lambda^\delta}^{[jm],\delta}]$ and the coupling matrix $B_{pq}^{[jm],h\delta} = \int_{\Gamma^{[jm]}} \xi_p^{[jm],\delta} \cdot \varphi_{q,j}^h$. Then, Eq. (4.7) can be rewritten in algebraic form as

$$\begin{bmatrix} M_j^h \\ \end{bmatrix} + \begin{bmatrix} \dot{\underline{\mathbf{u}}}_j^h \\ \dot{\underline{\mathbf{p}}}_j^h \\ \dot{\underline{\lambda}}_j^\delta \end{bmatrix} + \begin{bmatrix} K_j^h + C_j^h(\underline{\mathbf{u}}_j^h) & (D_j^h)^T & (B_j^{h\delta})^T \\ & D_j^h & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{u}}_j^h \\ \underline{\mathbf{p}}_j^h \\ \underline{\lambda}_j^\delta \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{f}}_j^h \\ \underline{\mathbf{0}} \end{bmatrix}. \quad (4.10)$$

In Eq. (4.10), we denoted by $\underline{\lambda}^{j,\delta}$ the vector containing all $\underline{\lambda}^{[jm],\delta}$ and by $B_j^{h\delta}$ the matrix obtained by stacking the various $B^{[jm],h\delta}$, for every $m \in N(j)$.

Equality of the Lagrange multipliers at the same interface. A natural way to enforce constraint $\underline{\lambda}^{[jm]} = -\underline{\lambda}^{[mj]}$ in W4.1 is to choose $c_{jm} = 1 = -c_{mj}$ such that $\xi_i^{[jm],\delta} = -\xi_i^{[mj],\delta}$ and $\lambda_i^{[jm],\delta} = \lambda_i^{[mj],\delta}$ for all $m \in N(j)$ and for $i = 1, \dots, N_\lambda^\delta$. As a consequence, it is legitimate to introduce a numbering of the interfaces $\Gamma^{[1]}, \dots, \Gamma^{[N_\Gamma]}$ and to denote the corresponding (unique) vectors of dofs of the Lagrange multipliers by $\underline{\lambda}^{[1],\delta}, \dots, \underline{\lambda}^{[N_\Gamma],\delta}$. Furthermore, it holds that $B_{pq}^{[mj],h\delta} = -\int_{\Gamma^{[mj]}} \xi_p^{[mj],\delta} \cdot \varphi_{q,m}^h$. In the following, the coupling matrix between the i^{th} interface and the velocity in the j^{th} subdomain is also written $B_j^{[i],h\delta}$ (to be intended as null matrix, if $\Gamma^{[i]} \cap \overline{\Omega}_j = \emptyset$).

Discretization of the weak continuity equation. The numerical discretization of Eq. (4.8) entails the definition of a preferential side for $\Gamma^{[jm]}$ which determines the computational mesh and the FE basis functions to be used in the evaluation of the integral. However, the direct approximation of Eq. (4.8) is problematic because it requires the projection or interpolation of the velocity from one side of the interface to the other (such operation is required in methods such as the mortar method or INTERNODES, as discussed in Section 2.4). At the continuous level, it is evidently possible to write

$$\int_{\Gamma^{[jm]}} \boldsymbol{\eta} \cdot (\mathbf{u}_j - \mathbf{u}_m) = \int_{\Gamma^{[jm]}} \boldsymbol{\eta} \cdot \mathbf{u}_j - \int_{\Gamma^{[mj]}} \boldsymbol{\eta} \cdot \mathbf{u}_m = 0. \quad (4.11)$$

The advantage of separating the integral into the two contributions on $\Gamma^{[jm]}$ and $\Gamma^{[mj]}$ is apparent at the discrete level. Indeed, by substituting $\boldsymbol{\eta}$ with $\xi_i^{[jm],\delta}$ and $-\xi_i^{[mj],\delta}$ in the two integrals on the right-hand side of Eq. (4.11), Eq. (4.8) is discretized as

$$B^{[jm],h\delta} \underline{\mathbf{u}}_j^h + B^{[mj],h\delta} \underline{\mathbf{u}}_m^h = \underline{\mathbf{0}}.$$

Weak imposition of Dirichlet boundary conditions. We recall that we assume every inlet interface $\Gamma_{\text{in}}^{[m]}$ to be associated with a single subdomain Ω_j . Condition $\mathbf{u}_j^h = \mathbf{g}_m$ on $\Gamma_{\text{in}}^{[m]}$ is weakly imposed as

$$\int_{\Gamma_{\text{in}}^{[m]}} \boldsymbol{\xi}_{i,\text{in}}^{[m],\delta} \cdot (\mathbf{u}_j^h - \mathbf{g}_m) = 0, \quad i = 1, \dots, N_{\lambda}^{\delta}. \quad (4.12)$$

By introducing the interpolation of the boundary data \mathbf{g}_m onto the FE space spanned by $\{\varphi_{i,j}^h\}_{i=1}^{N_{u,z_j}^h}$, the corresponding vector of dofs $\underline{\mathbf{g}}_m^h$, and $(B_{\text{in},j}^{[m],h\delta})_{pq} = \int_{\Gamma_{\text{in}}^{[m]}} \boldsymbol{\xi}_{p,\text{in}}^{[m],\delta} \cdot \varphi_{q,j}^h$, Eq. (4.12) is approximated as

$$B_{\text{in},j}^{[m],h\delta} \underline{\mathbf{u}}_j^h - B_{\text{in},j}^{[m],h\delta} \underline{\mathbf{g}}_m^h = \mathbf{0},$$

and the first equation in Eq. (4.10) is accordingly modified as

$$M_j^h \underline{\dot{\mathbf{u}}}_j^h + \left(K_j^h + C_j^h(\underline{\mathbf{u}}_j^h) \right) \underline{\mathbf{u}}_j^h + (D_j^h)^{\text{T}} \underline{\mathbf{p}}_j^h + (B_j^{h\delta})^{\text{T}} \underline{\boldsymbol{\lambda}}_j^h + (B_{\text{in},j}^{[m],h\delta})^{\text{T}} \boldsymbol{\lambda}_{\text{in}}^{[m],\delta} = \underline{\mathbf{f}}_j^h$$

to account for the Lagrange multiplier $\boldsymbol{\lambda}_{\text{in}}^{[m],\delta}$.

Assembly of global system. It is possible to arrange the local systems corresponding to the subdomains in the form of a global block system as

$$\begin{bmatrix} \mathcal{M}^h \\ \mathbf{A}^{\delta} \end{bmatrix} \begin{bmatrix} \underline{\dot{\mathbf{W}}}^h \\ \underline{\boldsymbol{\Lambda}}^{\delta} \end{bmatrix} + \begin{bmatrix} \mathcal{A}^h(\underline{\mathbf{W}}^h) & (\mathcal{B}^{h\delta})^{\text{T}} \\ \mathcal{B}^{h\delta} & \end{bmatrix} \begin{bmatrix} \underline{\mathbf{W}}^h \\ \underline{\boldsymbol{\Lambda}}^{\delta} \end{bmatrix} = \begin{bmatrix} \underline{\mathbf{F}}^h \\ \underline{\mathbf{G}}^{h\delta} \end{bmatrix},$$

where

$$\begin{aligned} \mathcal{M}^h &:= \text{diag} \left(\begin{bmatrix} M_j^h \\ \end{bmatrix} \right)_{j=1, \dots, N_{\Omega}}, \\ \mathcal{A}^h(\underline{\mathbf{W}}^h) &:= \text{diag} \left(\begin{bmatrix} K_j^h + C_j^h(\underline{\mathbf{u}}_j^h) & (D_j^h)^{\text{T}} \\ D_j^h & \end{bmatrix} \right)_{j=1, \dots, N_{\Omega}}, \end{aligned} \quad (4.13)$$

$\mathcal{B}^{h\delta}$ is a block matrix such that $(\mathcal{B}^{h\delta})_{ij} = [B_j^{[i],h\delta}, O]$ (O being the null matrix) if $i \leq N_{\Gamma}$ and $(\mathcal{B}^{h\delta})_{ij} = [B_{\text{in},j}^{[i-N_{\Gamma}],h\delta}, O]$ otherwise,

$$\begin{aligned} \underline{\mathbf{W}}^h &= \text{vec}(\mathbf{w}_1^h, \dots, \mathbf{w}_{N_{\Omega}}^h), \\ \underline{\boldsymbol{\Lambda}}^{\delta} &= \text{vec}(\boldsymbol{\lambda}^{[1],\delta}, \dots, \boldsymbol{\lambda}^{[N_{\Gamma}],\delta}, \boldsymbol{\lambda}_{\text{in}}^{[1],\delta}, \dots, \boldsymbol{\lambda}_{\text{in}}^{[N_{\text{in}}],\delta}), \end{aligned}$$

where the `vec` function outputs the global vector obtained by stacking each input along the first direction, and $\underline{\mathbf{F}}^h$ and $\underline{\mathbf{G}}^{h\delta}$ are block vectors accounting for the forcing terms

and Dirichlet boundary conditions, respectively.

Discretization in time and global nonlinear residual. The discretization in time is performed with BDF schemes along the same lines of the discussion in Section 1.3. We define $\underline{\mathbf{Y}}^{h\delta} = [\underline{\mathbf{W}}^h, \underline{\Lambda}^\delta]$ and

$$\mathcal{H}^h := \begin{bmatrix} \mathcal{M}^h \\ \mathbf{0} \end{bmatrix}, \quad \underline{\mathbf{F}}^{h\delta}(t, \underline{\mathbf{Y}}^{h\delta}) := \begin{bmatrix} \underline{\mathbf{F}}^h(t) \\ \underline{\mathbf{G}}^{h\delta}(t) \end{bmatrix} - \begin{bmatrix} \mathcal{A}^h(\underline{\mathbf{W}}^h) & (\mathcal{B}^{h\delta})^\top \\ \mathcal{B}^{h\delta} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{W}}^h \\ \underline{\Lambda}^\delta \end{bmatrix}.$$

Then, given $\underline{\mathbf{Y}}_{k-j+1}^{h\delta}$ for $j = 1, \dots, \sigma$, the solution at timestep t_{k+1} is found by solving

$$\underline{\mathbf{R}}(\underline{\mathbf{Y}}_{k+1}^{h\delta}) := \mathcal{H}^h \underline{\mathbf{Y}}_{k+1}^{h\delta} - \sum_{j=1}^{\sigma} \alpha_j \mathcal{H}^h \underline{\mathbf{Y}}_{k-j+1}^{h\delta} - \Delta t \beta \underline{\mathbf{F}}^{h\delta}(t_{k+1}, \underline{\mathbf{Y}}_{k+1}^{h\delta}) = \underline{\mathbf{0}}. \quad (4.14)$$

4.3 Efficient solution of the global nonlinear system

Eq. (4.14) is nonlinear and hence solved using the Newton–Raphson algorithm. In particular, given an initial guess $\underline{\mathbf{Y}}^{(0)}$, the $(l+1)$ th iteration of the algorithm for the solution of $\underline{\mathbf{R}}(\underline{\mathbf{Y}}) = \underline{\mathbf{0}}$ is

$$\underline{\mathbf{Y}}^{(l+1)} = \underline{\mathbf{Y}}^{(l)} - \left(\mathcal{J}_R(\underline{\mathbf{Y}}^{(l)}) \right)^{-1} \underline{\mathbf{R}}(\underline{\mathbf{Y}}^{(l)}), \quad (4.15)$$

where \mathcal{J}_R is the tangent matrix of $\underline{\mathbf{R}}$. The stopping criterion is based on a user-provided tolerance τ_{NR} and reads $\|\underline{\mathbf{R}}(\underline{\mathbf{Y}}^{(l)})\|_2 / \|\underline{\mathbf{R}}(\underline{\mathbf{Y}}^{(0)})\|_2 < \tau_{\text{NR}}$.

In order to efficiently solve the linear system in Eq. (4.15) via iterative methods such as GMRES [SS86], we need to develop a preconditioner for the tangent matrix \mathcal{J}_R . Differentiating Eq. (4.14) with respect to its only argument yields

$$\mathcal{J}_R(\underline{\mathbf{Y}}) = \begin{bmatrix} \mathcal{M}^h + \Delta t \beta \mathcal{A}^h(\underline{\mathbf{W}}) & \Delta t \beta (\mathcal{B}^{h\delta})^\top \\ \Delta t \beta \mathcal{B}^{h\delta} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \tilde{\mathcal{A}}(\underline{\mathbf{W}}) & \tilde{\mathcal{B}}^\top \\ \tilde{\mathcal{B}} & \mathbf{0} \end{bmatrix}. \quad (4.16)$$

Thus, the tangent matrix features a saddle-point structure stemming directly from the original differential problem W4.1—which is in fact a saddle-point problem. In the remainder of this section, we omit the explicit dependence of \mathcal{J}_R and $\tilde{\mathcal{A}}$ on $\underline{\mathbf{Y}}$ and $\underline{\mathbf{W}}$, respectively, for the sake of clarity of notation. A possible strategy to design a preconditioner is based on the (exact) decomposition

$$\begin{bmatrix} \tilde{\mathcal{A}} & \tilde{\mathcal{B}}^\top \\ \tilde{\mathcal{B}} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \mathcal{I} & \\ \tilde{\mathcal{B}} \tilde{\mathcal{A}}^{-1} & \mathcal{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathcal{A}} & \\ & \mathcal{S} \end{bmatrix} \begin{bmatrix} \mathcal{I} & \tilde{\mathcal{A}}^{-1} \tilde{\mathcal{B}}^\top \\ & \mathcal{I} \end{bmatrix}, \quad (4.17)$$

where $\mathcal{S} = -\tilde{\mathcal{B}} \tilde{\mathcal{A}}^{-1} \tilde{\mathcal{B}}^\top$ is the Schur complement of \mathcal{J}_R . This decomposition is the

4.3. Efficient solution of the global nonlinear system

Algorithm 4.1 Application of the saddle-point block preconditioner.

Input: Matrix \mathcal{J}_R defined as in Eq. (4.16), right-hand side $\mathbf{B} = [\mathbf{B}_w, \mathbf{B}_\lambda]$

Output: Solution $\mathbf{X} = [\mathbf{X}_w, \mathbf{X}_\lambda]$ such that $\mathcal{J}_R \mathbf{X} = \mathbf{B}$

- 1: assemble the Schur complement $\mathcal{S} = -\tilde{\mathcal{B}}\tilde{\mathcal{A}}^{-1}\tilde{\mathcal{B}}^T$
 - 2: compute $\mathbf{z}_w = \tilde{\mathcal{A}}^{-1}\mathbf{B}_w$
 - 3: find $\mathbf{X}_\lambda = \mathcal{S}^{-1}(\mathbf{B}_\lambda - \tilde{\mathcal{B}}\mathbf{z}_w)$
 - 4: find $\mathbf{X}_w = \mathbf{z}_w - \tilde{\mathcal{A}}^{-1}\tilde{\mathcal{B}}^T\mathbf{X}_\lambda$
-

foundation of several preconditioners for saddle-point systems, such as SIMPLE [SRV10] and the nested block preconditioner for blood flow simulations proposed in [LYDM20]; see also [CLB20] for the derivation of a preconditioner based on Eq. (4.17) for the nonconforming coupling of Kirchoff plates with IGA. The solution of a linear system of the form $\mathcal{J}_R \mathbf{X} = \mathbf{B}$, with $\mathbf{X} = [\mathbf{X}_w, \mathbf{X}_\lambda]$ and $\mathbf{B} = [\mathbf{B}_w, \mathbf{B}_\lambda]$, amounts to solving

$$\begin{bmatrix} \tilde{\mathcal{A}} & \tilde{\mathcal{B}}^T \\ & \mathcal{S} \end{bmatrix} \begin{bmatrix} \mathbf{X}_w \\ \mathbf{X}_\lambda \end{bmatrix} = \begin{bmatrix} \tilde{\mathcal{A}} & \\ & \mathcal{S} \end{bmatrix} \begin{bmatrix} \mathcal{I} & \tilde{\mathcal{A}}^{-1}\tilde{\mathcal{B}}^T \\ & \mathcal{I} \end{bmatrix} \begin{bmatrix} \mathbf{X}_w \\ \mathbf{X}_\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{B}_w & \\ \mathbf{B}_\lambda - \tilde{\mathcal{B}}\tilde{\mathcal{A}}^{-1}\mathbf{B}_w \end{bmatrix}, \quad (4.18)$$

hence $\mathbf{X}_\lambda = \mathcal{S}^{-1}(\mathbf{B}_\lambda - \tilde{\mathcal{B}}\mathbf{z}_w)$ and $\mathbf{X}_w = \mathbf{z}_w - \tilde{\mathcal{A}}^{-1}\tilde{\mathcal{B}}^T\mathbf{X}_\lambda$, where $\mathbf{z}_w = \tilde{\mathcal{A}}^{-1}\mathbf{B}_w$ can be computed only once for efficiency. The procedure for the application of the preconditioner is summarized in Algorithm 4.1.

4.3.1 Bottlenecks of the global preconditioner

The bottlenecks of Algorithm 4.1 are evidently the computation of the Schur complement \mathcal{S} and the inversion of matrices $\tilde{\mathcal{A}}$ and \mathcal{S} .

Block (i, j) of the Schur complement explicitly takes the form

$$(\mathcal{S})_{ij} = - \sum_{l=1}^{N_\Omega} \sum_{m=1}^{N_\Omega} (\tilde{\mathcal{B}})_{il} (\tilde{\mathcal{A}}^{-1})_{lm} (\tilde{\mathcal{B}}^T)_{mj} = - \sum_{k=1}^{N_\Omega} (\tilde{\mathcal{B}})_{ik} (\tilde{\mathcal{A}}^{-1})_{kk} (\tilde{\mathcal{B}}^T)_{kj}, \quad (4.19)$$

where

$$(\tilde{\mathcal{A}}^{-1})_{kk} = \begin{bmatrix} M_k^h + \Delta t \beta \left(K_k^h + C_k^h(\mathbf{u}_k^h) \right) & (D_k^h)^T \\ & D_k^h \end{bmatrix}^{-1},$$

$$(\tilde{\mathcal{B}})_{ik} = \begin{bmatrix} B_k^{[i],h\delta} & O \end{bmatrix}.$$

The second equality in Eq. (4.19) comes from the fact that $\tilde{\mathcal{A}}$ is block diagonal. Each block of the Schur complement is therefore a sum of the contributions stemming from each subdomain Ω_j . However, we recall that $B_j^{[i],h\delta}$ is not null only if $\Gamma^{[i]}$ is an interface of Ω_j , which typically implies that only a small number of subdomains contribute to each $(\mathcal{S})_{ij}$.

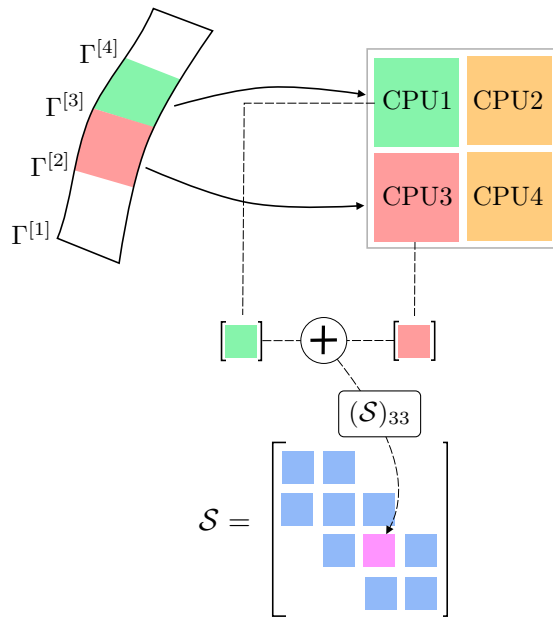


Figure 4.5 – Parallel computation of a single block of the Schur complement \mathcal{S} .

Additionally, as discussed in Chapter 2 and Section 4.2.3, the Lagrange multiplier basis functions required to achieve h -convergence are often very few; hence, the Navier–Stokes matrix inverse $(\mathcal{A}^{-1})_{kk}$ (or an approximation thereof) can be efficiently applied to every column of $(\mathcal{B}^T)_{kj}$, whose number is typically small.

Remark 4.5. The special structure of the Schur complement (4.19) makes its computation particularly attractive in the context of high-performance computing, assuming that each subdomain is assigned to one or few processors. An effective partition strategy should take into account the number of dofs associated with each geometrical building block in order to preserve the load balancing among the set of computational nodes. At the same time, the fact that the coupling conditions do not require to explicitly transfer information from one side of the interface to the other can be exploited to limit the amount of processor communications. For example, Fig. 4.5 depicts a case in which two subdomains sharing a common interface are mapped onto separate cores. Each contribution to the Schur complement \mathcal{S} is computed in parallel and communication is only required when the two local matrices need to be added. In this thesis, we only consider a serial implementation, but future directions of the current work include investigations on the parallel performance of the preconditioner.

Remark 4.6. The computation of the Schur complement \mathcal{S} , whose cost scales linearly with the number of columns of $\tilde{\mathcal{B}}$ (i.e., the total number of Lagrange multiplier basis functions), is, in fact, the most expensive step of the application of the preconditioner discussed in this section. In our numerical experiments, however, we observed that significant performance improvements are achieved by reusing the same Schur complement for

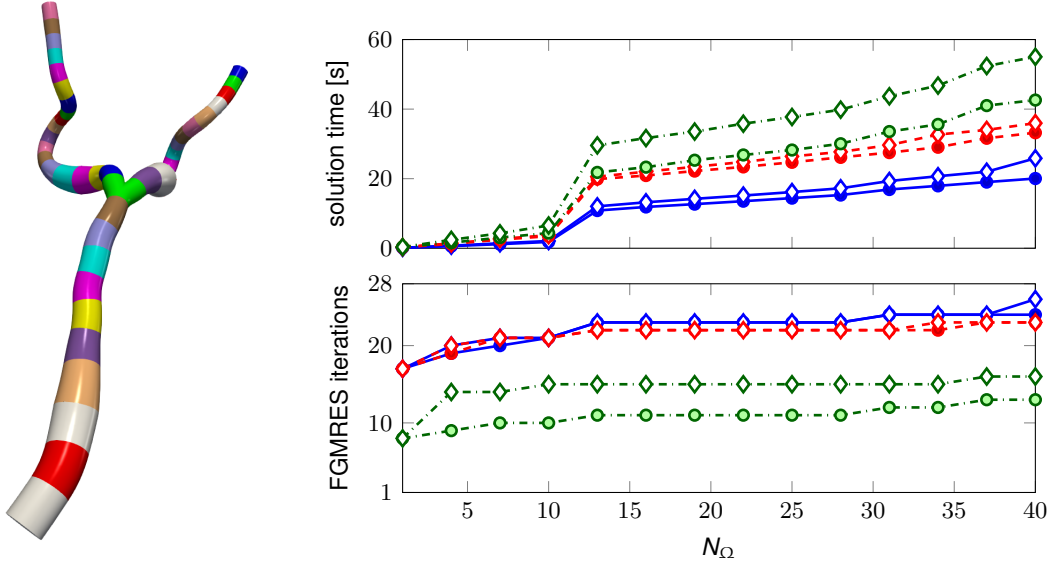


Figure 4.6 – Solution time of a single linear system preconditioned as discussed in Section 4.3 (right-top) and corresponding number of FGMRES iterations (right-bottom), as functions of number of subdomains N_Ω in the geometry of the aorta and iliac arteries (left). Blue solid lines: $\tilde{\mathcal{A}}^{-1}$ approximated with a single application of SIMPLE for each subdomain; red dashed lines and green dash-dotted lines: $\tilde{\mathcal{A}}^{-1}$ solved with GMRES and tolerance $5 \cdot 10^{-1}$ and 10^{-2} , respectively; circles: $N_\lambda^\delta = 3$; diamonds = $N_\lambda^\delta = 84$.

a number of n consecutive applications of the preconditioner. This strategy does not significantly affect the number of GMRES iterations when n is small enough (in our case, n is roughly 20).

Let us now address the application of the inverses of $\tilde{\mathcal{A}}$ and \mathcal{S} . Since $\mathcal{S} \in \mathbb{R}^{N_\Gamma N_\lambda^\delta \times N_\Gamma N_\lambda^\delta}$, the Schur complement is typically small and is inexpensively inverted by solving the linear system either directly or via iterative methods (complemented with standard preconditioners such as multigrid [KN03, BPX90] or ILU [Saa03]). Matrix $\tilde{\mathcal{A}}$ features a block diagonal structure in which each diagonal block is itself a saddle-point system and inverting $\tilde{\mathcal{A}}$ is therefore equivalent to solving linear systems which are local to each subdomain Ω_j . Employing, instead of $\tilde{\mathcal{A}}^{-1}$, a suitable approximation thereof, gives rise to different suitable preconditioners.

In this work, we choose to approximate every block diagonal matrix of $\tilde{\mathcal{A}}^{-1}$ by considering a single application of the SIMPLE preconditioner, both in the computation of the Schur complement shown in Eq. (4.19) and in the application of the preconditioner. We recall that SIMPLE is also based on Eq. (4.17) applied to the Navier–Stokes equations: here, matrix $\tilde{\mathcal{A}}$ —in this case the Navier–Stokes matrix—is approximated with its diagonal and this results in a fast computation of its inverse [SRV10]. In Fig. 4.6, we show the robustness of our preconditioner with respect to the number of blocks and to the number

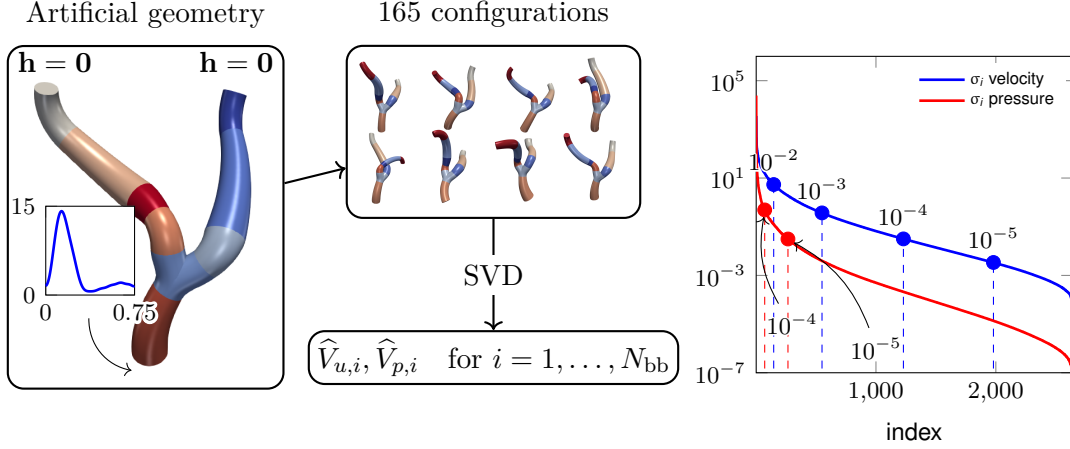


Figure 4.7 – Offline phase on an artificial geometry featuring $N_\Omega = 9$ subdomains and $N_{\text{bb}} = 4$ building blocks (left) and singular values decay for the velocity and pressure reduced bases built on the bifurcation (right). The colored dots on the plot on the right refer to different values of POD tolerances ε_u and ε_p .

of basis functions for the Lagrange multipliers per interface N_λ^δ . The considered geometry is that of the aorta and the iliac arteries in Fig. 4.6 (left). The blocks are sequentially added starting from the inlet (for this reason, we remark that the size of the system increases proportionally with the number of blocks). We compare the preconditioner performance with that achieved by inverting every block in $\tilde{\mathcal{A}}$ with GMRES and relatively large tolerances ($5 \cdot 10^{-1}$ and 10^{-2}). We highlight that, as the preconditioner in the latter approach varies at each iteration, we are compelled to employ flexible GMRES (FGMRES) [Saa93]. If the local systems are solved exactly, the preconditioner is, in fact, the original global matrix, as Eq.(4.17) is an exact decomposition. For this reason, solving the local linear systems with GMRES leads to a better performance in terms of the number of iterations. However, approximating each local inverse with SIMPLE is more efficient in terms of solution time, as each FGMRES iteration is less computationally expensive. We conclude by observing that the increase in solution time occurring at $N_\Omega = 13$ is due to the introduction of the bifurcation—which is composed of a larger number of elements and which also requires more dofs for the coupling (due to the higher number of interfaces) than the other blocks—in the set of considered subdomains.

4.4 The Reduced Basis Element method for flow in arteries

As discussed in Section 4.2, the subdomains in the target geometry Ω^m are obtained from the parametrized geometrical deformation of a number of building blocks $\hat{\Omega}_i$, $i = 1, \dots, N_{\text{bb}}$. In this chapter, these are a model symmetric bifurcation (B), and straight tubes with aspect ratios length/diameter 1:1 (T1), 1:2 (T2) and 1:3 (T3). The

4.4. The Reduced Basis Element method for flow in arteries

Table 4.1 – Velocity RB sizes $N_{u,i}$ with respect to different POD tolerances ε_u , for 4 different building blocks (B: bifurcation, T1, T2 and T3: tubes with aspect ratios diameter/length 1:1, 1:2, 1:3, respectively).

	ε_u					
	$1.6 \cdot 10^{-2}$	$8 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	10^{-3}	$5 \cdot 10^{-4}$
B	99	172	270	394	549	732
T1	69	134	233	379	582	848
T2	60	103	162	243	354	503
T3	23	45	78	131	211	324

offline phase of our ROM defines RB functions in each of these building blocks $\widehat{\Omega}_i$. The snapshots are collected from a single decomposed “artificial” geometry $\Omega^m = \bigcup_{j=1}^{N_\Omega} \Omega_j$ by sampling the geometrical parameters $\underline{\mu}_1, \dots, \underline{\mu}_{N_\Omega}$ describing each subdomain from uniform distributions centered on the values characterizing the original configuration, as depicted in Fig. 4.7. The snapshots are found by solving a flow problem with $\rho_f = 1.06 \text{ gr cm}^{-3}$, $\mu_f = 0.04 \text{ gr cm}^{-1} \text{ s}^{-1}$, the imposed inflow flow rate Q shown in Fig. 4.7 (in the box on the left; y -axis: Q [cc/s], x -axis: t [s]) with a parabolic profile and homogeneous Neumann conditions $\mathbf{h} = \mathbf{0}$ on the outlets, on 165 random configurations of the artificial geometry. There exist other equally valid possibilities to generate the database of snapshots. For example, these could be taken by solving flow problems on a collection of target geometries. This approach allows us to avoid issues related to the random sampling of the geometrical parameters—e.g., physiological feasibility of the resulting global geometry—but requires the aid of an automatic algorithm for the decomposition to be efficient. The development of such an algorithm is one of the possible future extensions of the present work. The simulations are run from $t_0 = 0 \text{ s}$ to $T = 0.3 \text{ s}$ with a BDF scheme of order $\sigma = 2$ and $\Delta t = 2.5 \cdot 10^{-3} \text{ s}$. The initial condition at t_0 is computed by gradually increasing the inflow flow rate profile at the inlet by the law $Q(t) = Q_0[1 - \cos((t - t_0^{\text{ramp}})\pi/(t_0 - t_0^{\text{ramp}}))]/2$, Q_0 being the desired flow rate at time t_0 , from $t = t_0^{\text{ramp}} = -2 \cdot 10^{-2} \text{ s}$ to $t = t_0$. For the discretization of the Lagrange multipliers on each interface, we employ the set of basis functions $\widehat{\Xi}_n$ with $n = 5$, which corresponds to $N_\lambda^\delta = 63$ basis functions. We remark that the *Artificial geometry* in Fig. 4.7 is not included in the configurations used for the snapshots generation and is considered in Section 5.2 to assess the performance of the method.

The $N_{s,i}$ snapshots for velocity and pressure for the i^{th} building block are collected in matrices $\widehat{S}_{u,i} \in \mathbb{R}^{N_{u,i}^h \times N_{s,i}}$ and $\widehat{S}_{p,i} \in \mathbb{R}^{N_{p,i}^h \times N_{s,i}}$. We remark that, since we are dealing with unsteady problems, these matrices include snapshots sampled at different timesteps for different values of the geometrical parameters. It is worth noting that each velocity snapshot, which is divergence free in the deformed configuration, does not retain such property on the reference building block. In order to consider snapshots which are divergence-free in the reference configuration, the columns of $\widehat{S}_{u,i}$ are scaled by means

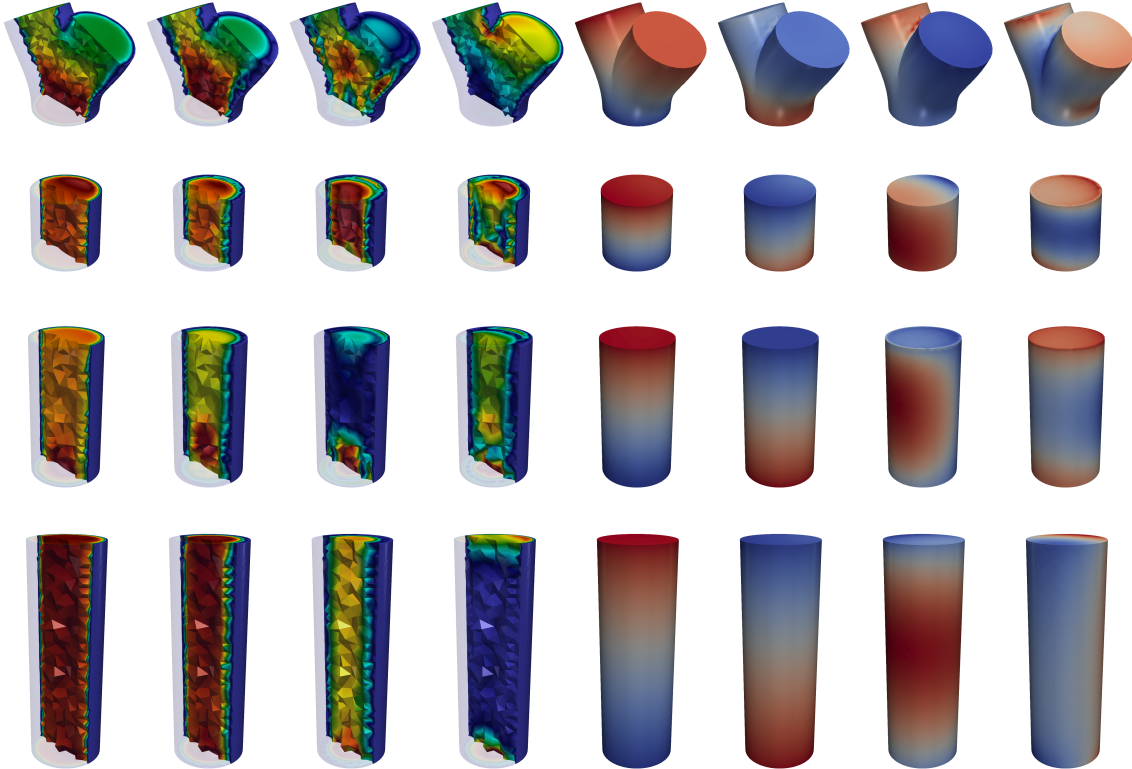


Figure 4.8 – First four modes of velocity (left) and pressure (right) for the four building blocks. From top to bottom: bifurcation (B), tubes with aspect ratio length/diameter 1:1 (T1), 1:2 (T2) and 1:3 (T3).

Table 4.2 – Pressure RB sizes $N_{p,i}$ with respect to different POD tolerances ε_p , for 4 different building blocks (B: bifurcation, T1, T2 and T3: tubes with aspect ratios diameter/length 1:1, 1:2, 1:3, respectively).

	ε_p			
	$8 \cdot 10^{-5}$	$4 \cdot 10^{-5}$	$2 \cdot 10^{-5}$	10^{-5}
B	82	131	189	265
T1	54	86	133	198
T2	59	92	131	181
T3	29	45	66	95

of the divergence-preserving Piola transformation, which is defined as explained in Section 3.3.1.

Remark 4.7. In Section 4.2.1 we describe how the nonaffine deformation $\varphi_{z_j}(\cdot; \underline{\mu}_j)$ is defined in analytic form for the tubes. Consequently, for those building blocks, the Jacobian is computed exactly (although, due to the reasons discussed in Remark 3.4 and Remark 4.4, the output of the Piola transformation is only approximately divergence-free in practice). The nonaffine deformation of the bifurcation is performed by prescribing

4.4. The Reduced Basis Element method for flow in arteries

the position of the outlets in the physical configuration and by solving a linear elasticity problem such that the displacement field operates the desired rotation of such interfaces. Due to the complications of the evaluation of the Jacobian at the mesh nodes, in the bifurcation we consider $J_{(\varphi_{z_j})^{-1}}(\mathbf{x}; \underline{\boldsymbol{\mu}}_j) \approx I$. This simplification is also justified by the fact that, for the bifurcation, we restrict ourselves to small deformations.

The basis matrices $\widehat{V}_{u,i} = [\widehat{\underline{\boldsymbol{\zeta}}}_{1,i}^h | \dots | \widehat{\underline{\boldsymbol{\zeta}}}_{N_{u,i,i}}^h] \in \mathbb{R}^{N_{u,i}^h \times N_{u,i}}$ and $\widehat{V}_{p,i} = [\widehat{\underline{\boldsymbol{\eta}}}_{1,i}^h | \dots | \widehat{\underline{\boldsymbol{\eta}}}_{N_{p,i,i}}^h] \in \mathbb{R}^{N_{p,i}^h \times N_{p,i}}$ are constructed by POD as described in Section 3.2.1 by considering two tolerances ε_u and ε_p for every building block. The columns of $\widehat{V}_{u,i}$ and $\widehat{V}_{p,i}$ are made orthonormal with respect to $\widehat{X}_{u,i}^h$ (matrix discretization of the H^1 norm on the i^{th} reference building block) and $\widehat{X}_{p,i}^h$ (matrix discretization of the L^2 norm on the i^{th} reference building block) respectively, by following the procedure presented in Remark 3.1. The first four modes of velocity and pressure for each building block are depicted in Fig. 4.8. We also introduce the local basis matrices $V_{u,j}$, which are obtained by applying to each column of \widehat{V}_{u,z_j} the Piola transformation from the reference configuration $\widehat{\Omega}_{z_j}$ to the physical one Ω_j (being dependent on the geometrical parameters \mathcal{M} , these must be computed during the online phase). Table 4.1 and Table 4.2 report the size of the reduced bases for velocity and pressure in the $N_{\text{bb}} = 4$ building blocks composing the artificial geometry used for the data generation. We denote by B the bifurcation, and by T1, T2 and T3 tubes with aspect ratios diameter/length 1:1, 1:2, 1:3, respectively. The reduced bases were generated from $N_{s,B} = 2'640$, $N_{s,T1} = 7'920$, $N_{s,T2} = 7'920$ and $N_{s,T3} = 5'280$ snapshots. Moreover, the FE spaces for each of these building blocks have the following sizes: $N_{u,B}^h = 76'974$, $N_{p,B}^h = 3'552$, $N_{u,T1}^h = 42'708$, $N_{p,T1}^h = 2'162$, $N_{u,T2}^h = 76'416$, $N_{p,T2}^h = 3'830$, $N_{u,T3}^h = 103'728$, $N_{p,T3}^h = 5'211$. Although the RB sizes are considerably smaller than the FE ones, the number of basis functions needed to achieve low POD tolerances is substantial. This indicates that the amount of information carried by the snapshots impedes the reduction of the problem. The basis size could be decreased by considering narrower sampling intervals for the geometrical parameters describing each building block. However, as in Section 5.3 and Section 5.4 we test the ability of the same reduced basis to generalize to the case of geometries which are not considered in the offline phase, here we decide to allow for significant deviations of the configurations from the original geometry during the snapshots generation. It is worth noting that, in order to decrease the already high computational burden of the offline phase, we settle for a number of configurations (165) that is possibly too limited to capture the geometrical variability we consider in the dataset (see Fig. 4.7 for examples of some the configurations). As we verify in Section 5.2, the errors that we obtain in the online phase are—although sufficiently low for most cardiovascular applications—considerably larger than the POD tolerance.

4.4.1 Supremizers for pressure and coupling Lagrange multipliers

We recall that the Navier–Stokes equations represent an example of saddle-point equations and that this class of problems is associated with stability issues related to the discretization spaces employed for the primal and dual fields (velocity and pressure, respectively). Furthermore, the global system obtained from the nonconforming method introduced in Section 4.2.3 is also a saddle-point problem where the velocity and the Lagrange multipliers play the role of the primal and dual fields, respectively. In this work, we recover the well-posedness of the reduced global problem by enriching the velocity reduced basis by supremizers that allow us to satisfy the inf-sup condition. This procedure is formalized in Section 3.4; here, we contextualize it for the specific problem of the coupling of the reduced Navier–Stokes equations on modular geometries.

Let us consider W4.1, which we assume to be well-posed in the continuous setting. We first address the stability with respect to the constraint imposed by the pressure (divergence free velocity). At the FE level, the inf-sup condition requires the existence of $\beta_{p,j}^h \in \mathbb{R}$ such that, for all $j = 1, \dots, N_\Omega$,

$$\beta_{p,j}^h = \inf_{\underline{\mathbf{q}} \neq \mathbf{0}} \sup_{\underline{\mathbf{v}} \neq \mathbf{0}} \frac{\underline{\mathbf{q}}^T D_j^h \underline{\mathbf{v}}}{\|\underline{\mathbf{v}}\|_{\mathcal{V}_j^h} \|\underline{\mathbf{q}}\|_{\mathcal{Q}_j^h}} > 0,$$

where we used the notation $\|\underline{\mathbf{v}}\|_{\mathcal{V}_j^h} = \underline{\mathbf{v}}^T X_{u,j}^h \underline{\mathbf{v}} = \|\mathbf{v}^h\|_{\mathcal{V}_j}$ and $\|\underline{\mathbf{p}}\|_{\mathcal{Q}_j^h} = \underline{\mathbf{p}}^T X_{p,j}^h \underline{\mathbf{p}} = \|p^h\|_{\mathcal{Q}_j}$. Taylor–Hood elements [HT74] are an example of stable choice of elements ensuring that $\beta_{p,j}^h > 0$, as mentioned in Section 1.4.3. In the RB context, we introduce

$$X_{u,j}^N = (V_{u,j})^T X_{u,j}^h V_{u,j}, \quad X_{p,j}^N = (\widehat{V}_{p,j})^T X_{p,j}^h \widehat{V}_{p,j},$$

such that $\|\underline{\mathbf{v}}^N\|_{\mathcal{V}_j^N} = \underline{\mathbf{v}}^T X_{u,j}^N \underline{\mathbf{v}}$ and $\|\underline{\mathbf{p}}^N\|_{\mathcal{Q}_j^N} = \underline{\mathbf{p}}^T X_{p,j}^N \underline{\mathbf{p}}$, and $D_j^N = (\widehat{V}_{p,z_j})^T D_j^h V_{u,j}$. The inf-sup condition becomes

$$\beta_{p,j}^N = \inf_{\underline{\mathbf{q}} \neq \mathbf{0}} \sup_{\underline{\mathbf{v}} \neq \mathbf{0}} \frac{\underline{\mathbf{q}}^T D_j^N \underline{\mathbf{v}}}{\|\underline{\mathbf{v}}\|_{\mathcal{V}_j^N} \|\underline{\mathbf{q}}\|_{\mathcal{Q}_j^N}} > 0.$$

The main idea of supremizers enrichment is to augment the reduced basis for the velocity with vectors (the supremizers) specifically computed from the pressure modes to ensure the positivity of the inf-sup constant [Roz05]. In order to ensure the efficiency of the offline phase, we adopt the approximate approach in Section 3.4; the equation to be solved for each supremizer $\widehat{\mathbf{s}}_{i,j}^h \in \mathbb{R}^{N_{u,z_j}^h}$ is the counterpart of Eq (3.14) for the case of pressure stabilization in the multi-domain context, namely, for $j = 1, \dots, N_{p,j}$,

$$\widehat{X}_{u,z_j}^h \widehat{\mathbf{s}}_{i,j}^h = (\widehat{D}_{z_j}^h)^T \widehat{\boldsymbol{\eta}}_{i,j}^h.$$

where $\widehat{X}_{u,j}^h$ and $\widehat{D}_{z_j}^h$ are the norm and divergence matrices assembled on the reference

domain, and $\widehat{\boldsymbol{\eta}}_{i,j}^h$ are the pressure modes. The coupling stabilization is performed by following the same procedure. In particular, in this case the problems to be solved for each reference building block $\widehat{\Omega}_i$ read

$$\widehat{X}_{u,j}^h \widehat{\boldsymbol{z}}_{i,j}^{[m],h} = (\widehat{B}_j^{[m],h\delta})^\top \mathbf{e}_i,$$

for $l = 1, \dots, N_\lambda^\delta$. Vector $\widehat{\boldsymbol{z}}_{i,j}^{[m],h} \in \mathbb{R}^{N_{u,z_j}^h}$ is the i^{th} coupling supremizer, $\widehat{B}_i^{[m],h\delta}$ is the matrix assembled on $\widehat{\Omega}_i$ discretizing the coupling with the m^{th} interface (specifically, if $\widehat{\Omega}_i$ is a tube $m = 1, 2$, whereas if it is a bifurcation $m > 2$) and $\mathbf{e}_i \in \mathbb{R}^{N_\lambda^\delta}$ is the i^{th} canonic vector. As discussed in Section 3.4, we assume that adopting the enriched basis $\widehat{V}_{u,j}^\dagger$, which includes the RB velocity functions $\widehat{\boldsymbol{c}}_{1,i}^h$, the pressure supremizers $\widehat{\boldsymbol{s}}_{i,j}^h$ and the coupling supremizers $\widehat{\boldsymbol{z}}_{i,j}^{[m],h}$ for every m , ensures that $\beta_{p,j}^N > 0$ and $\beta_{\lambda,j}^{[m],N\delta} > 0$ (here, $\beta_{\lambda,j}^{[m],N\delta}$ is the inf-sup constant for the coupling problem on the m^{th} interface). These speculations are supported by the empirical evidence that the linear systems we consider in Chapter 5, which are obtained by following this stabilization strategy, are well-conditioned. In the following, we simply denote by $\widehat{V}_{u,i}$ the enriched basis matrix for the velocity in $\widehat{\Omega}_i$; the enriched basis $\widehat{V}_{u,i}$ is made orthonormal with respect to $\widehat{X}_{u,i}^h$ with the Gram–Schmidt algorithm.

4.4.2 Assembly and solution of the global reduced system

Let us recall that the velocity basis \widehat{V}_{u,z_j} needs to be scaled by the Piola transformation to be employed in every subdomain in their physical setting Ω_j , for $j = 1, \dots, N_\Omega$. Hence, the basis matrix for the j^{th} subdomain is denoted $V_{u,j}$ (i.e., without hat notation, as it is referred to the deformed configuration). Let us define the global basis matrix

$$\mathcal{W} := \text{diag} \left(\left[\begin{array}{c} V_{u,j} \\ \widehat{V}_{p,z_j} \end{array} \right] \right)_{j=1, \dots, N_\Omega},$$

the matrices

$$\mathcal{M}^N := \mathcal{W}^\top \mathcal{M}^h \mathcal{W}, \quad \mathcal{A}^N(\underline{\mathbf{W}}^N) := \mathcal{W}^\top \mathcal{A}^h(\mathcal{W} \underline{\mathbf{W}}^N) \mathcal{W}, \quad \mathcal{B}^{N\delta} := \mathcal{B}^{h\delta} \mathcal{W},$$

the vector of reduced dofs for all the subdomains $\underline{\mathbf{W}}^N = \text{vec}(\mathbf{w}_1^N, \dots, \mathbf{w}_{N_\Omega}^N)$, and the vectors encoding the data $\underline{\mathbf{F}}^N := \mathcal{W}^\top \underline{\mathbf{F}}^h$ and $\underline{\mathbf{G}}^{N\delta} := \underline{\mathbf{G}}^{h\delta}$. Matrices \mathcal{M}^N and $\mathcal{A}^N(\underline{\mathbf{W}}^N)$ have been defined in Eq. (4.13). Then, the reduced residual at timestep t_{k+1} is obtained from Eq. (4.14) and reads

$$\underline{\mathbf{R}}^N(\underline{\mathbf{Y}}_{k+1}^{N\delta}) := \mathcal{H}^N \underline{\mathbf{Y}}_{k+1}^{N\delta} - \sum_{j=1}^{\sigma} \alpha_j \mathcal{H}^N \underline{\mathbf{Y}}_{k-j+1}^{N\delta} - \Delta t \beta \underline{\mathbf{F}}^{N\delta}(t_{k+1}, \underline{\mathbf{Y}}_{k+1}^{N\delta}) = \underline{\mathbf{0}},$$

where

$$\mathcal{H}^N := \begin{bmatrix} \mathcal{M}^N \end{bmatrix}, \quad \underline{\mathbf{F}}^{N\delta}(t, \underline{\mathbf{Y}}^{N\delta}) := \begin{bmatrix} \underline{\mathbf{F}}^N(t) \\ \underline{\mathbf{G}}^{N\delta}(t) \end{bmatrix} - \begin{bmatrix} \mathcal{A}^N(\underline{\mathbf{W}}^N) & (\mathcal{B}^{N\delta})^\top \\ \mathcal{B}^{N\delta} \end{bmatrix} \begin{bmatrix} \underline{\mathbf{W}}^N \\ \underline{\boldsymbol{\Lambda}}^\delta \end{bmatrix},$$

and $\underline{\mathbf{Y}}^{N\delta} := \text{vec}(\underline{\mathbf{W}}^N, \underline{\boldsymbol{\Lambda}}^\delta)$.

As discussed in Section 4.3, finding the root of nonlinear equations using the Newton–Raphson algorithm entails the solution of a nonlinear system in the tangent matrix of the corresponding residual. Formally, solving $\underline{\mathbf{R}}^N(\underline{\mathbf{Y}}) = \mathbf{0}$ given an initial guess $\underline{\mathbf{Y}}^{(0)}$ leads to the iterative algorithm

$$\underline{\mathbf{Y}}^{(l+1)} = \underline{\mathbf{Y}}^{(l)} - \left(\mathcal{J}_{R^N}(\underline{\mathbf{Y}}^{(l)}) \right)^{-1} \underline{\mathbf{R}}^N(\underline{\mathbf{Y}}^{(l)}),$$

which is equivalent to Eq.(4.15) in the reduced context.

The efficiency of the reduction relies on the fast assembly of the tangent matrix \mathcal{J}_R and residual $\underline{\mathbf{R}}^N$. Regarding the former, we consider the following approximation

$$\tilde{\mathcal{J}}_{\underline{\mathbf{R}}}^N = \begin{bmatrix} \mathcal{M}^N + \Delta t \beta \mathcal{A}_{\text{lin}}^N & \Delta t \beta (\mathcal{B}^{N\delta})^\top \\ \Delta t \beta \mathcal{B}^{N\delta} \end{bmatrix}, \quad (4.20)$$

where

$$\mathcal{A}_{\text{lin}}^N := \mathcal{W}^\top \mathcal{A}_{\text{lin}} \mathcal{W}, \quad \mathcal{A}_{\text{lin}} := \text{diag} \left(\left[\begin{array}{c} K_j^h \\ D_j^h \end{array} \right] (D_j^h)^\top \right)_{j=1, \dots, N_\Omega},$$

is the matrix obtained by neglecting the convective terms in the Navier–Stokes equations. We remark that the reduced tangent matrix features a saddle-point structure as its full order counterpart in Eq. (4.16). Therefore, system $\tilde{\mathcal{J}}_{R^N} \underline{\mathbf{X}}^N = \underline{\mathbf{B}}^N$ can be solved directly by applying the reduced version of Eq. (4.18). The advantages of this approach are: (i) the tangent matrix $\tilde{\mathcal{J}}_{\underline{\mathbf{R}}}^N$ is never entirely allocated, because every stage for applying Eq. (4.18) involves operations that are local to either subdomains or interfaces (we recall that inverting \mathcal{A}_{lin} amounts to inverting each of its diagonal blocks), (ii) as a result of approximation (4.20), the tangent matrix is the same for every solution of the linear system; hence the reduced Schur complement is assembled only once and it can be factorized—along with the other local matrices to be inverted—at the start of the simulation. The linearized version of the tangent matrix (4.20) is nonconsistent, which implies that the Newton–Raphson algorithm is not expected to convergence quadratically. However, the reduced complexity of the assembly results in a considerable performance gain overall.

The problematic part of the computation of $\underline{\mathbf{R}}^N(\underline{\mathbf{Y}})$ is evidently the nonlinear term, as

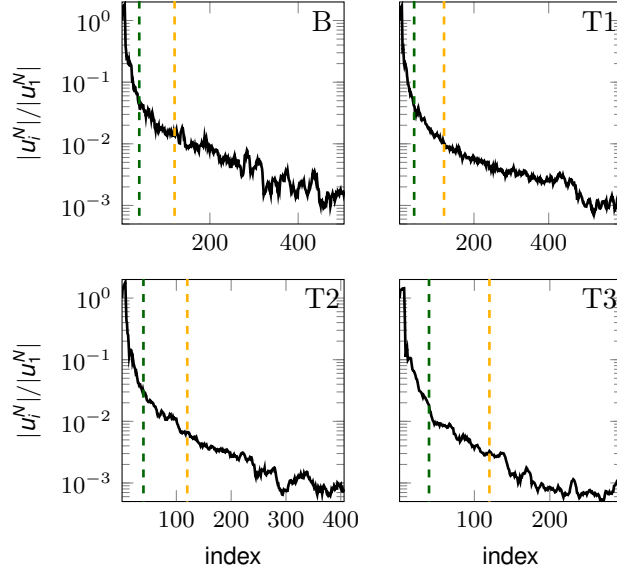


Figure 4.9 – Average over time of the RB velocity solutions—normalized with respect to the first coefficient—in the four building blocks with $\varepsilon_u = 4 \cdot 10^{-3}$ and $\varepsilon_p = 8 \cdot 10^{-5}$. The left and right dashed lines in every plot correspond to the indices 40 and 120.

the matrices encoding the linear ones are computed only once and the corresponding contributions are found at each timestep by inexpensive matrix-vector multiplications. After trivial but repetitious steps, we find that the blocks of the nonlinear part of the reduced residual read, for all $j = 1, \dots, N_\Omega$,

$$\mathbf{c}_j^N = V_{u,j}^T C(V_{u,j} \mathbf{u}_j^N) V_{u,j} \mathbf{u}_j^N. \quad (4.21)$$

One way to compute the nonlinear term for every block is then to assemble the full order nonlinear term $C(V_{u,j} \mathbf{u}_j^N) V_{u,j} \mathbf{u}_j^N$ and to project it onto the reduced space. Another strategy is based on the decomposition

$$\mathbf{c}_{i,j}^N = \sum_{l=1}^{N_{u,z_j}} \sum_{m=1}^{N_{u,z_j}} u_{l,j}^N u_{m,j}^N \mathbf{k}_{lm,j}^N, \quad (\mathbf{k}_{lm,j}^N)_i = \int_{\Omega_j} \left[(\boldsymbol{\zeta}_{m,j}^h \cdot \nabla) \boldsymbol{\zeta}_{l,j}^h \right] \boldsymbol{\zeta}_{i,j}^h. \quad (4.22)$$

The vectors $\mathbf{k}_{lm,j}^N$ are independent of the reduced solution and can be computed as a setup step in the first stages of the simulation. However, the amount of computation increases quadratically with the size of the reduced basis and could therefore nullify the performance gain. Using the fact that the velocity modes in $V_{u,j}$ are sorted in order of significance as a consequence of Proposition 3.1, it is legitimate to consider the following approximation

$$\mathbf{c}_j^N \approx \sum_{l=1}^{N_{c,z_j}} \sum_{m=1}^{N_{c,z_j}} u_{l,j}^N u_{m,j}^N \mathbf{k}_{lm,j}^N. \quad (4.23)$$

where $0 < N_{c,z_j} \leq N_{u,z_j}$. In other words, the observation that the magnitude of the reduced coefficients $u_{i,j}^N$ quickly decreases as i increases—as we show in Fig. 4.9 for the case of one of the simulations presented in Section 5.2—allows to truncate the two sums in Eq. (4.22) to the first $N_{c,j}$ terms. In the numerical results in Chapter 5 we investigate the effects of considering both Eq. (4.21) and Eq. (4.23) for the computation of the convective part of the residual.

4.5 Concluding remarks

We presented an implementation of the RBE method for the solution of the unsteady 3D Navier–Stokes equations in the context of cardiovascular simulations. We first considered the problem of coupling FE solutions defined on subdomains obtained from parametrized geometrical deformations of reference building blocks. This was necessary, as the offline phase of our ROM requires the generation of snapshots from coupled FE flow solutions obtained on a variety of geometries. In order to improve the efficiency of the coupled FE solver, we devised an ad-hoc preconditioner which takes advantage of the saddle-point structure of the discretized linear system. In the following parts of the chapter, we formulated the ROM by projecting the matrices and variables (velocity and pressure) onto the RB spaces. This procedure is beneficial because it allows us to considerably reduce the number of dofs (hence, the size of the linear system to be solved at each iteration of the Newton–Raphson algorithm).

Fig. 4.10 and Fig. 4.11 summarize the geometrical approximation, setup, and solve phases of the method in two flow charts. In Fig. 4.10 we show how, given a new target geometry, the first step of the technique computes a geometrical approximation based on the predefined building blocks. A greedy algorithm to accomplish this is presented in Section 5.1. After that step, the reduced bases (precomputed during the offline stage) are loaded, and the velocity basis is scaled by the Piola transformation. In case one decided to optimize the computation of the nonlinear term through the procedure described in Section 4.4.2, it is necessary to compute terms $\mathbf{k}_{lm,j}^N$ in all the subdomains Ω_j and for all $l \leq N_{c,z_j}$ and $m \leq N_{c,z_j}$. In the remaining part of the setup phase, the RB matrices are computed either by MDEIM (which, in this thesis, we do not consider) or by computing the FE element counterparts and projecting them onto the RB space. It is worth noting that both the computation of terms $\mathbf{k}_{lm,j}^N$ and the RB matrices are computationally demanding. The solve phase shown in Fig. 4.10 is essentially composed of an outer loop over the timesteps and an inner one in which the residual of the reduced Navier–Stokes equations is evaluated, and the current solution is updated according to the Newton–Raphson algorithm. Being that this latter step requires the solution of a linear system in the Jacobian matrix, the gain of performance coming from the reduced dimensionality of the system is achieved in this particular phase of the method. We also highlight that, depending on the availability of terms $\mathbf{k}_{lm,j}^N$, the computation of the convective term can be made considerably less expensive as discussed in Section 4.4.2.

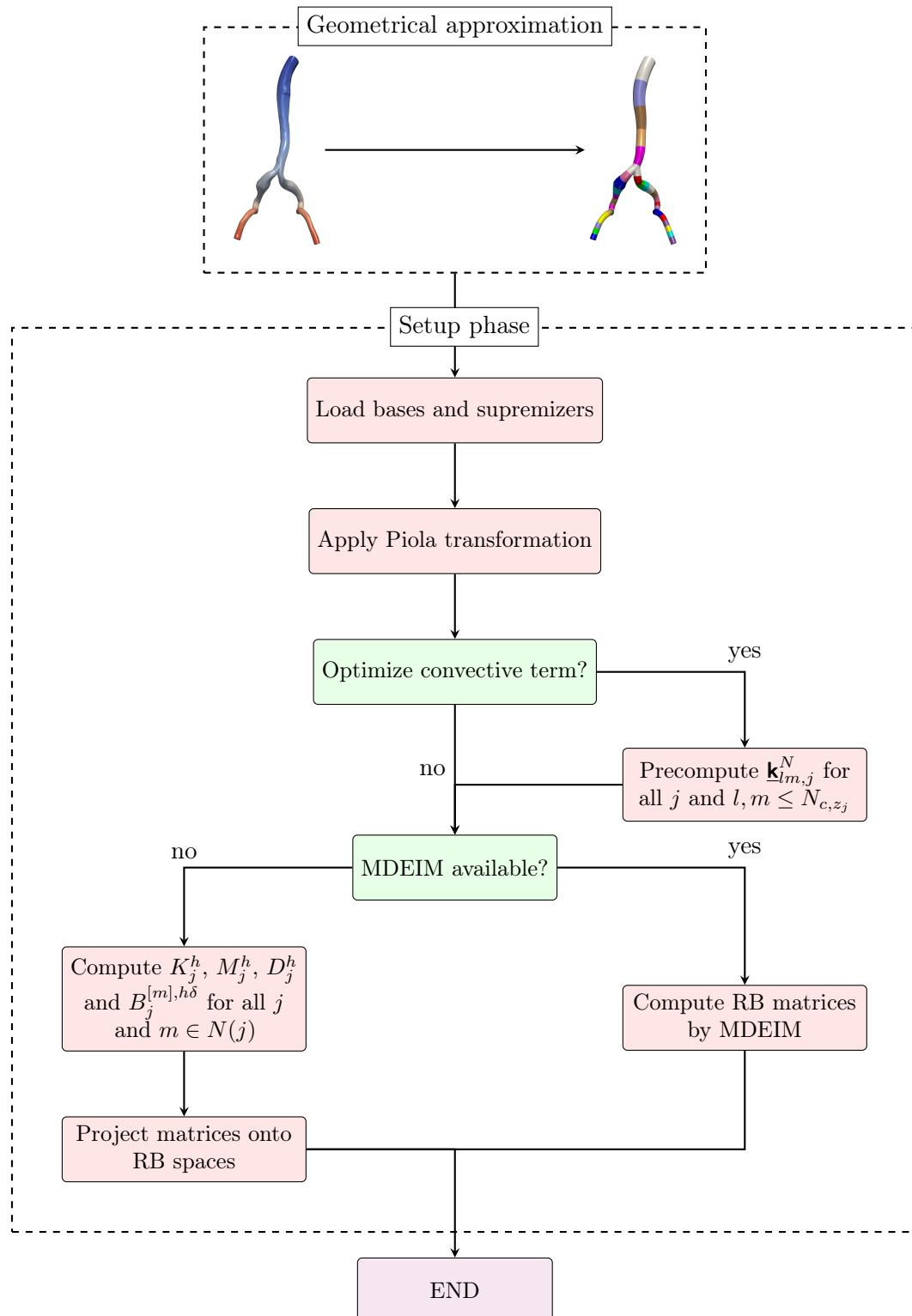


Figure 4.10 – Flowchart of the geometrical approximation and setup phases of the method.

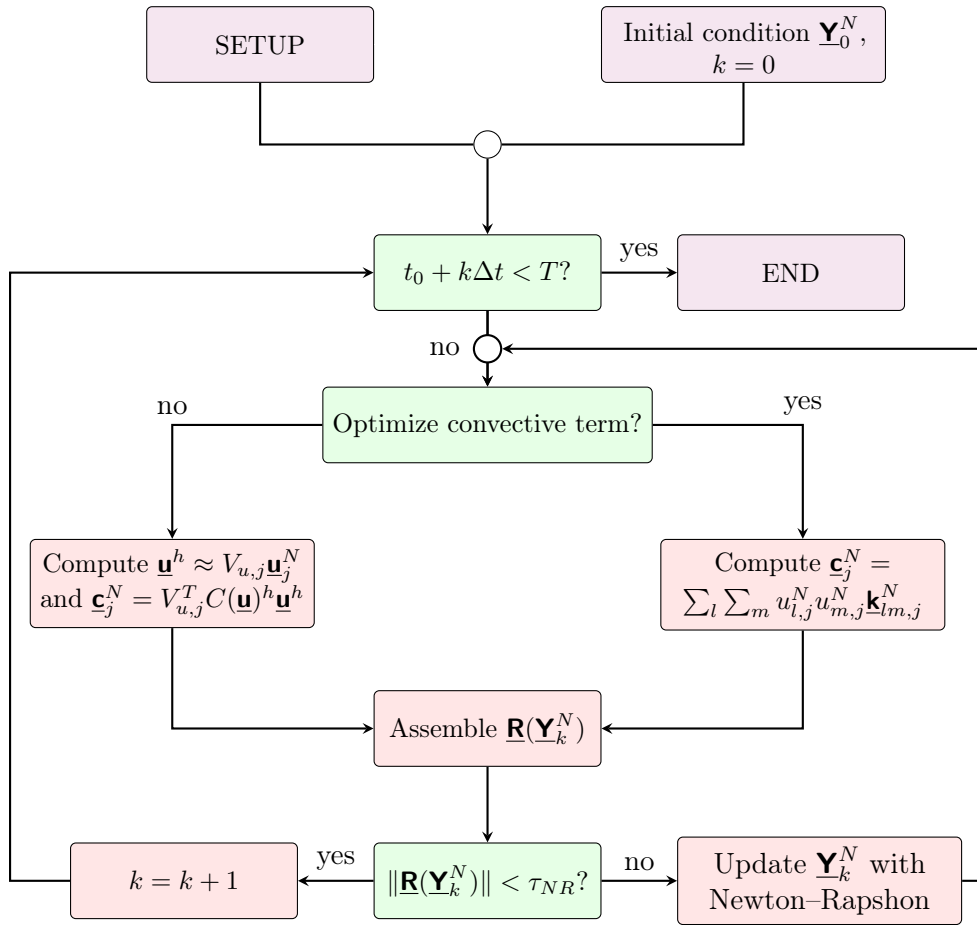


Figure 4.11 – Flowchart of the solve phase of the method.

5 Results on artificial and physiological geometries

In the previous chapter, we presented a reduced order technique for cardiovascular simulations based on the geometrical decomposition of blood vessels into a number of subdomains. Each of these is equipped with its own set of RB functions, which are used during the online phase of the method to reconstruct the local solutions. Thus, there coexist two separate levels of approximations: the first is due to the RB projection, while the second is related to the geometrical simplification of the domain.

The present chapter focuses on the analysis of the performance of the ROM in terms of both discretization and geometrical errors, as well as on the efficiency gain with respect to the FE method. We are also interested in extending the method to the general case of FE-RB coupling. Specifically, we wish to test the applicability of the reduction strategy in scenarios where a small number of subdomains is discretized by means of the FE method. The motivation to consider such a possibility is that, in certain cases, a specific area of the geometry needs to be modeled with particular care (either from the point of view of the richness of the functional space or that of the geometry itself). For example, this could occur in geometries with abnormalities (e.g., stenoses), which can hardly be represented by a predefined set of geometrical building blocks.

The chapter is structured as follows. In Section 5.1, we present the greedy algorithm for the generation of modular geometries that has been used to approximate some of the geometries considered in the numerical simulations. Then, we devise three different numerical simulations to address each of the questions mentioned above. In Section 5.2, we consider the same modular artificial geometry employed in the offline phase presented in Section 4.4, and we compare the results obtained by solving the flow problem using the RB and the FE methods. The advantage here is that we are able to rigorously compute the H^1 and L^2 error of velocity and pressure, respectively, being that the computational mesh is the same in the reference and reduced solutions. In Section 5.3, we consider a simple but more physiological geometry of an aorta and the two iliac arteries. In this case, we compare the results obtained with the RB method against the ones obtained on a

reference geometry (i.e., not partitioned into approximated subdomains) with SimVascular [UWM⁺17]. Finally, in Section 5.4 we consider a geometry (an aortic arch with carotid and subclavian arteries) in which the bifurcations are not well approximated by the reference one. Therefore, the bifurcations here are treated by the FE method.

For all the simulations, we fix $\rho_f = 1.06 \text{ gr cm}^{-3}$, $\mu_f = 0.04 \text{ gr cm}^{-1} \text{ s}^{-1}$, and we consider the same choice for the discrete Lagrange multipliers space as in the snapshot generation phase presented in Section 4.4 (i.e. $N_\lambda^\delta = 63$ for each interface). Importantly, we choose to employ the same set of RB basis functions computed in the offline phase described in Section 4.4 in all the test cases in this chapter. This is motivated by the perspective of employing the method in realistic scenarios which may be considerably different from the ones explored during the offline phase.

The results presented in Section 5.2 and Section 5.3 are taken from [PPMD20].

5.1 Automatic generation of modular geometries

We present in this section, the algorithm that has been developed for the generation of the physiological geometries considered in Section 5.3 and Section 5.4. Although not optimal from the accuracy standpoint, this algorithm is based on an efficient greedy strategy, which allows us to generate approximated geometries with a negligible computational effort.

Before delving into the details of our approach, in Section 5.1.1 we recall the basics of the typical pipeline that is commonly followed in practice to generate computational meshes out of medical images. Then, in Section 5.1.2 we describe the main features of our greedy algorithm. Finally, in Section 5.1.3 we expose the limitations of the approach and possible alternative ways to tackle the problem of the automatic generation of modular geometries.

5.1.1 Generation of meshes for cardiovascular simulations

The typical pipeline for the generation of meshes for cardiovascular simulations starts with medical images of the region to be modeled. These are usually large files obtained from standard imaging techniques—such as computed tomography (CT) or magnetic resonance imaging (MRI)—which are commonly stored in Digital Imaging and Communications in Medicine (DICOM) format.

Given the medical image of interest, the computational mesh is generated in four steps.

1. The first step is the identification of the centerlines of the vessel(s) that compose the cardiovascular district of interest. This process is usually performed manually

5.1. Automatic generation of modular geometries

and requires determining a set of points that will be marked as members of the centerline. The centerline is then found as interpolation of these manually defined points, for example, by using B-Splines.

2. Once the centerlines are defined, the next step is the so-called segmentation stage. During this process, a number of sections of the computational domain are identified along the centerlines. Even in this case, each section is usually found by interpolating some contour points specified by the users. Automatic algorithms for segmentation based on Neural Networks have been recently proposed and have shown great promise in achieving results close to human precision in a fraction of the required time [MWM19, Mah20, MFSM20].
3. The contours defined during the segmentation stages are lofted to generate a single Computer-Aided Design (CAD) model, usually by a B-Splines- or NURBS-based approach.
4. Finally, the computational mesh is generated out of the CAD model by automatic algorithms. We refer to TetGen [Si15] for an example of software implementing one of such algorithms based on Delaunay triangulations.

The entire pipeline is summarized in Fig. 5.1, where we show the output generated by SimVascular [UWM⁺17] when generating the mesh for a geometry of an aorta, carotids and subclavian artery.

5.1.2 A greedy algorithm for geometric assembly

We take advantage of the centerline and contours generated during the first two steps of the procedure presented in the previous section to devise an efficient algorithm to assemble the building blocks representing the vascular district in an optimal manner. The algorithm is described as greedy because the choices made at each step are only locally optimal.

Initially, the centerlines are used to identify the locations of the bifurcations. We do not focus on the details of the algorithm for what concerns the optimal placement of these building blocks. From a high-level perspective, in the case of parametrized bifurcations (such as the one considered in Section 5.3), the orientation and geometrical parameters $\underline{\mu}$ are determined by finding the position of the desired inlet and the desired outlets and by identifying the optimal choice of parameters through an optimization process similar to the one discussed in the following for the tubes. In case the bifurcations are directly taken from the geometry of interest (such is the case for the simulation presented in Section 5.4), the subdomain is already placed in the optimal location, and no further optimization is needed.

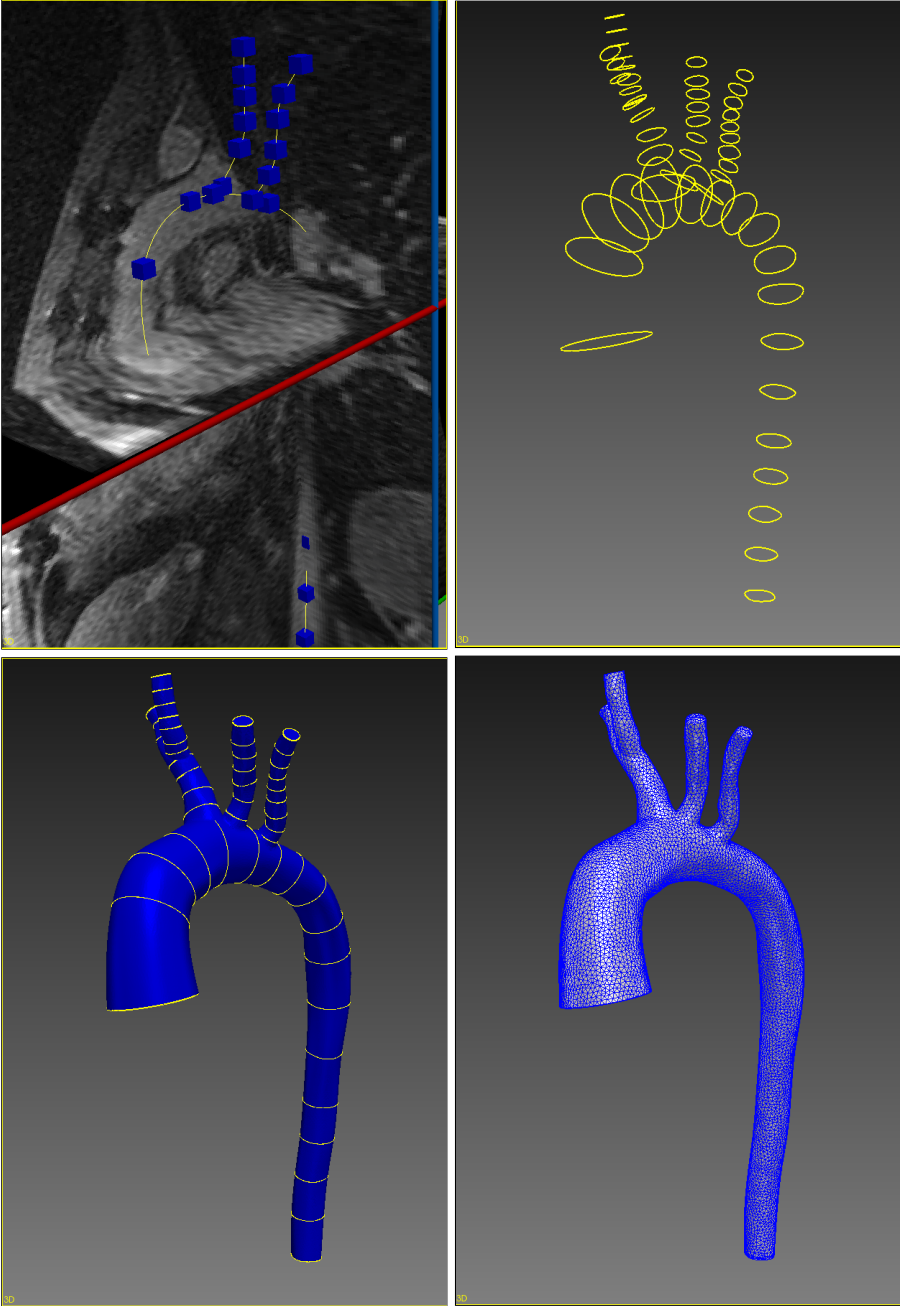


Figure 5.1 – Pipeline for the generation of meshes in cardiovascular simulations. From left to right and from top to bottom: centerline generation, segmentation, model lofting, and mesh generation.

5.1. Automatic generation of modular geometries

Once the bifurcations are in the correct position, these need to be connected through deformed tubes that mimic the model geometry. We follow a greedy algorithm in which the tubes are placed sequentially starting from the first target interface; the advantages and drawbacks of this choice are discussed in Section 5.1.3. After the first tube is placed in a way that certain conditions—which will be discussed in the following—are satisfied, the second tube is positioned such that its inlet interface matches the outlet interface of the first one, and so on. Finally, the last tube is placed so that its outlet corresponds to either a physical outlet or the interface of a bifurcation that has been previously placed. The core of this strategy are the conditions that the tubes need to satisfy in the physical configuration, which is the topic of the rest of this section.

To formalize the algorithm, we consider the following setting. Let $\mathbf{z}(t) : \mathbb{R} \mapsto \mathbb{R}^3$ a curve modeling the centerline of the vessel. We associate with every point $t \in \mathbb{R}$ a circular interface $\Gamma(t) \in \mathbb{R}^3$ with radius $R(t)$; $\Gamma(t)$ is centered in $\mathbf{z}(t)$ and is oriented perpendicularly with respect to the tangent $\dot{\mathbf{z}}(t)$ to the curve in t . These circular interfaces model the contours extracted during the segmentation phase.

Remark 5.1. The contours extracted during the segmentation phase are not perfect circles. Given the contour at coordinate t , we approximate $R(t)$ as the average distance between each point of the contour and the corresponding centerline point. Moreover, the contours are only available at a discrete set of points along the centerline of the vessel. The continuous function $R(\cdot)$ can be obtained, for example, through interpolation of the values $R(t_1), \dots, R(t_s)$ (where t_i are the coordinates of the contours in the arclength). We consider linear interpolation, but more sophisticated approaches are also possible.

We introduce the tube $\Omega_T(\underline{\boldsymbol{\mu}}) = \Phi(\widehat{\Omega}_T; \underline{\boldsymbol{\mu}})$, which is obtained from the parametrized deformation of a reference tube $\widehat{\Omega}_T$. As explained in Section 4.2, we focus on geometrical maps of the form

$$\Phi(\widehat{\mathbf{x}}; \underline{\boldsymbol{\mu}}) = Q(\underline{\boldsymbol{\mu}})\varphi(\widehat{\mathbf{x}}; \underline{\boldsymbol{\mu}}) + \mathbf{t}(\underline{\boldsymbol{\mu}}), \quad \forall \widehat{\mathbf{x}} \in \widehat{\Omega}_T,$$

where $Q(\underline{\boldsymbol{\mu}}) \in \mathbb{R}^{3 \times 3}$ is a rotation matrix, $\mathbf{t}(\underline{\boldsymbol{\mu}}) \in \mathbb{R}^3$ is a translation vector, and $\varphi(\cdot; \underline{\boldsymbol{\mu}})$ is a nonaffine transformation. The tube in the deformed configuration $\Omega_T(\underline{\boldsymbol{\mu}})$ is characterized by an inlet face Γ_{in} and an outlet face Γ_{out} ; we recall that $\varphi(\cdot; \underline{\boldsymbol{\mu}})$ takes the form $\varphi(\cdot, \underline{\boldsymbol{\mu}}) = \varphi_{\text{b}}(\cdot; \mu_2, \mu_3) \circ \varphi_{\text{s}}(\cdot; \mu_1, \mu_2)$, as described in Section 4.2.1, and that it is designed to preserve circular faces for every value of the parameter $\underline{\boldsymbol{\mu}}$. Parameters μ_1 , μ_2 and μ_3 model the deformation in the outlet radius, arclength and bending angle, respectively.

Let us consider the following problem. Given an interface $\Gamma(t_{\text{in}})$, we wish to find the values of the parameter $\underline{\boldsymbol{\mu}}$ such that

$$\Gamma_{\text{in}}(\underline{\boldsymbol{\mu}}) = \Gamma(t_{\text{in}})$$

and such that $\Gamma_{\text{out}}(\underline{\boldsymbol{\mu}})$ is *sufficiently close* to $\Gamma(\tilde{t})$, for some value $\tilde{t} \in \mathbb{R}$. The condition

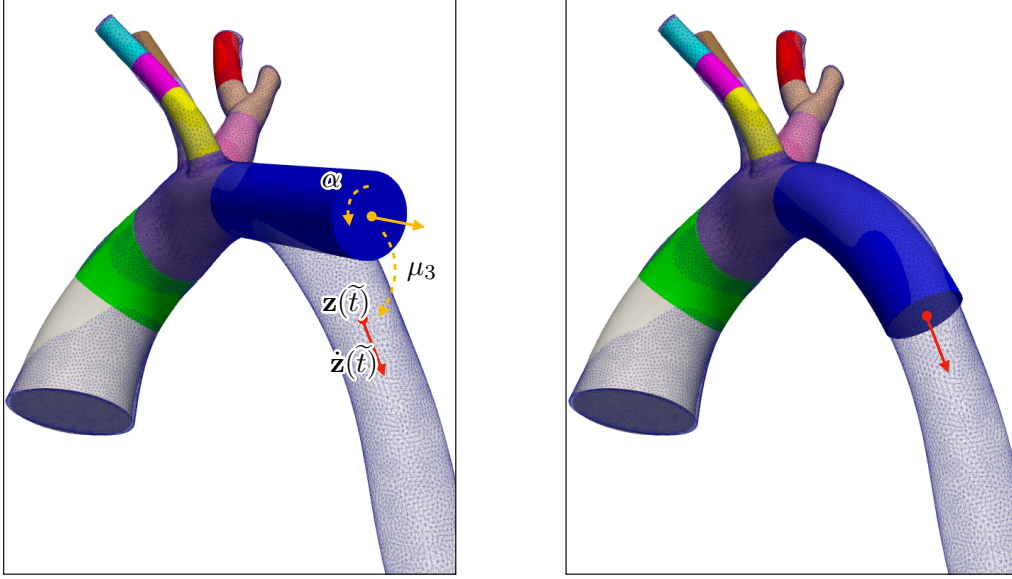


Figure 5.2 – After the tube is rotated, scaled and translated so that $\Gamma_{\text{in}} = \Gamma(t_{\text{in}})$ is satisfied, the two remaining parameters (the rotation angle along its axis α and the bending angle μ_3) are determined such that the center and the normal of the outlet Γ_{out} approximately match the centerline point $\mathbf{z}(\tilde{t})$ and the tangent $\dot{\mathbf{z}}(\tilde{t})$.

on the inlet interface is readily satisfied, as it is easy to determine $\underline{\boldsymbol{\mu}}$ such that $Q(\underline{\boldsymbol{\mu}})$ and $\mathbf{t}(\underline{\boldsymbol{\mu}})$ perform the desired rotation, scaling and translation.

Remark 5.2. Due to the axial symmetry of the interface $\Gamma_{\text{in}}(\underline{\boldsymbol{\mu}})$, there exists an additional free degree of freedom α which represents the rotation of the tube about its axis. This parameter is fixed in the remainder of the algorithm.

The condition on the outlet face $\Gamma_{\text{out}}(\underline{\boldsymbol{\mu}})$ needs to be addressed with greater care. We recall that maps (4.2) and (4.3) are such that, upon application of the nonaffine function $\boldsymbol{\varphi}(\cdot; \underline{\boldsymbol{\mu}})$, the arclength of the tube is unchanged (let us denote this quantity L). We assume that parameter μ_2 , which controls the scaling in the longitudinal direction, does not play any role at the moment. Its importance will be highlighted in the following. Therefore, after the affine transformation mapping $\Gamma_{\text{in}}(\underline{\boldsymbol{\mu}})$ into $\Gamma(t_{\text{in}})$, it is possible to find \tilde{t} by looking for the point at distance L from $\mathbf{z}(t_{\text{in}})$. Parameter μ_1 (i.e. the parameter which controls the radius in the deformed configuration) is then determined by $R(\tilde{t})$. Then, the value of the geometrical parameters $\underline{\boldsymbol{\mu}}$ which have not been determined yet (namely, the bending angle μ_3 and the rotational angle α about the normal of interface Γ_{in} , as shown in Fig. 5.2) are found by solving the optimization problem

$$\underline{\hat{\boldsymbol{\mu}}} = \arg \min_{\underline{\boldsymbol{\mu}} \in \mathcal{D}} K_1 \frac{\|\Phi(\hat{\mathbf{c}}_{\text{out}}; [\underline{\hat{\boldsymbol{\mu}}}; \underline{\hat{\boldsymbol{\mu}}}] - \mathbf{z}(\tilde{t}))\|_2}{\|\mathbf{z}(\tilde{t})\|_2} + K_2 \left| \frac{\Phi(\hat{\mathbf{n}}_{\text{out}}; [\underline{\hat{\boldsymbol{\mu}}}; \underline{\hat{\boldsymbol{\mu}}}]^T \dot{\mathbf{z}}(\tilde{t}))}{\|\dot{\mathbf{z}}(\tilde{t})\|_2} - 1 \right|,$$

5.1. Automatic generation of modular geometries

where $\hat{\mathbf{c}}_{\text{out}}$ and $\hat{\mathbf{n}}_{\text{out}}$ are center and normal of the outlet face $\hat{\Gamma}_{\text{out}}$ in the reference configuration, $\dot{\underline{\mu}} = [\mu_3, \alpha]$ and $\tilde{\underline{\mu}} = \underline{\mu} \setminus \dot{\underline{\mu}}$ (the latter is the set of all parameters which, at this stage, have already been fixed), and $K_1 \in \mathbb{R}$ and $K_2 \in \mathbb{R}$ are two positive hyperparameters to be tuned. Therefore, the bending and rotational angles are determined by ensuring that the center of the outlet face is close to the target point $\mathbf{z}(\tilde{t})$ and that its normal is nearly parallel to the tangent of the centerline. The optimization problem can be numerically solved, for example, by employing the gradient descent method [BB88, GBC16]. We remark that the Jacobian of the functional to be minimized can be efficiently approximated by finite differences, as the optimization problem is defined on only two parameters and every evaluation of the functional itself is inexpensive.

The last parameter which has not been discussed yet, i.e., μ_2 , controls the shortening or stretching of the tube in the longitudinal direction. This parameter plays a role only when the last tube in a particular segment with no bifurcations is placed. Specifically, whenever the target point $\mathbf{z}(\tilde{t})$ is close to the final outlet of the region to be reconstructed, the length of the tube is scaled so that the outlet Γ_{out} is approximately coincident with such final outlet.

5.1.3 Limitations and alternatives

The method discussed in Section 5.1.2 allows us to efficiently “fill the gap” between two interfaces in the target vessel with tubes that approximately follow the centerline and the radius of the actual geometry. However, this approach suffers from several disadvantages.

Firstly, by sequentially placing the tubes in a greedy fashion, there is no guarantee that, at each stage, the choices made in the previous ones are optimal. In particular, even if we did not discuss this possibility in this section, the library of building blocks might be composed of tubes with different aspect ratios (such is the case, for example, in the numerical simulations in Chapter 5), and, whenever a new tube needs to be placed, we are faced with the choice of the best building block for that particular section of the vessel. Even if one choice could be optimal locally, this does not imply the global optimality of the final result.

Secondly, the objective of the algorithm is to maximize the similarity between the center and normals of the tubes and the points in the centerline and its tangent. However, these optimization problems are only based on quantities defined in a very sparse set of locations along the centerlines, and, as a result, the tubes do not account for variations from the target geometry within the inlet and outlet.

We believe that the first limitation could be overcome by adopting a decision tree-based strategy [RM05], which would allow us to explore the landscape of possibilities in a much more exhaustive way compared to the greedy approach. At each step of the algorithm, whenever a choice—e.g., which type of tube should be used—is made, a new branch of

the decision tree is created. Each leaf of the decision tree is associated with a certain value of a functional to be optimized (based on some feature of the approximated and target geometries), and at the end of the algorithm, the best choice is selected according to the value of such functional. This ultimately leads to a final decision that is more informed and optimal in the set of possibilities considering while exploring the tree.

Finally, the second drawback could be overcome by considering a second layer of optimization which would take into account the whole set of geometrical parameters \mathcal{M} . After the first approximated geometry is generated by the greedy algorithm, we would define the optimal set of geometrical parameters $\widetilde{\mathcal{M}}$ as the solution of the optimization problem

$$\widetilde{\mathcal{M}} = \arg \min_{\mathcal{M} \in \Pi_{i=1}^{N_\Omega} D_i} \mathcal{J}(\Omega^m(\mathcal{M}), \Omega)$$

where functional $\mathcal{J}(\Omega^m, \Omega)$ could be defined, for example, as

$$\mathcal{J}(\Omega^m(\mathcal{M}), \Omega) = \frac{|\Omega \setminus (\Omega^m(\mathcal{M}) \cap \Omega)|}{|\Omega|}$$

(here, notation $|\cdot|$ indicates the measure of the set, namely its volume). This additional step would allow to modify $\Omega^m(\mathcal{M})$ to account for the similarity with the original geometry Ω . However, the evaluation of functionals depending on the volume of Ω and $\Omega^m(\mathcal{M})$ is considerably expensive, and the fact that the optimization problem depends on the whole set of parameters \mathcal{M} makes the computation of the Jacobian by finite differences in this scenario unfeasible. Therefore, such a strategy would represent a viable alternative to the sole greedy algorithm only in the presence of efficient ways to evaluate the discrepancy between $\Omega^m(\mathcal{M})$ and Ω .

5.2 Online phase on an artificial geometry

We evaluate the performance of our ROM on the artificial problem employed for the generation of the reduced basis. We recall that the *Artificial geometry* in Fig. 4.7, which is not included in the set of 165 configurations used to produce the snapshots, is a legitimate candidate to test the accuracy on geometries not “seen” in the offline phase. The solution by the RB method is compared to the global solution obtained by considering FE solutions in each subdomain (with the same meshes used in the RB case) coupled with the discretization strategy presented in Section 4.2.3. We consider the same choice for the discrete Lagrange multipliers space as in the snapshot generation phase. The reasons for considering such comparison are the following: (i) being that the geometry is exactly the same, it is possible to easily compute H^1 and L^2 errors for velocity and pressure in order to verify the convergence of the RB approximation with respect to the FE one, and (ii) it is feasible to fairly discuss the speedup achieved by the RB

5.2. Online phase on an artificial geometry

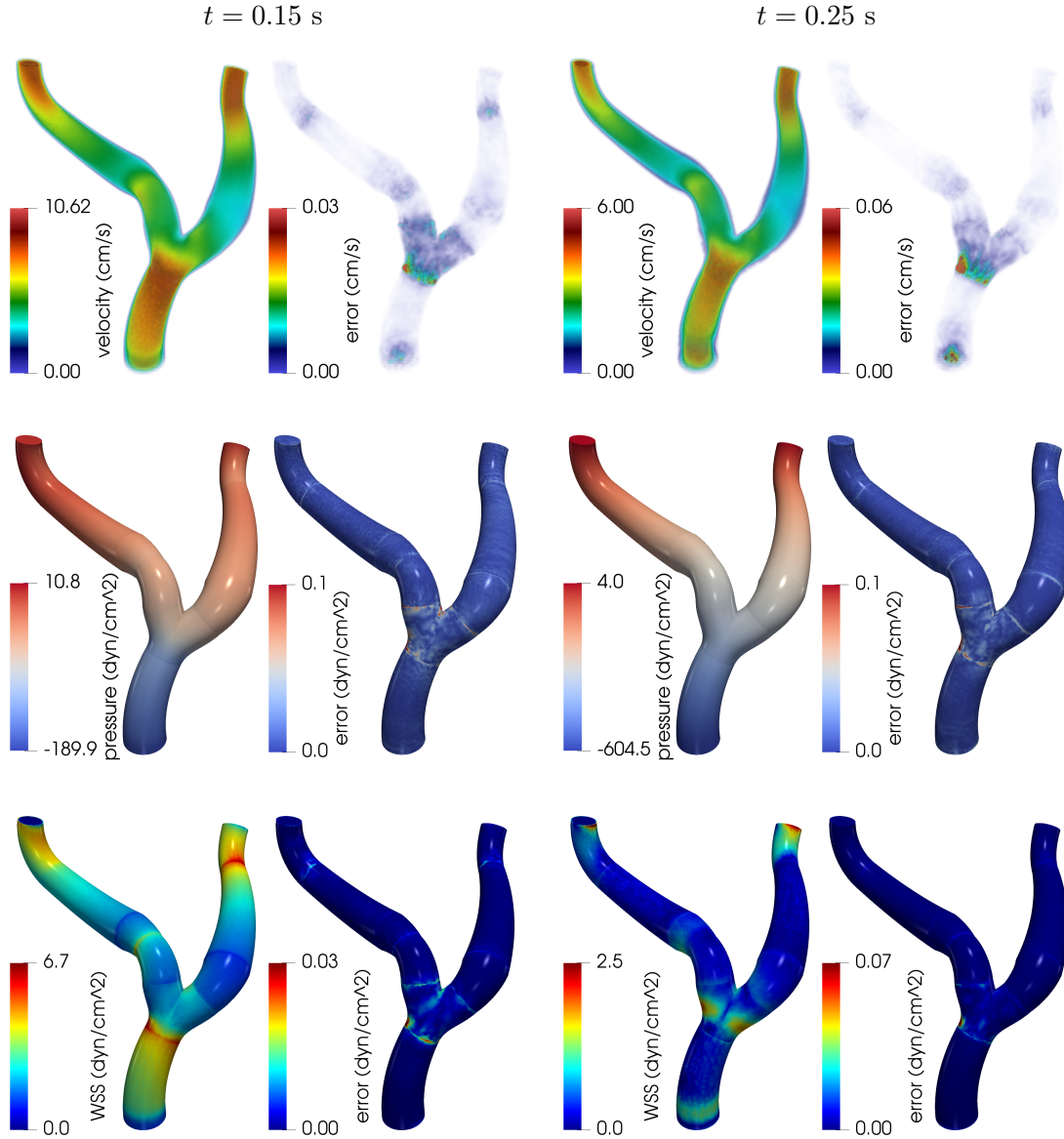


Figure 5.3 – The left and right columns—each composed of two sub-columns of plots—refer to time $t = 0.15$ s and $t = 0.25$ s, respectively. First row: velocity magnitude volume plot of the RB solution (sub-column left) and magnitude of the point-wise velocity error w.r.t. the FE solution (sub-column right). Second row: pressure plot of the RB solution (sub-column left) and absolute value of the point-wise pressure error w.r.t. the FE solution (sub-column right). Third row: magnitude of the WSS of the RB solution (sub-column left) and magnitude of the point-wise WSS error w.r.t. the FE solution (sub-column right). The RB solution corresponds to the choice $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$.

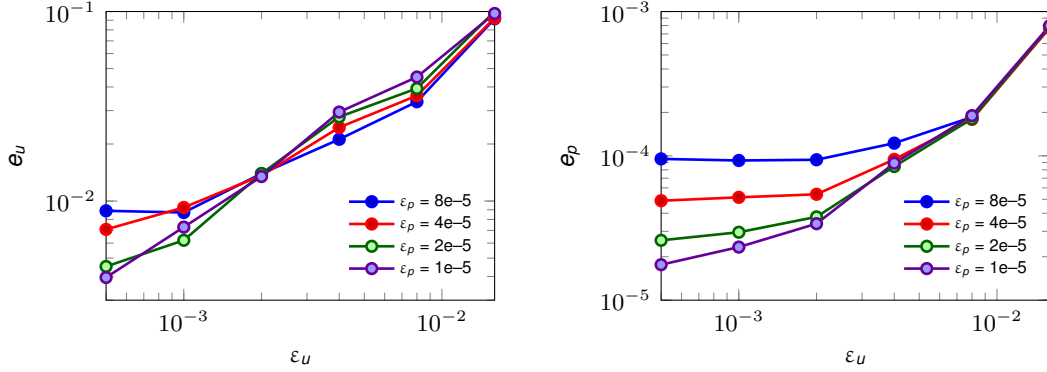


Figure 5.4 – Error in velocity e_u (left) and error in pressure e_p (right), computed as in Eq. (5.1), in function of the POD tolerances for velocity and pressure ε_u and ε_p .

method, as the RB and FE solutions share the same computational mesh. As for the generation of the reduced basis, we consider $t_0 = 0$ s, $T = 0.3$ s, and a second-order BDF scheme with $\Delta t = 2.5 \cdot 10^{-3}$ s. The boundary conditions are the same as the ones shown in Fig. 4.7; therefore, by considering the mean velocity and diameter of the inlet as characteristic velocity and dimension of the problem, the Reynolds number is about $\text{Re} = 150$. Fig. 5.3 shows, in the first two rows, the magnitude of the velocity field and pressure distribution at times $t = 0.15$ s and $t = 0.25$ s obtained with the RB method and the corresponding point-wise errors with respect to the FE solution. The POD tolerances in every subdomain have been set to $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$. We observe that, despite the global mesh being nonconforming, the velocity and pressure appear to be quite smooth at the interfaces. The comparison with the FE solution highlights the fact that the largest errors are committed in the region of the bifurcation. This is likely due to the fact that, as discussed in Section 4.4, the reduced basis for the corresponding building block (B) is based on a smaller number of snapshots. However, the RB and the FE solutions match quite accurately overall, as the relative error is negligible in every part of the domain. The last row of Fig. 5.3 depicts the distribution of the magnitude of the WSS on the boundary of the artery in the RB solution and the magnitude of the error with respect to the FE one. The influence of the coupling is noticeable: indeed, it is clearly possible to spot the interfaces as regions with abnormally low or high WSS. However, this effect is not due to the RB approximation but rather to the coupling strategy: indeed, the RB and FE approximations are extremely close, as proven by the small magnitude of the error on the WSS.

Fig. 5.4 shows the H^1 and L^2 relative errors in velocity and pressure integrated in time, defined as

$$e_u^2 := \frac{\int_0^T \|\mathbf{u}^h - \mathbf{u}^N\|_{\mathbf{b},\mathcal{V}}^2}{\int_0^T \|\mathbf{u}^h\|_{\mathbf{b},\mathcal{V}}^2}, \quad e_p^2 := \frac{\int_0^T \|p^h - p^N\|_{\mathbf{b},\mathcal{Q}}^2}{\int_0^T \|p^h\|_{\mathbf{b},\mathcal{Q}}^2}, \quad (5.1)$$

5.2. Online phase on an artificial geometry

Table 5.1 – Overall speedups w.r.t. the FE solution and, in parenthesis, speedups of the solve part of the online phase, i.e. speedup relative to the total running time excluding the setup part in which the reduced bases are loaded and the constant matrices are assembled and projected onto the reduced spaces.

$\varepsilon_p \backslash \varepsilon_u$	$1.6 \cdot 10^{-2}$	$8 \cdot 10^{-3}$	$4 \cdot 10^{-3}$	$2 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{-4}$
$8 \cdot 10^{-5}$	33(50)	30(48)	26(43)	22(39)	17(32)	14(31)
$4 \cdot 10^{-5}$	27(46)	25(44)	23(38)	21(38)	17(35)	13(30)
$2 \cdot 10^{-5}$	28(45)	26(43)	23(41)	28(38)	15(33)	12(28)
$1 \cdot 10^{-5}$	26(45)	25(44)	22(41)	17(33)	14(30)	12(29)

where $\|\mathbf{u}\|_{\mathbf{b},\mathcal{V}}^2 = \sum_{i=1}^{N_\Omega} \|\mathbf{u}_i\|_{H^1(\Omega_i)}^2$ and $\|p\|_{\mathbf{b},\mathcal{Q}}^2 = \sum_{i=1}^{N_\Omega} \|p_i\|_{L^2(\Omega_i)}^2$ are the broken norms. The errors are plotted as functions of the velocity and pressure POD tolerances ε_u and ε_p , highlighting the convergence of the RB solution to the FE one as the reduced basis size increases. Clearly, ε_u and ε_p both contribute to the errors in velocity and pressure. Indeed, for large ε_p , e_u and e_p set on a plateau as ε_u decreases, indicating that the error in the pressure is dominating the error in the velocity. For each data point in Fig. 5.4, the corresponding speedup is reported in Table 5.1; both the full order and reduced solutions are computed on a single core. The runtime of the reference FE solution—which is composed of 641'502 dofs for velocity and pressure and 567 dofs for the Lagrange multipliers—is 66'892 s (~ 18.5 hours). The speedups are relative to the total runtime and, in parentheses, to the part of the online phase after the initial setup (which include the loading of the reduced basis, the assembly of the constant matrices, and their projection onto the reduced spaces, as shown in Fig. 4.10). The motivations to consider both speedups are twofold. Firstly, in this thesis, we do not focus on the optimization of the assembly part of the system, which could considerably increase the total speedup; such optimization could be carried out, for example, by employing (M)DEIM, as mentioned in Section 3.2.2. Secondly, the setup part of the RB algorithm is particular to the geometry we are interested in. As a matter of fact, should we be interested in solving flow problems corresponding to different boundary conditions and/or fluid properties but on the same geometry of a specific patient, the setup phase can be executed only once, and for each solution of the reduced system we take advantage of the speedups relative to the only solve phase. The gain in performance is in all cases quite substantial (at least one order of magnitude with respect to the full order solution), and we observe, as expected, the trend of increasing speedup as the size of the reduced system decreases. However, the careful profiling of the simulation highlights that most of the time of the solve phase is spent in the assembly of the reduced convective term rather than in the actual solution of the reduced system. This is because, as discussed in Section 4.4.2, the exact assembly of the reduced convective terms entails two projections and the construction of the full order convective term.

With the purpose of achieving higher speedups during the solution time, we consider the

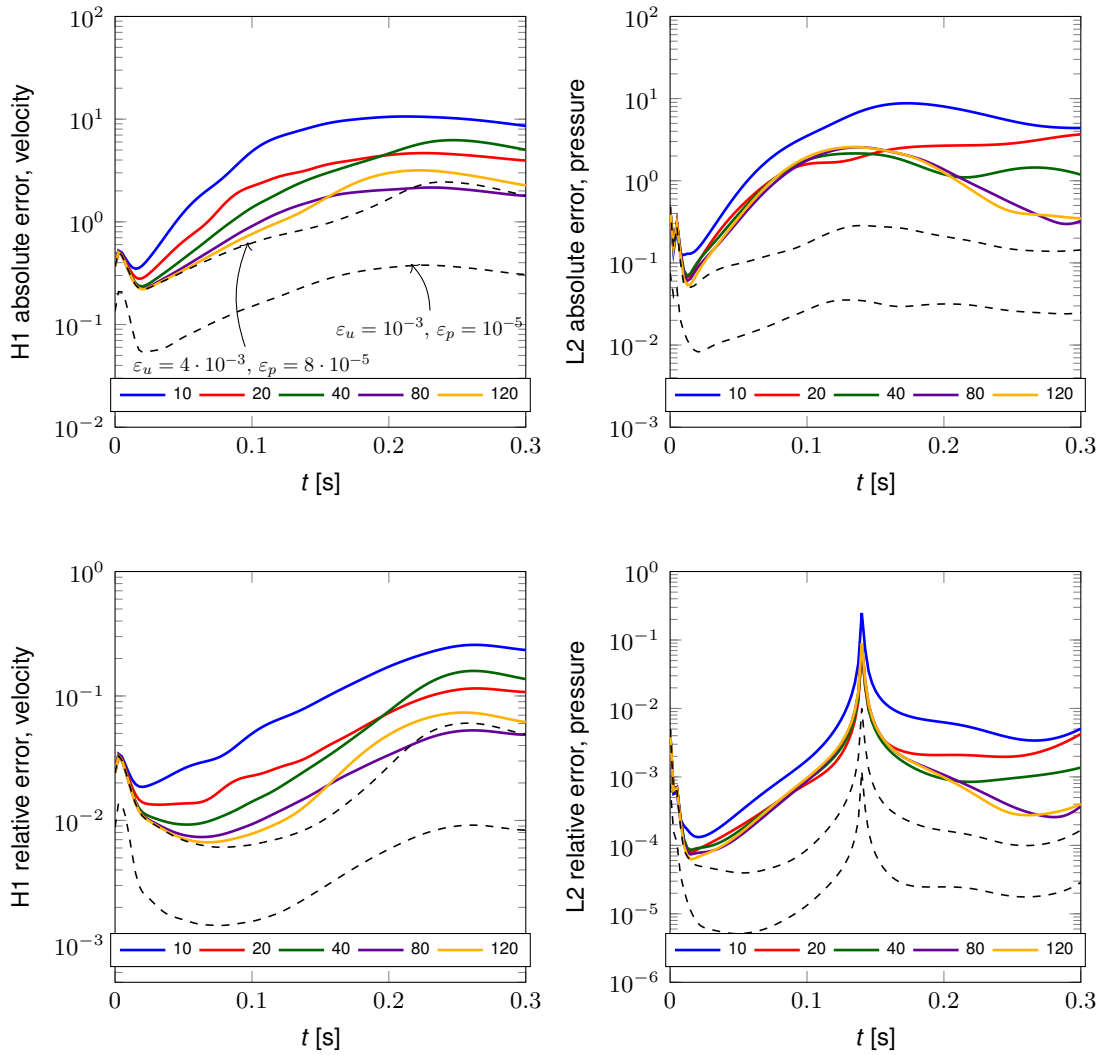


Figure 5.5 – Errors of reduced velocity and pressure against the FE solution vs time. The colored lines refer to different choices of N_c for the approximation of the nonlinear convective term. The black dashed lines show the reference errors obtained without approximation of the convective term with $\epsilon_u = 4 \cdot 10^{-3}$ and $\epsilon_p = 8 \cdot 10^{-5}$ (which are the same tolerances used in the simulations corresponding to the colored lines) and $\epsilon_u = 10^{-3}$ and $\epsilon_p = 10^{-5}$.

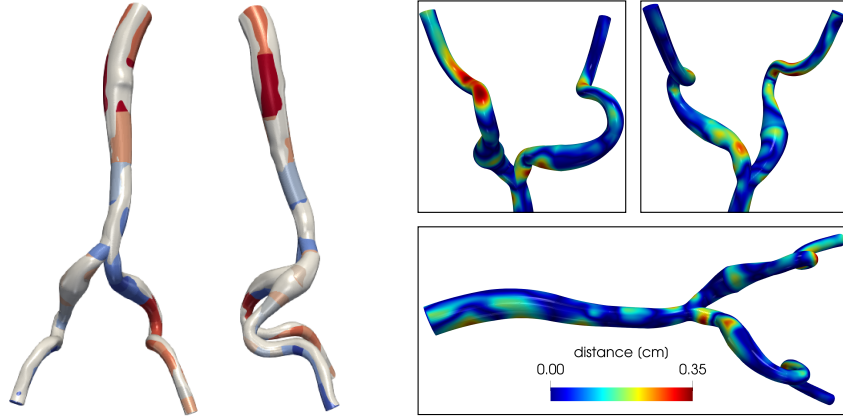


Figure 5.6 – On the left, qualitative comparison of the reference mesh with the decomposed one. On the right, quantitative estimation of the distance between the two.

approximation of the convective term given in Eq. (4.23). Fig. 5.5 shows the absolute and relative H^1 and L^2 errors in velocity and pressure over time in function of different degrees of truncation of the convective term (i.e. different values of N_{c,z_j} , which we set equal to $N_{c,z_j} = N_c$ for every subdomain). The achieved speedups are, from $N_c = 10$ to $N_c = 120$ and using the same notation adopted in Table 5.1, 56(998), 36(620), 22(464), 10(313), 5(215). The POD tolerances are constant and take the values $\varepsilon_u = 4 \cdot 10^{-3}$ and $\varepsilon_p = 8 \cdot 10^{-5}$. We remark that the values of N_c are to be considered in relation with the decay of the RB solutions shown in Fig. 4.9, which refer to the velocity coefficients of the reference solution corresponding to $\varepsilon_u = 4 \cdot 10^{-3}$ and $\varepsilon_p = 8 \cdot 10^{-5}$. As expected, the runtime of the solve phase is greatly decreased with respect to both the FE solution, against which the speedup achieved is always higher than 200, and with respect to the RB solution with the convective term computed as in Eq. (4.21). As N_c increases, the total speedup rapidly decreases due to the quadratic dependence on that parameter of the number of integrals computed during the setup phase. Nevertheless, we believe that this strategy for approximating the convective term is of great benefit whenever it is required to run multiple simulations on the same geometry, as, in this scenario, the setup phase is only performed once.

5.3 Online phase on the aorta and iliac arteries

We consider a physiological geometry of an aorta with the two iliac arteries¹. Our goal is to evaluate the effects of the geometrical approximation on the solution given by our ROM. In order to do so, we employ the geometries depicted in Fig. 5.6. Specifically, on the left we show the decomposed geometry along with the “exact” one. On the right, we

¹A SimVascular tutorial based on the same geometry considered here is available on the software website (<http://simvascular.github.io/docsQuickGuide.html>).

provide a quantitative analysis of the difference between the two.

The flow problem consists of imposing the same inflow profile shown in Fig. 4.7 at the inlet (the aorta) and homogeneous Neumann conditions to the outlets (the iliac arteries). The Reynolds number, computed as in Section 5.2, is approximately $\text{Re} = 105$. We take $T = 1.5$ s (i.e. two heartbeats) and $\Delta t = 1.25 \cdot 10^{-3}$ s.

Remark 5.3. Although this numerical example and the one presented in Section 5.4 deal with physiological geometries, the boundary conditions that we consider at the outlets (homogeneous Neumann conditions) are not. In cardiovascular simulations, indeed, it is customary to couple each outlet with models simulating the resistance of the cardiovascular system. One possibility is to employ a 0D three-element Windkessel model. However, such models are usually dependent on the flow rate at the corresponding outlet, which therefore complicates the computation of the Navier–Stokes Jacobian by introducing additional terms (since the boundary conditions themselves are nonlinear). Hence, in this study, we limit ourselves to non-physiological boundary conditions. We refer, e.g., to [LYDM20] for more information about the three-element Windkessel model and implementation details.

As already anticipated, the reference simulation is computed with the SimVascular solver (svSolver). This software is based on the FE method with P1-P1 elements and VMS-SUPG stabilization; see Section 1.5. It is therefore challenging to devise a *fair* comparison between the ROM—which, we recall, is built upon a P2-P1 discretization—and the reference solution in terms of efficiency and accuracy. Nevertheless, we provide for the sake of completeness some data regarding the reference solution. This is computed on a fine mesh (composed of 1’823’827 nodes) that is selected by studying the convergence of the WSS on the boundary; the simulation using SimVascular took 46’457 s (~ 13 hours). It is important to notice that, due to the large number of dofs, the reference simulation with Simvascular is obtained on 28 cores; in fact, due to implementation details, the software does not allow us to compute a single-core solution for this particular problem. This is another element that complicates the comparison with our ROM, which, in the current state, is not implemented in a parallel framework.

In Fig. 5.7, we show the qualitative comparison of the velocity field magnitude, pressure, and WSS distribution on the wall at two different timesteps. The RB solution is obtained with $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$. We observe that, despite the differences in the employed geometries and in the underlying numerical discretization, the solutions share similar features. For instance, the pressure distribution is qualitatively almost identical, and the ranges for velocity and WSS magnitude achieved in every region are comparable. It is apparent, however, that most of the error (on the velocity magnitude in particular) is in the vicinity of the bifurcation. Again, this is due to the fact that our choice of geometrical parameters for the corresponding building block does not allow for the reference bifurcation to be deformed into the target one with sufficient accuracy. For example, Fig. 4.2 shows that we do not take into account the possibility of varying the

5.3. Online phase on the aorta and iliac arteries

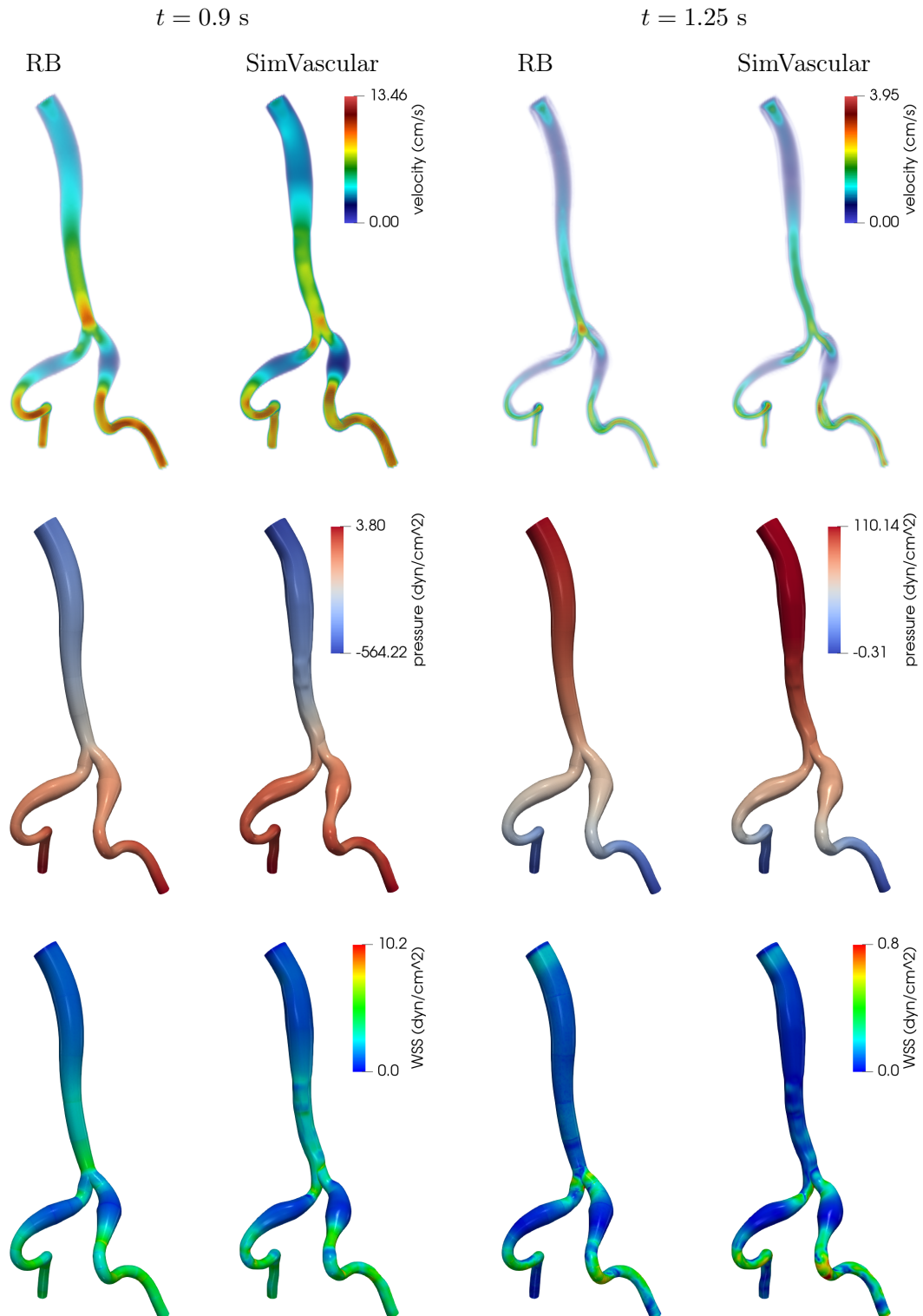


Figure 5.7 – The left and right columns refer to time $t = 0.9$ s and $t = 1.25$ s. First row: velocity magnitude volume plot of the RB and reference solutions. Second row: pressure plot of the RB and reference solutions. Third row: magnitude of the WSS of the RB and reference solutions. The RB solution corresponds to the choice $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$.

radii of the outlets, and this reflects in a large geometric error, particularly on one of the branches, as depicted in Fig. 5.6 (bottom branch in the bottom right plot).

A more quantitative analysis of the performance of the ROM with respect to the reference solution is presented in Fig. 5.8. Here, we show the average of the WSS magnitude over the three regions highlighted in the figure on the left, the pressure at the inlet and the flow rate at the outlets, for the reference solution and for RB solutions corresponding to different choices of tolerances and truncations for the approximation of the nonlinear term. We chose to focus on a “fine” RB solution (RB1), where we do not apply the truncation of the convective term, and more “coarse” but efficient RB solutions with the approximation of the convective term (RB2, RB3, RB4); for details regarding the employed POD tolerances and the number of terms in the truncated sum, we refer the reader to the caption of Fig. 5.8. The setup, solve and total runtimes in seconds for these simulations—which are run, differently from the reference simulation, on a single core—are the following: for RB1, $4'240 + 36'170 = 40'410$ (~ 11 hours, speedup of 1.1), for RB2, $19'382 + 4'760 = 24'142$ (~ 6.5 hours, speedup of 2), for RB3, $7'456 + 2'398 = 9'854$ (~ 2.5 hours, speedup of 5), for RB4, $1'816 + 3'774 = 5'590$ (~ 1.5 hours, speedup of 8). In all cases, we achieve speedups larger than one with respect to the reference simulation obtained with SimVascular (although the gain is negligible in the case of RB1) but on a single core instead of 28. From the results presented in Fig. 5.8, we note that, while the approximation of the pressure and flow rate is extremely precise for all RB settings, the performance on the WSS is more challenging. The curves for the average WSS in the two regions on the iliac arteries (B and C) are quite close to the reference ones compared to the average WSS on the bifurcation (A). However, this is likely an effect of the geometric approximation rather than the accuracy of the ROM per se. As a matter of fact, we already noted in Fig. 5.7 that the largest errors are located in that area. We also remark that the POD tolerance plays a more dramatic role in the quality of the solution than the number of terms retained in the truncated nonlinear term N_c . Indeed, the simulations with the smallest tolerances (RB1 and RB2) and the largest ones (RB3 and RB4) lead to similar results, regardless of the value of N_c . Nevertheless, truncating the convective term is beneficial to the efficiency of the ROM.

5.4 Online phase on the aortic arch

We observed how, in the numerical example of the previous section, the bifurcation is the region where the largest errors are committed. The reason is that mapping a reference bifurcation onto the physical configuration with good accuracy is a non-trivial task, even in relatively simple geometries such as the one of the aorta with the iliac arteries. Therefore, in this section, we investigate the possibility of considering a small number of “geometrically exact” subdomains while still performing a modular geometrical approximation of the remaining parts of the target geometry. The meshes of the non approximated blocks are obtained directly from the target model, and therefore a reduced

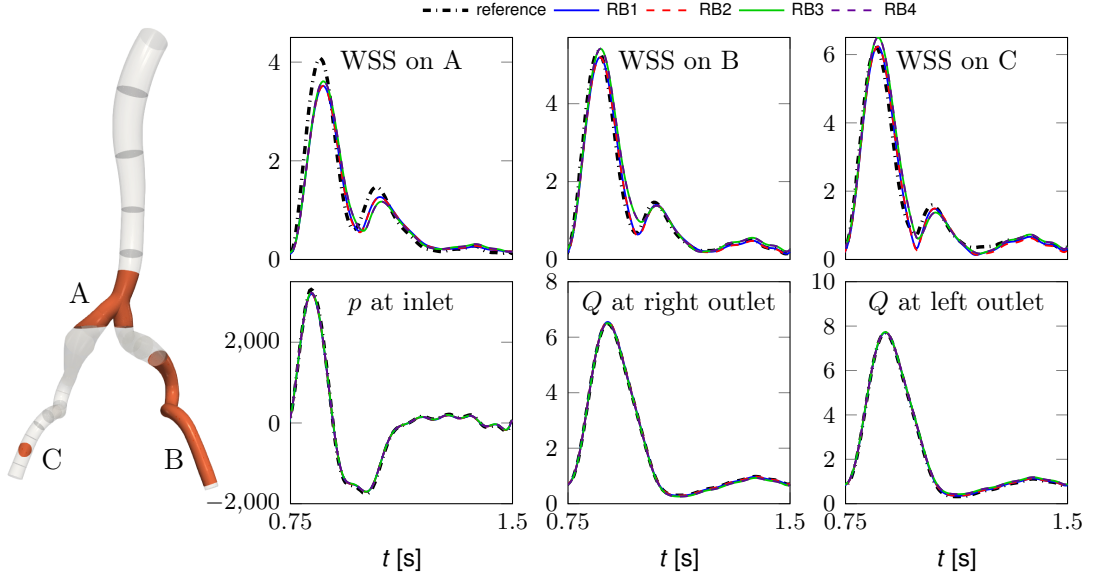


Figure 5.8 – Average WSS (in dyn/cm^2) on the three regions marked on the figure on the left (top row), and pressure p (in dyn/cm^2) and flow rates Q (in cm^3/s) at inlet and outlets, respectively (bottom row). The black dashed line refers to the reference solution computed by SimVascular, whereas the 4 colored lines are obtained with different RB settings. RB1: $\varepsilon_u = 4 \cdot 10^{-3}$, RB2: $\varepsilon_u = 8 \cdot 10^{-3}$ and $N_c = 80$, RB3: $\varepsilon_u = 6.4 \cdot 10^{-2}$ and $N_c = 40$, RB4: $\varepsilon_u = 6.4 \cdot 10^{-2}$ and $N_c = 20$. In all cases, $\varepsilon_p = 10^{-5}$.

basis defined on those subdomains is not available. Therefore, in those regions, we consider a local FE approximation.

Remark 5.4. In this section, we are mostly concerned with demonstrating a proof of concept. We do not focus on the speedup relative to a reference solution, as the current implementation lacks the necessary level of refinement to make the RB-FE hybrid method truly competitive. A particular challenge in this regard is the fact that, when dealing with FE simulations, one often prefers to adopt a sparse representation of the matrices so as to minimize the memory footprint and maximize the efficiency of basic algebraic operations. In the RB context, the matrices are instead typically dense. Therefore, an efficient resolution strategy needs to handle these two types of data structures with the goal of minimizing the conversions from one matrix representation to the other.

We consider a physiological geometry of an aorta with carotid and subclavian arteries. As shown in Fig. 5.9, the two bifurcations—one in the aortic arch and one in the brachiocephalic artery separating the right carotid artery from the right subclavian artery—are more complex than the one appearing in the previous section. Here we choose to employ P1-P1 elements with SUPG stabilization, which has been introduced in Section 1.5. This choice, combined with the fact that we consider coarse meshes in those blocks, leads to a relatively small number of dofs. The aortic arch FE space is composed of 17'781 and 5'927 velocity and pressure dofs, whereas the fluid in the bifurcation of

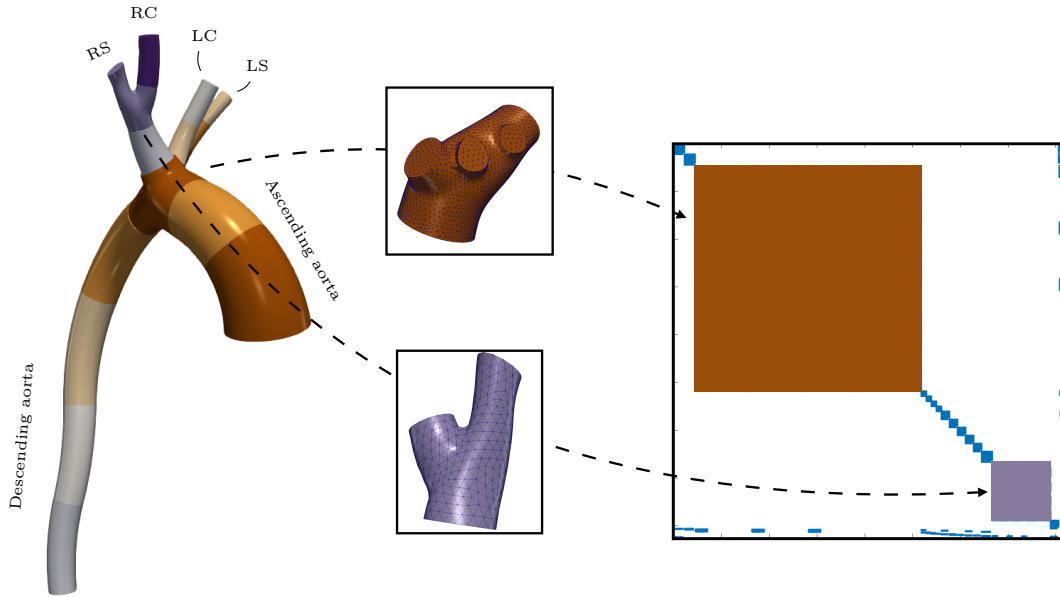


Figure 5.9 – On the left, geometry decomposed into blocks. On the right, pattern of the global matrix (the dimensions of the individual blocks are in scale) when employing the RB tolerances $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$. LC: left common carotid artery. LS: left subclavian artery. RC: right common carotid artery. RS: right subclavian artery.

the brachiocephalic artery is approximated with 4'590 and 1'530 dofs for velocity and pressure, respectively. Hence, the total number of FE dofs is 29'828. As shown in Fig. 5.9 (right), this represents the large majority of the variables in the global system when all the other blocks are approximated with the RB method. Indeed, by employing the POD tolerances $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$, the global system has size $40'300 + 945 = 41'245$, where 40'300 is the number of dofs associated to the primal variables (RB and FE dofs for velocity and pressure in all the blocks) and 945 is the total number of coupling Lagrange multiplier basis functions. A simulation with the same meshes in the RB subdomains but in which these are discretized with P2-P1 FE elements would require a total of 1'081'670 dofs for velocity and pressure.

As in the previous numerical example, we consider Dirichlet boundary conditions on the aortic root and homogeneous Neumann boundary conditions on the outlets. The imposed flow rate profile is the same as in Section 5.3, which leads to a Reynolds number of $\text{Re} = 90$. Regarding the time discretization, we employ a BDF scheme of order 2 with $\Delta t = 1.25 \cdot 10^{-3}$ s; the simulation is run from $t_0 = 0$ s to $T = 1.5$ s.

Remark 5.5. The Reynolds number we consider in this example is far from physiological, as in the aorta it is common to observe $\text{Re} = 5'000$ or higher [SS76]. However, one of the limitations of the current solver is that, as presented in Section 4.4.2, we employ a nonconsistent approximation of the Jacobian of the global system residual in order to

increase the efficiency of the online phase of the algorithm. However, our approach is valid in regimes in which the convective phenomena are not prevalent; in such cases, the Newton–Raphson algorithm converges. The accurate approximation of the blood flow in turbulent regimes requires then to devise more sophisticated ways to compute the Jacobian.

Fig. 5.10 shows the velocity field magnitude, pressure and WSS in two time instances of the simulation. Some of the conclusions drawn in the previous case also apply here: the velocity field appears qualitatively smooth at the interface—we remark that at the interfaces between RB and FE subdomains we are coupling solutions defined on quadratic finite elements with linear ones—while the WSS approximation is smooth far from the interfaces, which are clearly visible. This is likely due to the fact that, in this numerical example, the geometrical approximation at the interfaces is even more challenging than in Section 5.3 because the interfaces of the FE building blocks are not circles (hence, the match between the neighboring subdomains is not exact). The pressure also appears to be discontinuous across the interfaces, and this is probably a consequence of the geometrical approximation too.

In Fig. 5.11, we show the flow rates over the second cardiac cycle at the inlet and each outlet. Here we denote RB1 and RB2 the reduced bases obtained with POD tolerances set to $\varepsilon_u = 10^{-2}$ and $\varepsilon_u = 10^{-3}$, respectively; in both cases $\varepsilon_p = 10^{-5}$. Moreover, we provide a sliced view of the geometry to show that the element sizes vary considerably across all the subdomains (and even between neighboring ones). This is to stress once again that the coupling method presented in Chapter 2 allows us to employ drastically different mesh sizes and multiple discretization methods, but at the same time to obtain a smooth global velocity field—the one in the figure corresponds to RB2—even with a limited number of Lagrange multiplier basis functions ($N_\lambda^\delta = 63$ per interface, for a total of 945 basis functions). We remark that the POD tolerance does not play a significant role in the flow rate at each outlet. This is due to the fact that the flow repartition across the different arteries is, in this problem, performed by the bifurcations, which are here modeled by the FE method and hence independent of the POD tolerance. Had the boundary conditions at the outlets been more physiological, the flow repartition would have been less dependent on the geometry of the bifurcations.

Remark 5.6. We recall that the kinematic condition of the hybrid method discussed in Chapter 2 and appearing in W4.1 at the continuous level reads

$$\int_{\Gamma^{[jm]}} \boldsymbol{\eta} \cdot (\mathbf{u}_j - \mathbf{u}_m) = 0 \quad \forall \boldsymbol{\eta} \in \mathcal{L}^{[jm]}.$$

Choosing $\boldsymbol{\eta} = \mathbf{n}$, where \mathbf{n} is the normal to the interface $\Gamma^{[jm]}$, is equivalent to imposing the equilibrium of the flow rate across $\Gamma^{[jm]}$. Since the inclusion of the constant function in the set of basis functions for the Lagrange multiplier is a necessary condition to the well-posedness of the discrete problem, the flow rate is a priori conserved in the discrete

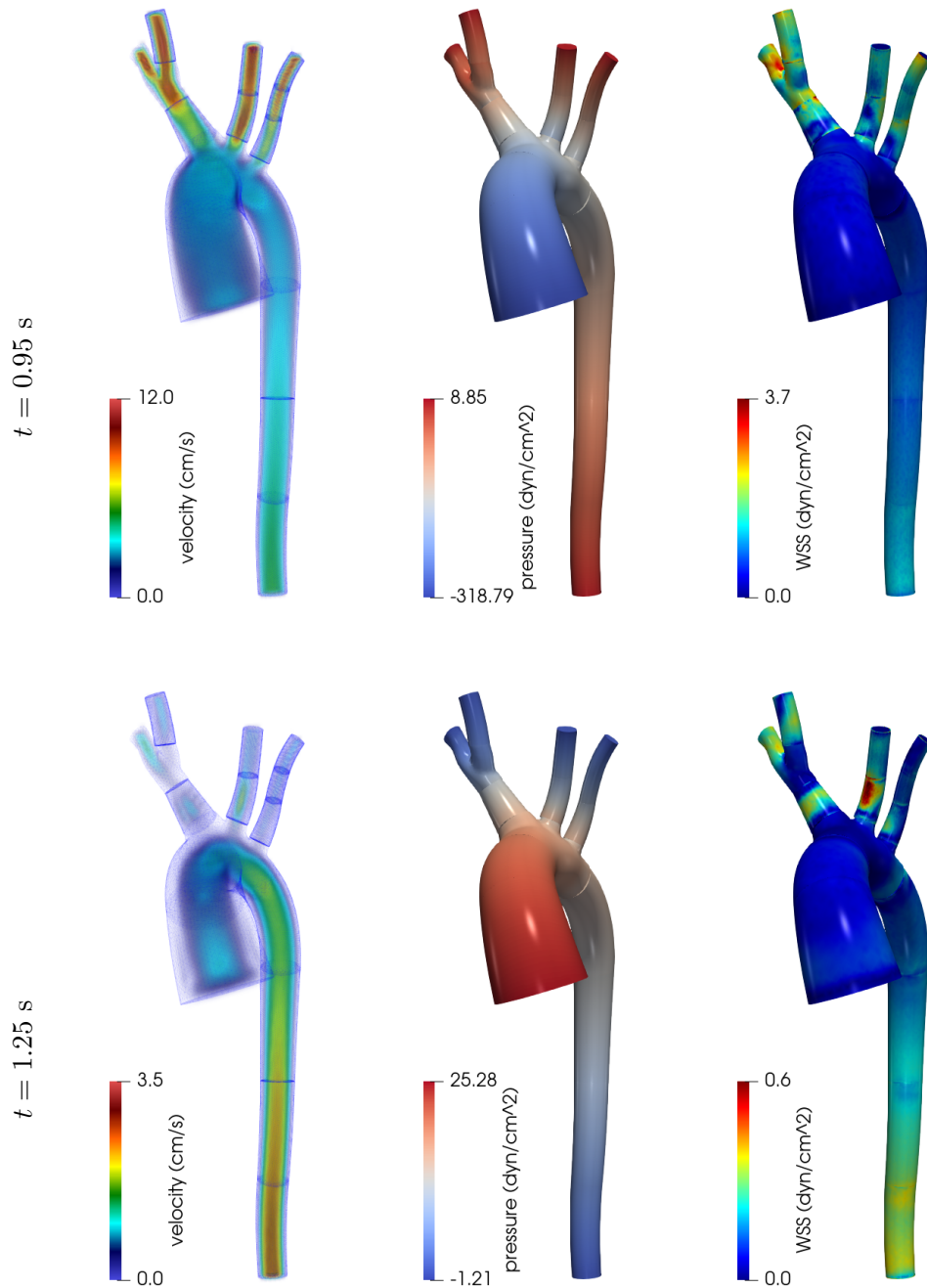


Figure 5.10 – Velocity magnitude (left), pressure (center) and WSS on the vessel boundary (right), at $t = 0.95$ s and $t = 1.25$ s (top and bottom row, respectively). The results are obtained with POD tolerances $\varepsilon_u = 10^{-3}$ and $\varepsilon_p = 10^{-5}$.

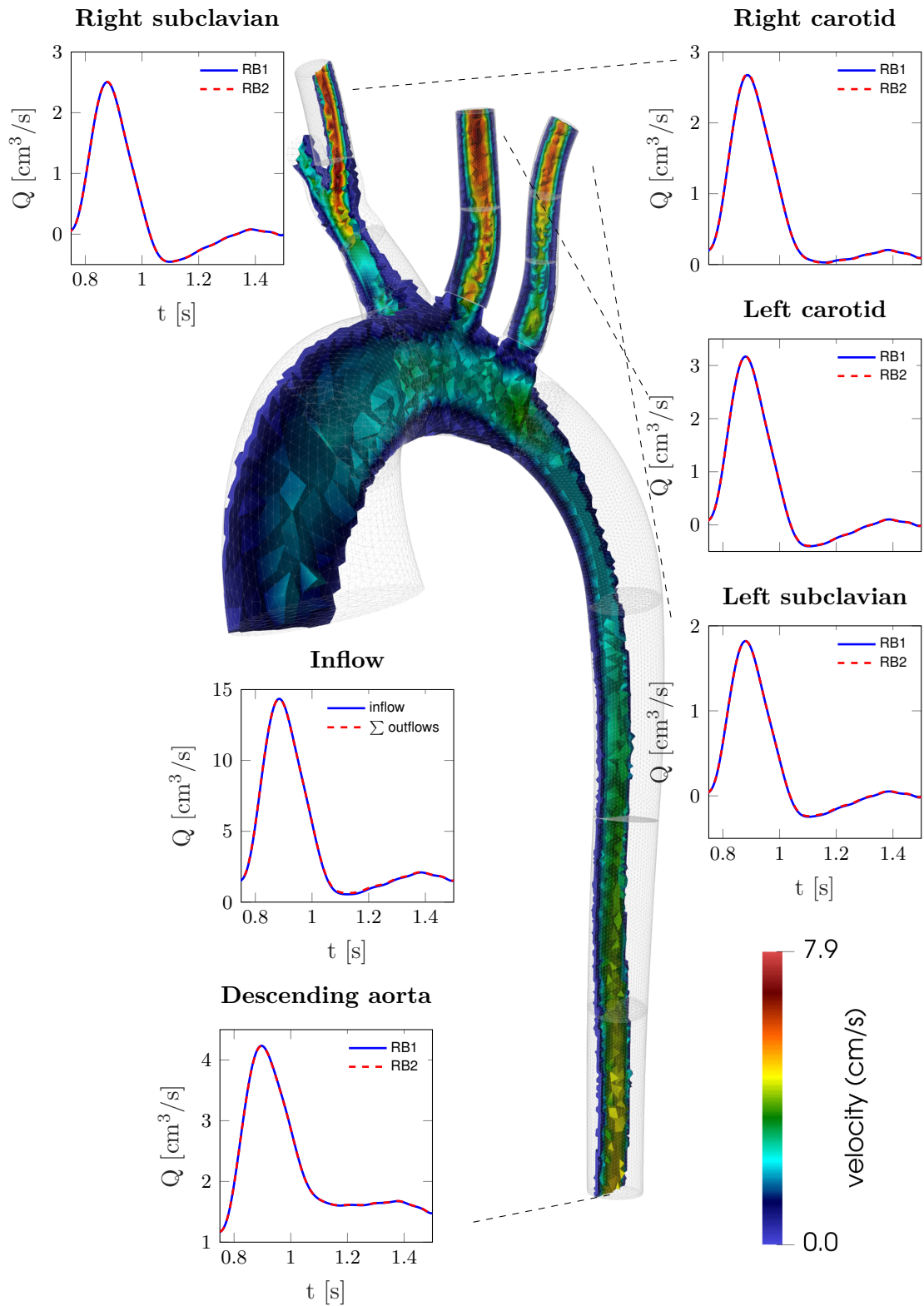


Figure 5.11 – Flow rates at the inlet and outlets, and velocity field magnitude at $t = 1$ s across several slices of the geometry. RB1: $\varepsilon_u = 10^{-2}$. RB2: $\varepsilon_u = 10^{-3}$.

setting. This is true, however, if the interface $\Gamma^{[jm]}$ is the same for both neighboring subdomains. In this particular numerical example, we have already discussed how the interfaces are non-matching because of the presence of the FE blocks (which do not feature circular interfaces). The preservation of the flow rate is, therefore, a question that needs to be addressed in case of imperfect geometrical approximation. In the inflow profile plot in Fig. 5.11 we show, together with the imposed one, the sum of the flow rates over all the outlets obtained with RB2 (the same function obtained with RB1 is nearly identical and therefore it is not shown here). The difference between the two curves is not significant. From a quantitative point of view, over the whole time period $(0, T = 1.5)$ we register the relative difference of the flow rate

$$\frac{|\int_0^T Q_{\text{out}}(t) - \int_0^T Q_{\text{in}}(t)|}{|\int_0^T Q_{\text{in}}(t)|} = 0.9\%,$$

where $Q_{\text{in}}(t)$ is the imposed flow rate at time t and $Q_{\text{out}}(t)$ is obtained by summing the contributions over all the outlets. As mentioned, this discrepancy is likely to be caused by the geometrical approximation and can be decreased by devising more sophisticated methods to map reference circular interfaces onto the physical ones.

5.5 Concluding remarks

In this chapter, we demonstrated the capabilities of the method formalized in Chapter 4. We employed the same geometry used for the offline phase and two physiological geometries: one consisting of an aorta with the two iliac arteries and the other representing the aortic arch with carotid and subclavian arteries. In the first case, we observed considerable speedups—from 12 to 33 over the total runtime and from 29 to 50 over the sole solve phase—with respect to the full order solution. A considerable gain in performance has also been achieved in the second case for some choices of POD tolerances, although the fact that we considered a reference solution obtained with a different solver (i.e., SimVascular) made the comparison in terms of runtime more complex. In both applications, we also analyzed the performance in the WSS reconstruction, which is possible in our ROM—as opposed, for example, to geometrical multiscale methods—because the three-dimensional nature of the flow problem is preserved. Our goal with the last numerical experiment was to demonstrate the feasibility of modeling a small subset of the subdomains with the FE method. We showed that even in this scenario employing RB solutions in some blocks allows us to reduce the size of the system, which in our test was largely composed of FE dofs. We also verified that the velocity field was qualitatively smooth and that the conservation of mass is approximately satisfied. However, we observed how the geometrical approximation of the interfaces caused noticeable discontinuities in the pressure and the WSS. This indicates that, to make this hybrid FE-RB approach a viable option in realistic scenarios, a more involved mapping of the parametrized interfaces—that in our case are circular for every choice of the geometrical parameters—needs to be devised.

6 Conclusions

In this thesis, we described and validated a numerical method for the MOR of cardiovascular flow. The main idea of the technique is to consider a geometrical discretization of the geometry of interest into elementary subdomains, for example, tubes and bifurcations, which are obtained as deformations of reference building blocks. Each of these building blocks is equipped with a set of RB functions generated during a demanding offline phase. In this phase, a large number of snapshots are sampled by solving flow problems with the FE method on geometries decomposed by the same approach mentioned above. The reduced basis is then generated by applying POD to the set of snapshots. In the online phase of the reduced order method, the local solutions in the subdomains are found as a linear combination of the RB functions.

A significant part of the algorithm is the coupling strategy. In the thesis, we presented a nonconforming method based on the theory of primal hybrid methods in which the coupling (kinematic and dynamic) conditions give rise to a global system featuring a saddle-point structure. We defined a set of spectral basis functions on each interface, and we used these to approximate the corresponding Lagrange multiplier. Our numerical experiments demonstrated that this approach allows us to retain the expected h -convergence rate of the primal discretization method by enlarging the system with a small number of additional dofs (namely, those associated with the Lagrange multipliers). The method is also flexible in that it allowed us to couple different discretization strategies—FE, IGA, and RB methods—and different PDEs in a fluid-structure interaction benchmark.

The fact that the global system presents a saddle-point structure requires particular attention, as this class of problems is classically associated with stability issues. Specifically, it is crucial to verify that the inf-sup condition is satisfied in a discrete setting. While, in the FE context, we verified only numerically that the inf-sup condition is generally verified by considering a small number of Lagrange multiplier basis functions (which, however, is typically also sufficient to obtain optimal convergence rates in h) when using the RB method we resorted to ad-hoc supremizers. These are functions specifically

computed to satisfy the inf-sup condition and are used to enrich the reduced basis. Since we were dealing with two saddle-point problems at the same time (one involving the incompressibility constraint in the Navier–Stokes equations and one related to the coupling conditions), the stability of the global problem is recovered by enriching the velocity reduced basis with two different sets of supremizers.

In the numerical results in Chapter 5, we addressed questions relative to the effects of the RB and geometrical approximations, as well as to the expected gain in performance with respect to the high-fidelity models. We showed that the results obtained with our ROM were close to the ones relative to the FE method in terms of velocity, pressure, and WSS and that, in some cases, it is reasonable to expect speedups of the order of one order of magnitude (or two, if some of the bottlenecks of the algorithm, such as the computation of the nonlinear term, are carefully optimized). We also observed that the geometrical approximation plays an essential role in the approximation of specific quantities such as the WSS.

We believe that the results reported in this thesis show great promise for the proposed ROM to be successfully applied in realistic scenarios. Compared to other reduction techniques for cardiovascular simulations such as 0D and 1D models, the combined use of the RB method and Domain Decomposition has the advantage to preserve important three-dimensional features of the flow and to allow for a local approximation of the WSS distribution on the vessel boundary. Nevertheless, this work suggests several areas of improvement that are addressed in more detail in the next section.

6.1 Perspectives and future work

The work carried out in this thesis could be extended in several directions. These include:

- *Stabilization of the coupling problem.* As mentioned, when using our nonconforming method, the inf-sup condition in the FE context does not typically pose problems, as the number of Lagrange multiplier basis functions necessary to achieve h -convergence is usually smaller than the one for which the stability is lost. We believe, nonetheless, that a viable way to ensure the well-posedness of the discrete problem for any refinement level of the Lagrange multiplier space is through the introduction of a stabilization term in the kinetic condition of the coupled problem. As a matter of fact, stabilization approaches such as the ones presented in Section 1.5 aim at modifying the saddle-point structure of the system such that the discretized global matrix features nonzero matrices in the diagonal blocks corresponding to the Lagrange multipliers dofs. This is sometimes achieved by adding terms depending on the residual of the equations in strong form, as in the VMS-SUPG approach. In the case of the coupling, we believe that a possible candidate for such a stabilization term could be derived by exploiting the fact that, at the continuous level, the

difference between the Lagrange multiplier and the stress at the interface is zero.

- *Parallel implementation of the saddle-point preconditioner.* In Section 4.3, we presented a preconditioner for the global problem which takes advantage of its saddle-point structure. We also described how the computation of the Schur complement depends on operations which are mainly local to the subdomains. Hence, an exciting development of our study is the investigation of the performance of a parallel preconditioner in which the subdomains are mapped to separate cores (see Remark 4.5).
- *Use of reduced fluid-structure interaction models in modular geometries.* A standard reduced model for fluid-structure interaction in cardiovascular applications considers a vessel wall modeled as a membrane (as discussed in Section 1.6). Implementing this model in our framework for fluid simulations in modular geometries would significantly improve the accuracy of the numerical approximation in physiological scenarios. Moreover, the popular 1D models are based on the assumption of deformable vessel walls, and therefore a direct comparison between them and our ROM makes sense only when fluid-structure interaction is taken into account.
- *Validation of the ROM in cases with physiological boundary conditions.* We noted in Remark 5.3 that the boundary conditions employed in this thesis are not physiological. In future applications of our method to clinical cases, it will be necessary to consider more complex nonlinear boundary conditions (e.g., the three-element Windkessel model), which complicate the resolution of the problem, e.g., in the computation of the Jacobian of the nonlinear system.
- *Optimization of the current bottlenecks of the Reduced Order Method.* In this thesis, the setup of the ROM is expensive because it is performed by projecting the high-fidelity matrices onto the reduced spaces. This also applies to the computation of the nonlinear term of the Navier–Stokes equations (which is done in each iteration of the Newton–Raphson algorithm). An approach based on interpolation methods such as DEIM or its matrix variant (see Section 3.2.2) will greatly improve the efficiency of the ROM and will be considered in the future.

Bibliography

- [AHS11] Kendall Atkinson, Weimin Han, and David E. Stewart. *Numerical solution of ordinary differential equations*, volume 108. John Wiley & Sons, 2011.
- [AP16] Sanjib Kumar Acharya and Ajit Patel. Primal hybrid method for parabolic problems. *Applied Numerical Mathematics*, 108:102–115, 2016.
- [ASB78] Bo O. Almroth, Perry Stern, and Frank A. Brogan. Automatic choice of global shape functions in structural analysis. *Aiaa Journal*, 16(5):525–528, 1978.
- [Bab73a] Ivo Babuška. The finite element method with lagrangian multipliers. *Numerische Mathematik*, 20(3):179–192, 1973.
- [Bab73b] Ivo Babuška. The finite element method with penalty. *Mathematics of computation*, 27(122):221–228, 1973.
- [Bal96] Etienne Balmes. Parametric families of reduced finite element models. theory and applications. *Mechanical Systems and Signal Processing*, 10(4):381–394, 1996.
- [BB88] Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.
- [BB90] Franco Brezzi and Klaus-Jürgen Bathe. A discourse on the stability conditions for mixed finite element formulations. *Computer methods in applied mechanics and engineering*, 82(1-3):27–57, 1990.
- [BBB⁺16] David S. Blom, Philipp Birken, Hester Bijl, Fleur Kessels, Andreas Meister, and Alexander H. van Zuijlen. A comparison of Rosenbrock and ESDIRK methods combined with iterative solvers for unsteady compressible flows. *Advances in Computational Mathematics*, 42(6):1401–1426, 2016.
- [BBC⁺14] Gang Bao, Yuri Bazilevs, Jae-Hyun Chung, Paolo Decuzzi, Horacio D. Espinosa, Mauro Ferrari, Huajian Gao, Shaolie S. Hossain, Thomas J. R. Hughes, Roger D. Kamm, et al. USNCTAM perspectives on mechanics in medicine. *Journal of The Royal Society Interface*, 11(97):20140301, 2014.

Bibliography

- [BBC⁺15] Francesco Bassi, Lorenzo Botti, Alessandro Colombo, Antonio Ghidoni, and Francesco Massa. Linearly implicit Rosenbrock-type Runge–Kutta schemes applied to the Discontinuous Galerkin solution of compressible and incompressible unsteady flows. *Computers & Fluids*, 118:305–320, 2015.
- [BBC⁺16] Francesco Bassi, Lorenzo Botti, Alessandro Colombo, Andrea Crivellini, Antonio Ghidoni, and Francesco Massa. On the development of an implicit high-order discontinuous Galerkin method for DNS and implicit LES of turbulent flows. *European Journal of Mechanics-B/Fluids*, 55:367–379, 2016.
- [BBF13] Daniele Boffi, Franco Brezzi, and Michel Fortin. *Mixed finite element methods and applications*, volume 44. Springer, 2013.
- [BBM⁺18] Pablo J. Blanco, Carlos A. Bulant, Lucas O. Müller, G. D. Maso Talou, C. Guedes Bezerra, Pedro A. Lemos, and Raúl A. Feijóo. Comparison of 1D and 3D models for the estimation of fractional flow reserve. *Scientific reports*, 8(1):1–12, 2018.
- [BBWW15] Ericka Brivadis, Annalisa Buffa, Barbara Wohlmuth, and Linus Wunderlich. Isogeometric mortar methods. *Computer Methods in Applied Mechanics and Engineering*, 284:292–319, 2015.
- [BCC⁺07] Yuri Bazilevs, Victor M. Calo, J. Austin Cottrell, Thomas J. R. Hughes, Alessandro Reali, and Guglielmo Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer methods in applied mechanics and engineering*, 197(1-4):173–201, 2007.
- [BCD⁺11] Peter Binev, Albert Cohen, Wolfgang Dahmen, Ronald DeVore, Guergana Petrova, and Przemyslaw Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM journal on mathematical analysis*, 43(3):1457–1472, 2011.
- [BCS03] Faker B. Belgacem, Lawrence K. Chilton, and Padmanabhan Seshaiyer. The hp-mortar finite-element method for the mixed elasticity and stokes problems. *Computers & Mathematics with Applications*, 46(1):35–55, 2003.
- [BDF⁺17] Luca Bertagna, Simone Deparis, Luca Formaggia, Davide Forti, and Alessandro Veneziani. The LifeV library: engineering mathematics beyond the proof of concept. *arXiv preprint arXiv:1710.06596*, 2017.
- [BDFS11] Annalisa Buffa, Carlo De Falco, and Giancarlo Sangalli. Isogeometric analysis: stable elements for the 2D Stokes equation. *International Journal for Numerical Methods in Fluids*, 65(11-12):1407–1422, 2011.

-
- [BDW99] Dietrich Braess, Wolfgang Dahmen, and Christian Wieners. A multigrid algorithm for the mortar finite element method. *SIAM Journal on Numerical Analysis*, 37(1):48–69, 1999.
- [Bel99] Faker B. Belgacem. The mortar finite element method with Lagrange multipliers. *Numerische Mathematik*, 84(2):173–197, 1999.
- [Ber89] Christine Bernardi. A new nonconforming approach to domain decomposition: the mortar element method. *Nonlinear partial equations and their applications*, 1989.
- [BFI⁺16] Francesco Ballarin, Elena Faggiano, Sonia Ippolito, Andrea Manzoni, Alfio Quarteroni, Gianluigi Rozza, and Roberto Scrofani. Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD–Galerkin method and a vascular shape parametrization. *Journal of Computational Physics*, 315:609–628, 2016.
- [BHS03] Roland Becker, Peter Hansbo, and Rolf Stenberg. A finite element method for domain decomposition with non-matching grids. *ESAIM: Mathematical Modelling and Numerical Analysis*, 37(2):209–225, 2003.
- [BM94] Franco Brezzi and Donatella Marini. A three-field domain decomposition method. *Contemporary Mathematics*, 157:27–34, 1994.
- [BM01] Franco Brezzi and Donatella Marini. Error estimates for the three-field formulation with bubble stabilization. *Mathematics of computation*, 70(235):911–934, 2001.
- [BMNP04] Maxime Barrault, Yvon Maday, Ngoc C. Nguyen, and Anthony T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *Comptes Rendus Mathématique Académie des Sciences Paris*, 339(9):667–672, 2004.
- [BMQR15] Francesco Ballarin, Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Supremizer stabilization of POD–Galerkin approximation of parametrized steady incompressible Navier–Stokes equations. *International Journal for Numerical Methods in Engineering*, 102(5):1136–1161, 2015.
- [BMR05] Christine Bernardi, Yvon Maday, and Francesca Rapetti. Basics and some applications of the mortar element method. *GAMM-Mitteilungen*, 28(2):97–123, 2005.
- [BMT12] Evren Bayraktar, Otto Mierka, and Stefan Turek. Benchmark computations of 3D laminar flow around a cylinder with CFX, OpenFOAM and FeatFlow. *International Journal of Computational Science and Engineering*, 7(3):253–266, 2012.

Bibliography

- [Bof94] Daniele Boffi. Stability of higher order triangular Hood-Taylor methods for the stationary Stokes equations. *Mathematical Models and Methods in Applied Sciences*, 4(02):223–235, 1994.
- [Bof97] Daniele Boffi. Three-dimensional finite element methods for the Stokes problem. *SIAM Journal on Numerical Analysis*, 34(2):664–670, 1997.
- [BP79] Michel Bercovier and Olivier Pironneau. Error estimates for finite element method solution of the Stokes problem in the primitive variables. *Numerische Mathematik*, 33(2):211–224, 1979.
- [BPX90] James H. Bramble, Joseph E. Pasciak, and Jinchao Xu. Parallel multilevel preconditioners. *Mathematics of Computation*, 55(191):1–22, 1990.
- [Bre74] Franco Brezzi. On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 8(R2):129–151, 1974.
- [CBA17] Kevin Carlberg, Matthew Barone, and Harbir Antil. Galerkin v. least-squares Petrov–Galerkin projection in nonlinear model reduction. *Journal of Computational Physics*, 330:693–734, 2017.
- [CDF17] Claudia Maria Colciago, Simone Deparis, and Davide Forti. Fluid-structure interaction for vascular flows: from supercomputers to laptops. Technical report, Walter De Gruyter, 2017.
- [CDFQ11] Paolo Crosetto, Simone Deparis, Gilles Fourestey, and Alfio Quarteroni. Parallel algorithms for fluid-structure interaction problems in haemodynamics. *SIAM Journal on Scientific Computing*, 33(4):1598–1622, 2011.
- [CDQ14] Claudia Maria Colciago, Simone Deparis, and Alfio Quarteroni. Comparisons between reduced order models and full 3D models for fluid–structure interaction problems in haemodynamics. *Journal of Computational and Applied Mathematics*, 265:120–138, 2014.
- [Céa64] Jean Céa. Approximation variationnelle des problèmes aux limites. *Ann. Inst. Fourier (Grenoble)*, 14(fasc. 2):345–444, 1964.
- [CG05] Kristopher S. Cunningham and Avrum I. Gotlieb. The role of shear stress in the pathogenesis of atherosclerosis. *Laboratory investigation*, 85(1):9–23, 2005.
- [CHB09] J. Austin Cottrell, Thomas J. R. Hughes, and Yuri Bazilevs. *Isogeometric analysis: toward integration of CAD and FEA*. John Wiley & Sons, 2009.

-
- [CHR07] J. Austin Cottrell, Thomas J. R. Hughes, and Alessandro Reali. Studies of refinement and continuity in isogeometric structural analysis. *Computer methods in applied mechanics and engineering*, 196(41-44):4160–4183, 2007.
- [Cia88] Philippe G. Ciarlet. *Mathematical Elasticity: Volume I: three-dimensional elasticity*. North-Holland, 1988.
- [Cia02] Philippe G. Ciarlet. *The finite element method for elliptic problems*. SIAM, 2002.
- [CK06] François E. Cellier and Ernesto Kofman. *Continuous system simulation*. Springer Science & Business Media, 2006.
- [CLB20] Luca Coradello, Gabriele Loli, and Annalisa Buffa. A projected superpenalty method for the c^1 -coupling of multi-patch isogeometric kirchhoff plates. *arXiv preprint arXiv:2007.14343*, 2020.
- [Col14] Claudia Maria Colciago. *Reduced order fluid-structure interaction models for haemodynamics applications*. PhD thesis, EPFL, 2014.
- [CQR13] Peng Chen, Alfio Quarteroni, and Gianluigi Rozza. Simulation-based uncertainty quantification of human arterial network hemodynamics. *International journal for numerical methods in biomedical engineering*, 29(6):698–721, 2013.
- [Cro11] Paolo Crosetto. *Fluid-structure interaction problems in hemodynamics: parallel solvers, preconditioners, and applications*. PhD thesis, PhD thesis, EPFL, 2011.
- [CS10] Saifon Chaturantabut and Danny C. Sorensen. Nonlinear model reduction via discrete empirical interpolation. *SIAM Journal on Scientific Computing*, 32(5):2737–2764, 2010.
- [DB72] Carl De Boor. On calculating with B-splines. *Journal of Approximation theory*, 6(1):50–62, 1972.
- [DDFQ06] Simone Deparis, Marco Discacciati, Gilles Fourestey, and Alfio Quarteroni. Fluid-structure algorithms based on Steklov-Poincaré operators. *Computer Methods in Applied Mechanics and Engineering*, 195(41-43):5797–5812, 2006.
- [DDM⁺19] Simone Deparis, Michel O. Deville, Filippo Menghini, Luca Pegolotti, and Alfio Quarteroni. Application of the Rosenbrock methods to the solution of unsteady 3D incompressible Navier-Stokes equations. *Computers & Fluids*, 179:112–122, 2019.
- [DDQ06] Simone Deparis, Marco Discacciati, and Alfio Quarteroni. A domain decomposition framework for fluid-structure interaction problems. *Computational Fluid Dynamics 2004*, pages 41–58, 2006.

Bibliography

- [DFGQ16] Simone Deparis, Davide Forti, Paola Gervasio, and Alfio Quarteroni. INTERNODES: an accurate interpolation-based method for coupling the Galerkin solutions of PDEs on subdomains featuring non-conforming interfaces. *Computers & Fluids*, 141:22–41, 2016.
- [DFQ14] Simone Deparis, Davide Forti, and Alfio Quarteroni. A rescaled localized radial basis function interpolation on non-cartesian and nonconforming grids. *SIAM Journal on Scientific Computing*, 36(6):A2745–A2762, 2014.
- [DIP19] Simone Deparis, Antonio Iubatti, and Luca Pegolotti. Coupling non-conforming discretizations of PDEs by spectral approximation of the Lagrange multiplier space. *ESAIM: Mathematical Modelling and Numerical Analysis*, 53(5):1667–1694, 2019.
- [DSDMQ19] Niccolò Dal Santo, Simone Deparis, Andrea Manzoni, and Alfio Quarteroni. An algebraic least squares reduced basis method for the solution of nonaffinely parametrized stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 344:186–208, 2019.
- [DSM18] Niccolò Dal Santo and Andrea Manzoni. Hyper-reduced order models for parametrized unsteady Navier-Stokes equations on domains with variable shape. 2018.
- [DX14] Charles F. Dunkl and Yuan Xu. *Orthogonal polynomials of several variables*. Number 155. Cambridge University Press, 2014.
- [Edm71] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical programming*, 1(1):127–136, 1971.
- [EPGW14] Andreas Ehrl, Alexander Popp, Volker Gravemeier, and Wolfgang A. Wall. A dual mortar approach for mesh tying within a variational multiscale method for incompressible flow. *International Journal for Numerical Methods in Fluids*, 76(1):1–27, 2014.
- [FBTH09] C. Alberto Figueroa, Seungik Baek, Charles A. Taylor, and Jay D. Humphrey. A computational framework for fluid–solid-growth modeling in cardiovascular simulations. *Computer methods in applied mechanics and engineering*, 198(45-46):3583–3602, 2009.
- [FD15] Davide Forti and Luca Dedè. Semi-implicit BDF time discretization of the Navier–Stokes equations with VMS-LES modeling in a high performance computing framework. *Computers & Fluids*, 117:168–182, 2015.
- [For16] Davide Forti. *Parallel algorithms for the solution of large-scale fluid-structure interaction problems in hemodynamics*. PhD thesis, EPFL, 2016.

- [FR83] J. P. Fink and Werner C. Rheinboldt. On the error behavior of the reduced basis technique for nonlinear finite element approximations. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*, 63(1):21–28, 1983.
- [FTM17] C. Alberto Figueroa, Charles A. Taylor, and Alison L. Marsden. *Blood Flow*, pages 1–31. American Cancer Society, 2017.
- [FVCJ⁺06] C. Alberto Figueroa, Irene E. Vignon-Clementel, Kenneth E. Jansen, Thomas J. R. Hughes, and Charles A. Taylor. A coupled momentum method for modeling blood flow in three-dimensional deformable arteries. *Computer methods in applied mechanics and engineering*, 195(41-43):5685–5706, 2006.
- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [GH73] William J Gordon and Charles A Hall. Transfinite element methods: blending-function interpolation over arbitrary curved element domains. *Numerische Mathematik*, 21(2):109–129, 1973.
- [GLS88] Kjell Gustafsson, Michael Lundh, and Gustaf Söderlind. API stepsize control for the numerical solution of ordinary differential equations. *BIT Numerical Mathematics*, 28(2):270–287, 1988.
- [GMAB⁺20] Sofia Guzzetti, Luis A. Mansilla Alvarez, Pablo J. Blanco, Kevin Carlberg, and Alessandro Veneziani. Propagating uncertainties in large-scale hemodynamics models via network uncertainty quantification and reduced-order modeling. *Computer Methods in Applied Mechanics and Engineering*, 358:112626, 2020.
- [GPV18] Sofia Guzzetti, Perotto Perotto, and Alessandro Veneziani. Hierarchical model reduction for incompressible fluids in pipes. *International Journal for Numerical Methods in Engineering*, 114(5):469–500, 2018.
- [GQ16] Paola Gervasio and Alfio Quarteroni. Analysis of the internodes method for non-conforming discretizations of elliptic equations. *MATHICSE report*, 2016.
- [GR12] Vivette Girault and Pierre A. Raviart. *Finite element methods for Navier-Stokes equations: theory and algorithms*, volume 5. Springer Science & Business Media, 2012.
- [GVL12] Gene H. Golub and Charles F. Van Loan. *Matrix computations*, volume 3. JHU press, 2012.

Bibliography

- [HCB05] Thomas J. R. Hughes, J. Austin Cottrell, and Yuri Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. *Computer methods in applied mechanics and engineering*, 194(39-41):4135–4195, 2005.
- [HGC⁺14] Christian Hesch, Antonio J. Gil, A. Arranz Carreño, Javier Bonet, and Peter Betsch. A mortar approach for fluid–structure interaction problems: Immersed strategies for deformable and rigid bodies. *Computer Methods in Applied Mechanics and Engineering*, 278:853–882, 2014.
- [HKP13] Dinh B. P. Huynh, David J. Knezevic, and Anthony T. Patera. A static condensation reduced basis element method: approximation and a posteriori error estimation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 47(1):213–251, 2013.
- [HRS16] Jan S. Hesthaven, Gianluigi Rozza, and Benjamin Stamm. *Certified reduced basis methods for parametrized partial differential equations*. Springer, 2016.
- [HSF18] Thomas J. R. Hughes, Guglielmo Scovazzi, and Leopoldo P. Franca. Multi-scale and stabilized methods. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–64, 2018.
- [HSZ14] Jan S. Hesthaven, Benjamin Stamm, and Shun Zhang. Efficient greedy algorithms for high-dimensional parameter spaces with applications to empirical interpolation and reduced basis methods. *ESAIM: Mathematical Modelling and Numerical Analysis*, 48(1):259–283, 2014.
- [HT74] P. Hood and C. Taylor. Navier-Stokes equations using mixed interpolation. *Finite element methods in flow problems*, pages 121–132, 1974.
- [HU18] Jan S. Hesthaven and Stefano Ubbiali. Non-intrusive reduced order modeling of nonlinear problems using neural networks. *Journal of Computational Physics*, 363:55–78, 2018.
- [Hug12] Thomas J. R. Hughes. *The finite element method: linear static and dynamic finite element analysis*. Courier Corporation, 2012.
- [HW10] Ernst Hairer and Gerhard Wanner. *Solving ordinary differential equations. II*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2010.
- [Iap12] Laura Iapichino. *Reduced basis methods for the solution of parametrized PDEs in repetitive and complex networks with application to CFD*. PhD thesis, 2012.
- [IQR12] Laura Iapichino, Alfio Quarteroni, and Gianluigi Rozza. A reduced basis hybrid method for the coupling of parametrized domains represented by

-
- fluidic networks. *Computer Methods in Applied Mechanics and Engineering*, 221:63–82, 2012.
- [IRQ10] Laura Iapichino, Gianluigi Rozza, and Alfio Quarteroni. Reduced basis (element) methods for the study of parametrized cardiovascular geometries and networks. 2010.
- [IVA93] Moshe Israeli, Lev P. Vozovoi, and Amir Z. Averbuch. Domain decomposition methods for solving parabolic PDEs on multiprocessors. *Applied numerical mathematics*, 12(1):193–212, 1993.
- [JIR14] Christoph Jäggli, Laura Iapichino, and Gianluigi Rozza. An improvement on geometrical parameterizations by transfinite maps. *Comptes Rendus Mathématique*, 352(3):263–268, 2014.
- [JMR06] Volker John, Gunar Matthies, and Joachim Rang. A comparison of time-discretization/linearization approaches for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 195(44):5995–6010, 2006.
- [Joh06] Volker John. On the efficiency of linearization schemes and coupled multigrid methods in the simulation of a 3D flow around a cylinder. *International Journal for Numerical Methods in Fluids*, 50(7):845–862, 2006.
- [JR10] Volker John and Joachim Rang. Adaptive time step control for the incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 199(9):514–524, 2010.
- [KN03] Andrew V. Knyazev and Klaus Neymeyr. Efficient solution of symmetric eigenvalue problems using multigrid preconditioners in the locally optimal block conjugate gradient method. *Electronic Transactions on Numerical Analysis*, 15:38–55, 2003.
- [KPKW11] Thomas Klöppel, Alexander Popp, Ulrich Küttler, and Wolfgang A. Wall. Fluid-structure interaction for non-conforming interfaces based on a dual mortar formulation. *Computer Methods in Applied Mechanics and Engineering*, 200(45-46):3111–3126, 2011.
- [KSN⁺19] Efthymios N. Karatzas, Giovanni Stabile, Leo Nouveau, Guglielmo Scovazzi, and Gianluigi Rozza. A reduced basis approach for pdes on parametrized geometries based on the shifted boundary finite element method and application to a stokes flow. *Computer Methods in Applied Mechanics and Engineering*, 347:568–587, 2019.
- [KV01] Karl Kunisch and Stefan Volkwein. Galerkin proper orthogonal decomposition methods for parabolic problems. *Numerische mathematik*, 90(1):117–148, 2001.

Bibliography

- [KV02] Karl Kunisch and Stefan Volkwein. Galerkin proper orthogonal decomposition methods for a general equation in fluid dynamics. *SIAM J. Numerical Analysis*, 40(2):492–515, 2002.
- [LMR06a] Emil Løvgrén, Yvon Maday, and Einar M. Rønquist. A reduced basis element method for complex flow systems. In *ECCOMAS CFD 2006: Proceedings of the European Conference on Computational Fluid Dynamics, Egmond aan Zee, The Netherlands, September 5-8, 2006*. Delft University of Technology; European Community on Computational Methods in Applied Sciences (ECCOMAS), 2006.
- [LMR06b] Emil Løvgrén, Yvon Maday, and Einar M. Rønquist. A reduced basis element method for the steady Stokes problem. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 40(3):529–552, 2006.
- [LMR06c] Emil Løvgrén, Yvon Maday, and Einar M. Rønquist. A reduced basis element method for the steady Stokes problem: Application to hierarchical flow systems. *Modeling, identification and control*, 27(2):79–94, 2006.
- [LMR07] Emil Løvgrén, Yvon Maday, and Einar M. Rønquist. The reduced basis element method for fluid flows. In *Analysis and simulation of fluid dynamics*, pages 129–154. Birkhäuser Basel, 2007.
- [LMR09] Emil Løvgrén, Yvon Maday, and Einar M. Rønquist. Global c1 maps on general domains. *Mathematical Models and Methods in Applied Sciences*, 19(05):803–832, 2009.
- [LR10] Toni Lassila and Gianluigi Rozza. Parametric free-form shape design with PDE models and reduced basis method. *Computer Methods in Applied Mechanics and Engineering*, 199(23-24):1583–1592, 2010.
- [LT08] Jens Lang and Delia Teleaga. Towards a fully space-time adaptive FEM for magnetoquasistatics. *IEEE Transactions on Magnetics*, 44(6):1238–1241, 2008.
- [LTM01] Patrick Le Tallec and Jean Mouro. Fluid structure interaction with large structural displacements. *Computer methods in applied mechanics and engineering*, 190(24-25):3039–3067, 2001.
- [LXLX16] Xiaodong Liu, Yidong Xia, Hong Luo, and Lijun Xuan. A comparative study of Rosenbrock-type and implicit Runge-Kutta time integration for discontinuous Galerkin method for unsteady 3D compressible Navier-Stokes equations. *Communications in Computational Physics*, 20(4):1016–1044, 2016.

- [LYDM20] Ju Liu, Weiguang Yang, Melody Dong, and Alison L. Marsden. The nested block preconditioning technique for the incompressible Navier–Stokes equations with emphasis on hemodynamic simulations. *Computer Methods in Applied Mechanics and Engineering*, 367:113122, 2020.
- [MABB⁺17] Luis A. Mansilla Alvarez, Pablo J. Blanco, Carlos A. Bulant, Eno Dari, Alessandro Veneziani, and Raul A. Feijóo. Transversally enriched pipe element method (tepem): An effective numerical approach for blood flow modeling. *International Journal for Numerical Methods in Biomedical Engineering*, 33(4), 2017.
- [MABBF19] Luis A. Mansilla Alvarez, Pablo J. Blanco, Carlos A. Bulant, and Raul A. Feijóo. Towards fast hemodynamic simulations in large-scale circulatory networks. *Computer Methods in Applied Mechanics and Engineering*, 344:734–765, 2019.
- [Mah20] Gabriel Maher. *Automated Segmentation and Uncertainty Quantification for Image-Based Cardiovascular Modeling with Convolutional Neural Networks*. PhD thesis, Stanford University, 2020.
- [Mal12] A. Cristiano I. Malossi. *Partitioned solution of geometrical multiscale problems for the cardiovascular system: models, algorithms, and applications*. PhD thesis, Ph. D. thesis, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2012.
- [MB13] A. Cristiano I. Malossi and Jean Bonnemain. Numerical comparison and calibration of geometrical multiscale models for the simulation of arterial flows. *Cardiovascular Engineering and Technology*, 4(4):440–463, 2013.
- [MBC⁺13] A. Cristiano I. Malossi, Pablo J. Blanco, Paolo Crosetto, Simone Deparis, and Alfio Quarteroni. Implicit coupling of one-dimensional and three-dimensional blood flow models with compliant vessels. *Multiscale Modeling & Simulation*, 11(2):474–506, 2013.
- [MEM15] Alison L. Marsden and Mahdi Esmaily-Moghadam. Multiscale modeling of cardiovascular flows for clinical decision support. *Applied Mechanics Reviews*, 67(3):030804, 2015.
- [MFSM20] Gabriel Maher, Casey Fleeter, Daniele Schiavazzi, and Alison L. Marsden. Geometric uncertainty in patient-specific cardiovascular modeling with convolutional dropout networks. *arXiv preprint arXiv:2009.07395*, 2020.
- [MNL⁺18] Francesco Massa, Gianmaria Noventa, Marco Lorini, Francesco Bassi, and Antonio Ghidoni. High-order linearly implicit two-step peer schemes for the discontinuous Galerkin solution of the incompressible Navier–Stokes equations. *Computers & Fluids*, 162:55–71, 2018.

Bibliography

- [Moo65] Gordon E. Moore. Cramming more components onto integrated circuits, 1965.
- [MQR12] Andrea Manzoni, Alfio Quarteroni, and Gianluigi Rozza. Shape optimization for viscous flows by reduced basis methods and free-form deformation. *International Journal for Numerical Methods in Fluids*, 70(5):646–670, 2012.
- [MR02] Yvon Maday and Einar M. Rønquist. A reduced-basis element method. *Comptes Rendus Mathématique*, 335(2):195–200, 2002.
- [MR04] Yvon Maday and Einar M. Rønquist. The reduced basis element method: application to a thermal fin problem. *SIAM Journal on Scientific Computing*, 26(1):240–258, 2004.
- [MS18] Alex Main and Guglielmo Scovazzi. The shifted boundary method for embedded domain computations. part i: Poisson and stokes problems. *Journal of Computational Physics*, 372:972–995, 2018.
- [MVCF⁺13] Mahdi E. Moghadam, Irene E. Vignon-Clementel, Richard Figliola, Alison L. Marsden, and Modeling of Congenital Hearts Alliance (MOCHA) Investigators. A modular numerical method for implicit 0D/3D coupling in cardiovascular finite element simulations. *Journal of Computational Physics*, 244:63–79, 2013.
- [MWM19] Gabriel Maher, Nathan Wilson, and Alison L. Marsden. Accelerating cardiovascular model building with convolutional neural networks. *Medical & biological engineering & computing*, 57(10):2319–2335, 2019.
- [Neg15] Federico Negri. *Efficient Reduction Techniques for the Simulation and Optimization of Parametrized Systems*. PhD thesis, EPFL, 2015.
- [New59] Nathan M. Newmark. A method of computation for structural dynamics. *Journal of the engineering mechanics division*, 85(3):67–94, 1959.
- [NMA15] Federico Negri, Andrea Manzoni, and David Amsallem. Efficient model reduction of parametrized systems by matrix discrete empirical interpolation. *Journal of Computational Physics*, 303:431–454, 2015.
- [Nob01] Fabio Nobile. *Numerical approximation of fluid-structure interaction problems with application to haemodynamics*. PhD thesis, EPFL, 2001.
- [NP83] Ahmed K. Noor and Jeanne M. Peters. Recent advances in reduction methods for instability analysis of structures. *Computers & Structures*, 16(1-4):67–80, 1983.
- [NW79] Syvert P. Nørsett and Arne Wolfbrandt. Order conditions for Rosenbrock type methods. *Numerische Mathematik*, 32(1):1–15, 1979.

-
- [PCVL⁺20] Martin Pfaller, Maria Cruz Varona, Jan Lang, Cristobal Bertoglio, and Wolfgang A. Wall. Using parametric model order reduction for inverse analysis of large nonlinear cardiac simulations. *International Journal for Numerical Methods in Biomedical Engineering*, 36(4):e3320, 2020.
- [PEV10] Simona Perotto, Alexandre Ern, and Alessandro Veneziani. Hierarchical local model reduction for elliptic problems: a domain decomposition approach. *Multiscale Modeling & Simulation*, 8(4):1102–1127, 2010.
- [PL04] Michael A. Puso and Tod A. Laursen. A mortar segment-to-segment contact method for large deformation solid mechanics. *Computer methods in applied mechanics and engineering*, 193(6-8):601–629, 2004.
- [Por85] TA Porsching. Estimation of the error in the reduced basis method solution of nonlinear equations. *Mathematics of Computation*, 45(172):487–496, 1985.
- [PPMD20] Luca Pegolotti, Martin Pfaller, Alison L. Marsden, and Simone Deparis. Model order reduction of flow based on a modular geometrical approximation of blood vessels. *arXiv preprint arXiv:2010.00285*, 2020.
- [Pus04] Michael A Puso. A 3D mortar method for solid mechanics. *International Journal for Numerical Methods in Engineering*, 59(3):315–336, 2004.
- [PW14] Alexander Popp and Wolfgang A. Wall. Dual mortar methods for computational contact mechanics—overview and recent developments. *GAMM-Mitteilungen*, 37(1):66–84, 2014.
- [QMN15] Alfio Quarteroni, Andrea Manzoni, and Federico Negri. *Reduced basis methods for partial differential equations: an introduction*, volume 92. Springer, 2015.
- [QSS10] Alfio Quarteroni, Riccardo Sacco, and Fausto Saleri. *Numerical mathematics*, volume 37. Springer Science & Business Media, 2010.
- [Qua14] Alfio Quarteroni. *Numerical models for differential problems*, volume 8. Springer-Verlag, 2014.
- [QV99] Alfio Quarteroni and Alberto Valli. *Domain decomposition methods for partial differential equations*. Numerical Mathematics and Scientific Computation. Oxford University Press, 1999.
- [QV08] Alfio Quarteroni and Alberto Valli. *Numerical approximation of partial differential equations*, volume 23. Springer Science & Business Media, 2008.
- [RA05] Joachim Rang and L. Angermann. New Rosenbrock W-methods of order 3 for partial differential algebraic equations of index 1. *BIT Numerical Mathematics*, 45(4):761–787, 2005.

Bibliography

- [Ran15] Joachim Rang. Improved traditional Rosenbrock–Wanner methods for stiff ODEs and DAEs. *Journal of Computational and Applied Mathematics*, 286:128–144, 2015.
- [Ran16] Joachim Rang. The Prothero and Robinson example: Convergence studies for Runge–Kutta and Rosenbrock–Wanner methods. *Applied Numerical Mathematics*, 108:37–56, 2016.
- [RM05] Lior Rokach and Oded Maimon. Decision trees. In *Data mining and knowledge discovery handbook*, pages 165–192. Springer, 2005.
- [Ros63] Howard H. Rosenbrock. Some general implicit processes for the numerical solution of differential equations. *The Computer Journal*, 5(4):329–330, 1963.
- [Roz05] Gianluigi Rozza. On optimization, control and shape design of an arterial bypass. *International Journal for Numerical Methods in Fluids*, 47(10–11):1411–1419, 2005.
- [RP03] Muruhan Rathinam and Linda R. Petzold. A new look at proper orthogonal decomposition. *SIAM Journal on Numerical Analysis*, 41(5):1893–1925, 2003.
- [RT77] Pierre A. Raviart and Jean M. Thomas. Primal hybrid finite element methods for 2nd order elliptic equations. *Mathematics of computation*, 31(138):391–413, 1977.
- [Ruh83] Axel Ruhe. Numerical aspects of Gram-Schmidt orthogonalization of vectors. *Linear algebra and its applications*, 52:591–601, 1983.
- [Saa93] Youcef Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.
- [Saa03] Yousef Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [SD00] Akram M. Shaaban and André J. Duerinckx. Wall shear stress and early atherosclerosis: a review. *American Journal of Roentgenology*, 174(6):1657–1665, 2000.
- [Ses03] Padmanabhan Seshaiyer. Stability and convergence of nonconforming hp finite-element methods. *Computers & Mathematics with Applications*, 46(1):165–182, 2003.
- [SF73] Gilbert Strang and George J. Fix. An analysis of the finite element method. 1973.
- [Si15] Hang Si. Tetgen, a delaunay-based quality tetrahedral mesh generator. *ACM Transactions on Mathematical Software (TOMS)*, 41(2):1–36, 2015.




- [SK12] Benjamin Sanderse and Barry Koren. Accuracy analysis of explicit Runge–Kutta methods applied to the incompressible Navier–Stokes equations. *Journal of Computational Physics*, 231(8):3041–3063, 2012.
- [SKCT16] Sethuraman Sankaran, Hyun Jin Kim, Gilwoo Choi, and Charles A. Taylor. Uncertainty quantification in coronary blood flow simulations: impact of geometry, boundary conditions and blood viscosity. *Journal of biomechanics*, 49(12):2540–2547, 2016.
- [SM11] Sethuraman Sankaran and Alison L. Marsden. A stochastic collocation method for uncertainty quantification and propagation in cardiovascular simulations. *Journal of biomechanical engineering*, 133(3), 2011.
- [SMK⁺12] Sethuraman Sankaran, Mahdi Esmaily Moghadam, Andrew M. Kahn, Elaine E. Tseng, Julius M. Guccione, and Alison L. Marsden. Patient-specific multiscale modeling of blood flow for coronary artery bypass graft surgery. *Annals of biomedical engineering*, 40(10):2228–2242, 2012.
- [Söd02] Gustaf Söderlind. Automatic control and adaptive time-stepping. *Numerical Algorithms*, 31(1-4):281–310, 2002.
- [SP16] Kathrin Smetana and Anthony T. Patera. Optimal local approximation spaces for component-based static condensation procedures. *SIAM Journal on Scientific Computing*, 38(5):A3318–A3356, 2016.
- [SR11] David Shirokoff and Rodolfo Ruben Rosales. An efficient method for the incompressible Navier–Stokes equations on irregular domains with no-slip boundary conditions, high order up to the boundary. *Journal of Computational Physics*, 230(23):8619–8646, 2011.
- [SRV10] Anthony Segal, Muhammad Rehman, and Cornelis Vuik. Preconditioners for incompressible Navier–Stokes solvers. *Numerical Mathematics: Theory, Methods and Applications*, 3(3):245–275, 2010.
- [SS76] Paul D. Stein and Hani N. Sabbah. Turbulent blood flow in the ascending aorta of humans with normal and diseased aortic valves. *Circulation research*, 39(1):58–65, 1976.
- [SS86] Youcef Saad and Martin H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869, 1986.
- [SS98] Padmanabhan Seshaiyer and Manil Suri. Convergence results for non-conforming hp methods: The mortar finite element method. *Contemporary Mathematics*, 218:453–459, 1998.

Bibliography

- [ST96] M. Schäref and S. Turek. Benchmark computations of laminar flow around a cylinder. *Flow Simulation with High-Performance Computers II*, 127:553–563, 1996.
- [Ste95] Gerd Steinebach. Order-reduction of ROW-methods for DAEs and method of lines applications. Preprint-Nr. 1741, FB Mathematik, TH Darmstadt, 1995.
- [TBI97] Lloyd N. Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.
- [TH73] Cedric Taylor and Paul Hood. A numerical solution of the Navier-Stokes equations using the finite element technique. *Computers & Fluids*, 1(1):73–100, 1973.
- [TR77] Jean M. Thomas and Pierre A. Raviart. A mixed finite element method for 2nd order elliptic problems. In *Mathematical aspects of finite element methods (Proc. Conf., Consiglio Naz. delle Ricerche (CNR), Rome, 1975)*, pages 292–315, 1977.
- [TW05] Andrea Toselli and Olof B. Widlund. *Domain decomposition methods: algorithms and theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer, 2005.
- [UWM⁺17] Adam Updegrove, Nathan M. Wilson, Jameson Merkow, Hongzhi Lan, Alison L. Marsden, and Shawn C. Shadden. Simvascular: an open source pipeline for cardiovascular simulation. *Annals of biomedical engineering*, 45(3):525–541, 2017.
- [Váz16] Rafael Vázquez. A new design for the implementation of isogeometric analysis in Octave and Matlab: GeoPDEs 3.0. *Computers & Mathematics with Applications*, 72(3):523–554, 2016.
- [Ver84] Rüdiger Verfürth. Error estimates for a mixed finite element approximation of the Stokes equations. *RAIRO. Analyse numérique*, 18(2):175–182, 1984.
- [VHK⁺15] Sylvain Vallaghé, Phuong Huynh, David J. Knezevic, Loi Nguyen, and Anthony T. Patera. Component-based reduced basis for parametrized symmetric eigenproblems. *Advanced Modeling and Simulation in Engineering Sciences*, 2(1):7, 2015.
- [VSBH99] Jan G. Verwer, Edwin J. Spee, Joke G. Blom, and Willem Hundsdorfer. A second-order Rosenbrock method applied to photochemical dispersion problems. *SIAM Journal on Scientific Computing*, 29(4):1456–1480, 1999.
- [WC14] Yuqi Wu and Xiao-Chuan Cai. A fully implicit domain decomposition based ALE framework for three-dimensional fluid–structure interaction

- with application in blood flow computation. *Journal of Computational Physics*, 258:524–537, 2014.
- [Woh00] Barbara Wohlmuth. A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM journal on numerical analysis*, 38(3):989–1012, 2000.
- [ZA98] Tulong Zhu and Satyanadham N. Atluri. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free galerkin method. *Computational Mechanics*, 21(3):211–222, 1998.

Luca Pegolotti

✉ pegolotti.luca@gmail.com ·  luca-pegolotti ·  11.05.1992 ·  Italian

EDUCATION

Swiss Federal Institute of Technology (EPFL), Lausanne 03.2017 - Present
PhD in Computational Mathematics

Thesis title: Reduction techniques for PDEs built upon Reduced Basis and Domain Decomposition Methods with applications to hemodynamics

Oral exam: 10.12.2020

Stanford University, California (USA) 01.2020 - 04.2020
Visiting Student Researcher

Program funded by a scholarship awarded by the Swiss National Science Foundation

Swiss Federal Institute of Technology (EPFL), Lausanne 09.2014 - 02.2017
MSc in Computational Science and Engineering (CSE)

Thesis title: Isogeometric Analysis of cardiac electrophysiology: application to the human atria
Final Grade: 5.5/6

Politecnico di Milano, Milan (Italy) 09.2011 - 09.2014
BSc in Mathematical Engineering

Final grade: 105/110

CORE EXPERIENCES

Mathematics Section, EPFL 03.2017 - 12.2019
Project co-supervisor

Participated actively as co-supervisor of two semester projects and one master project.

Mathematics Section, EPFL 02.2015 - 12.2019
Teaching assistant

Worked as teaching assistant in: (i) Analysis and Numerical Analysis courses for bachelor students, and (ii) courses in scientific programming and High Performance Computing in C and C++ for master students.

Siemens AG, München (Germany) 02.2016 - 08.2016
Intern

Worked in the Smart Embedded systems / Multicore Group on the EMB² library (C++ library for parallel computations).

Tasks included: implementation of concurrent data structures and algorithms, use of the library to create test cases to demonstrate the performance gain.

ADDITIONAL EXPERIENCES

EPFL Chapter of SIAM, EPFL
President

09.2018 - 09.2019

Taken the role of president of the EPFL Chapter of SIAM (Society for Industrial and Applied Mathematics), which aims at promoting interchanges between academia and industry through events specifically targeted to bachelor and master students.

Tasks included: coordination of members of the association, host events.

SCIENTIFIC PUBLICATIONS

1. **Luca Pegolotti**, Martin Pfaller, Alison Marsden, Simone Deparis. Model order reduction of flow based on a modular geometrical approximation of blood vessels. *Submitted to Computer Methods in Applied Mechanics and Engineering*.
2. Jean Bonnemain, Matthias Zeller, **Luca Pegolotti**, Lucas Liaudet, and Simone Deparis. Deep neural network implementation for evaluation of left ventricular systolic function in patients under left ventricular assist device. *In preparation*.
3. Jean Bonnemain, **Luca Pegolotti**, Lucas Liaudet, and Simone Deparis. Implementation and calibration of a deep neural network to predict parameters of left ventricular systolic function based on pulmonary and systemic arterial pressure signals. *Frontiers in Physiology 11 (2020): 1086*.
4. Niccolò Dal Santo, Simone Deparis, and **Luca Pegolotti**. Data driven approximation of parametrized PDEs by Reduced Basis and Neural Networks. *Journal of Computational Physics (2020): 109550*.
5. Simone Deparis, Antonio Iubatti, and **Luca Pegolotti**. Coupling non-conforming discretizations of PDEs by spectral approximation of the Lagrange multiplier space. *ESAIM: Mathematical Modelling and Numerical Analysis 53.5 (2019): 1667-1694*.
6. Simone Deparis, Michel O. Deville, Filippo Menghini, **Luca Pegolotti**, Alfio Quarteroni. Application of the Rosenbrock methods to the solution of unsteady 3D incompressible Navier-Stokes equations. *Computers Fluids 179 (2019): 112-122*.
7. **Luca Pegolotti**, Luca Dede, and Alfio Quarteroni. Isogeometric Analysis of the electrophysiology in the human heart: Numerical simulation of the bidomain equations on the atria. *Computer Methods in Applied Mechanics and Engineering 343 (2019): 52-73*.

AWARDS

- Awarded the Doc.Mobility scholarship by the Swiss National Foundation. The scholarship funded the researcher carried out at Stanford.
- Rewarded by president of the Politecnico di Milano, Prof. Giovanni Azzone, for the best score in the admission test (100/100), during the "Politest Top Schools" event.

TECHNICAL SKILLS

Programming	C, C++, Python, Matlab
Software	Eclipse, Microsoft Visual Studio, Version control systems (Git)
Operating Systems	Unix based systems (Mac OS/Linux), Windows
Github profile	https://github.com/lucapegolotti

LANGUAGES

Italian	Native
English	Fluent (C1)
French	Intermediate (B2)
German	Basic (A1/A2)

CONFERENCES & WORKSHOPS

- **Virtual Physiological Human 2020**, online conference, poster presenter 08.2020
Title: Reduced Order Modeling of blood flow with a Reduced Basis approach and Domain Decomposition.
- **Coupled Problems**, Sitges (Barcelona), presenter 07.2019
Title: Coupling non-conforming discretizations of PDEs by means of spectral Lagrange multipliers.
- **Finite Elements in Fluids**, Chicago, presenter 04.2019
Title: Efficient Reduced Coupling of non-conforming discretization of PDEs.
- **Scientific Machine Learning**, ICERM (Brown University), attendee 01.2019
- **World Congress in Computational Mechanics XIII**, New York, presenter 07.2018
Title: Efficient Reduced Coupling of PDEs Based on Weak Transmission Conditions.
- **Swiss Numerics Day**, ETH Zurich, poster presenter 04.2018
Title: Coupling non-conforming discretization of PDEs by spectral approximation of the Lagrange multiplier space.
- **Model Reduction of Parametrized Systems IV**, Nantes, attendee 04.2018

CO-SUPERVISED PROJECTS

- Master Project: Reduced order coupling in domain decomposition methods for parametrized PDEs (Antonio Iubatti, Double degree Politecnico di Milano-EPFL) – *Spring 2018*
- Semester Project: PDE-aware Machine Learning for Parametrized Parabolic PDEs (Riccardo Tenderini, Double degree Politecnico di Milano-EPFL) – *Spring 2019*
- Semester Project: Enhancing high-order time advancing schemes for PDEs by means of the Reduced Basis method (Benjamin Pedrotti, Master Program in CSE) – *Fall 2019*